

Jan Rauch
Zbigniew W. Raś
Petr Berka
Tapio Elomaa (Eds.)

LNAI 5722

Foundations of Intelligent Systems

18th International Symposium, ISMIS 2009
Prague, Czech Republic, September 2009
Proceedings

 Springer

Lecture Notes in Artificial Intelligence

5722

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

Subseries of Lecture Notes in Computer Science

Jan Rauch Zbigniew W. Raś Petr Berka
Tapio Elomaa (Eds.)

Foundations of Intelligent Systems

18th International Symposium, ISMIS 2009
Prague, Czech Republic, September 14-17, 2009
Proceedings

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

Jan Rauch, Petr Berka
University of Economics
Faculty of Informatics and Statistics
130 67, Prague 3, Czech Republic
E-mail: {rauch; berka}@vse.cz

Zbigniew W. Raś
University of North Carolina
Department of Computer Science
Charlotte, NC 28223, USA and
Polish-Japanese Institute
of Information Technology, Warsaw, Poland
E-mail: ras@uncc.edu

Tapio Elomaa
Tampere University of Technology
Institute of Software Systems
33101 Tampere, Finland
E-mail: tapio.elomaa@tut.fi

Library of Congress Control Number: 2009933282

CR Subject Classification (1998): I.2, H.2.8, H.3, H.4, H.5, I.5, I.2.3

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 1867-8211
ISBN-10 3-642-04124-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-04124-2 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.com

© Springer-Verlag Berlin Heidelberg 2009
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12752737 06/3180 5 4 3 2 1 0

Preface

This volume contains the papers selected for presentation at the 18th International Symposium on Methodologies for Intelligent Systems - ISMIS 2009, held in Prague, Czech Republic, September 14–17, 2009. The symposium was organized by the University of Economics, Prague. ISMIS is a conference series that was started in 1986. Held twice every three years, ISMIS provides an international forum for exchanging scientific research and technological achievements in building intelligent systems.

The following major areas were selected for ISMIS 2009: knowledge discovery and data mining, text mining, machine learning, applications of intelligent systems, logical and theoretical aspects of intelligent systems, information processing, agents, complex data, general AI, and uncertainty.

The 60 contributed papers were selected from 111 full-draft papers by the international Program Committee with 77 members from 23 countries and by 33 external referees. In addition, four plenary talks were given by Heikki Mannila, Gerhard Widmer, Maria Zemankova and by Jozef Kelemen. The plenary talks concerned hot topics relevant to ISMIS 2009. The First International Workshop on Topological Learning was organized by Michaël Aupetit, Hakim Hacid and Djamel A. Zighed.

ISMIS 2009 was supported and sponsored by the University of Economics, Prague, by Seznam.cz, a.s., by Československá obchodní banka, a. s., by The Czech Society for Cybernetics and Informatics and by TOVEK, spol. s r.o.

We wish to express our thanks to all ISMIS 2009 reviewers and to Heikki Mannila, Gerhard Widmer, Maria Zemankova and Jozef Kelemen, who presented invited talks at the symposium. We would like to express our appreciation to the supporters and sponsors of the symposium and to all authors of submitted papers. Our sincere thanks go to Aijun An, Jaime Carbonell, Nick Cercone, Floriana Esposito, Mohand-Said Hacid, Donato Malerba, Stan Matwin, Neil Murray, Zbigniew W. Raś (Chair), Lorenza Saitta, Giovanni Semeraro, Dominik Ślęzak, Shusaku Tsumoto, and Maria Zemankova, who served as members of ISMIS 2009 Steering Committee. We are grateful to Milena Zeithamlová and to the Action M Agency for their effort in the organization of the conference. Our thanks also go to Tomáš Kliegr and David Chudán who were responsible for preparing and managing the ISMIS 2009 website. Moreover, our thanks are due to Alfred Hofmann of Springer for his continuous support.

September 2009

Jan Rauch
Zbigniew W. Raś
Petr Berka
Tapio Elomaa

Organization

ISMIS 2009 was organized by the University of Economics, Prague, Faculty of Informatics and Statistics.

Executive Committee

General Chair	Zbigniew W. Raś(UNC-Charlotte, USA and PJIIT, Warsaw, Poland)
Conference Chair	Jan Rauch (University of Economics, Prague, Czech Republic)
Program Co-chairs	Petr Berka (University of Economics, Prague, Czech Republic) Tapio Elomaa (Tampere University of Technology, Finland)
Workshop Chair	Vojtěch Svátek (University of Economics, Prague, Czech Republic)
Organizing Chair	Milena Zeithamlová (Action M Agency, Prague, Czech Republic)

Steering Committee

Zbigniew W. Raś	UNC-Charlotte, USA and PJIIT, Warsaw, Poland, Chair
Aijun An	York University, Canada
Jaime Carbonell	CMU, USA
Nick Cercone	York University, Canada
Floriana Esposito	University of Bari, Italy
Mohand-Said Hacid	University of Lyon 1, France
Donato Malerba	University of Bari, Italy
Stan Matwin	University of Ottawa, Canada
Neil Murray	SUNY at Albany, USA
Lorenza Saitta	University of Piemonte Orientale, Italy
Giovanni Semeraro	University of Bari, Italy
Dominik Ślęzak	Infobright Inc., Canada and PJIIT, Warsaw, Poland
Shusaku Tsumoto	Shimane Medical University, Japan
Maria Zemankova	NSF, USA

Program Comittee

Luigia Carlucci Aiello, Italy
Troels Andreasen, Denmark
Salima Benbernou, France
Mária Bieliková, Slovak Republic
Ivan Bratko, Slovenia
Pavel Brazdil, Portugal
Cory Butz, Canada
Longbing Cao, Australia
Sandra Carberry, USA
Michelangelo Cecii, Italy
Bruno Cremilleux, France
Juan Carlos Cubero, Spain
Alfredo Cuzzocrea, Italy
Agnieszka Dardzinska, Poland
Edward Fox, USA
Dragan Gamberger, Croatia
Attilio Giordana, Italy
Vladimir Gorodetsky, Russia
Salvatore Greco, Italy
Jerzy Grzymala-Busse, USA
Allel Hadjali, France
Mirsad Hadzikadic, USA
Howard Hamilton, Canada
Xiaohua Tony Hu, USA
Seunghyun Im, USA
Nathalie Japkowicz, Canada
Radim Jiroušek, Czech Republic
Janusz Kacprzyk, Poland
Jozef Kelemen, Czech Republic
Mieczyslaw Klopotek, Poland
Jacek Koronacki, Poland
Patrick Lambrix, Sweden
Rory Lewis, USA
Chao-Lin Liu, Taiwan
Jiming Liu, Hong Kong
Ramon López de Mántaras, Spain
Michael Lowry, USA
David Maluf, USA
Krzysztof Marasek, Poland

Nicola Di Mauro, Italy
Paola Mello, Italy
John Mylopoulos, Canada
Pavol Návrat, Slovak Republic
Ján Paralič, Slovak Republic
Petra Perner, Germany
James Peters, Canada
Jean-Marc Petit, France
Juliana Peneva, Bulgaria
Jaroslav Pokorný, Czech Republic
Lubomír Popelínský, Czech Republic
Henri Prade, France
Seppo Puuronen, Finland
Naren Ramakrishnan, USA
William Ribarsky, USA
Gilbert Ritschard, Switzerland
Henryk Rybinski, Poland
Nahid Shahmehri, Sweden
Luo Si, USA
Arno Siebes, The Netherlands
Andrzej Skowron, Poland
Václav Snášel, Czech Republic
Jerzy Stefanowski, Poland
Olga Štěpánková, Czech Republic
Einoshin Suzuki, Japan
Vojtěch Svátek, Czech Republic
Krishnaprasad Thirunarayan, USA
Li-Shiang Tsay, USA
Athena Vakali, Greece
Christel Vrain, France
Gerhard Widmer, Austria
Alicja Wieczorkowska, Poland
Xintao Wu, USA
Xindong Wu, USA
Yiyu Yao, Canada
Xin Zhang, USA
Ning Zhong, Japan
Djamel Zighed, France

External Referees

Timo Aho	Corrado Loglisci	Francois Riout
Patrick Bosc	Donato Malerba	Markus Schedl
Federico Chesani	Marco Montali	Klaus Seyerlehner
Frederic Flouvat	Martin Mozina	Davide Sottara
Matej Guid	Andrea Orlandini	Konstantinos Stamos
Mohana Gurram	Symeon Papadopoulos	Lena Strömbäck
Tomáš Kincl	Jos Pena	He Tan
Jussi Kujala	Olivier Pivert	Rmi Tournaire
Martin Labský	Marc Plantevit	Szymon Wilk
Paolo Liberatore	Tim Pohle	Seungwon Yang
Ludovic Lietard	Riccardo Rasconi	Sine Zambach

Sponsoring and Supporting Institutions

Laboratory for Intelligent Systems, Faculty of Informatics and Statistics,
University of Economics, Prague

Seznam.cz, a.s.

Československá obchodní banka, a. s.

The Czech Society for Cybernetics and Informatics

TOVEK, spol. s r.o.

Springer, Heidelberg, Germany

Table of Contents

Invited Papers

Randomization Methods for Assessing the Significance of Data Mining Results	1
<i>Heikki Mannila</i>	
Dealing with Music in Intelligent Ways	2
<i>Gerhard Widmer</i>	
Intelligent Systems: From R.U.R. to ISMIS 2009 and beyond	3
<i>Maria Zemankova</i>	
The Art of Management and the Technology of Knowledge-Based Systems	5
<i>Jozef Kelemen and Ivan Polášek</i>	

Knowledge Discovery and Data Mining

Frequent Itemset Mining in Multirelational Databases	15
<i>Aída Jiménez, Fernando Berzal, and Juan-Carlos Cubero</i>	
A Multiple Scanning Strategy for Entropy Based Discretization	25
<i>Jerzy W. Grzymala-Busse</i>	
Fast Subgroup Discovery for Continuous Target Concepts	35
<i>Martin Atzmueller and Florian Lemmerich</i>	
Discovering Emerging Graph Patterns from Chemicals	45
<i>Guillaume Poezevara, Bertrand Cuissart, and Bruno Crémilleux</i>	
Visualization of Trends Using RadViz	56
<i>Lenka Nováková and Olga Štěpánková</i>	
Action Rules Discovery Based on Tree Classifiers and Meta-actions	66
<i>Zbigniew W. Raś and Agnieszka Dardzińska</i>	
Action Rules and the GUHA Method: Preliminary Considerations and Results	76
<i>Jan Rauch and Milan Šimůnek</i>	
Semantic Analytical Reports: A Framework for Post-processing Data Mining Results	88
<i>Tomáš Kliegr, Martin Ralbovský, Vojtěch Svátek, Milan Šimůnek, Vojtěch Jirkovský, Jan Nemrava, and Jan Zemánek</i>	

Applications of Intelligent Systems in Medicine

Medical Decision Making through Fuzzy Computational Intelligent Approaches	99
<i>Elpiniki I. Papageorgiou</i>	
Fuzzy Cognitive Map Based Approach for Assessing Pulmonary Infections	109
<i>Elpiniki I. Papageorgiou, Nikolaos Papandrianos, Georgia Karagianni, G. Kyriazopoulos, and D. Sfyras</i>	
A Knowledge-Based Framework for Information Extraction from Clinical Practice Guidelines	119
<i>Corrado Loglisci, Michelangelo Ceci, and Donato Malerba</i>	
RaJoLink: A Method for Finding Seeds of Future Discoveries in Nowadays Literature	129
<i>Tanja Urbančič, Ingrid Petrič, and Bojan Cestnik</i>	

Logical and Theoretical Aspects of Intelligent Systems

Automatic Generation of P2P Mappings between Sources Schemas	139
<i>Karima Toumani, H�elene Jaudoin, and Michel Schneider</i>	
An OWL Ontology for Fuzzy OWL 2	151
<i>Fernando Bobillo and Umberto Straccia</i>	
Fuzzy Clustering for Categorical Spaces: An Application to Semantic Knowledge Bases	161
<i>Nicola Fanizzi, Claudia d’Amato, and Floriana Esposito</i>	
Reasoning about Relations with Dependent Types: Application to Context-Aware Applications	171
<i>Richard Dapoigny and Patrick Barlatier</i>	
Quasi-Classical Model Semantics for Logic Programs – A Paraconsistent Approach	181
<i>Zhihu Zhang, Zuoquan Lin, and Shuang Ren</i>	
Prime Implicates and Reduced Implicate Tries	191
<i>Neil V. Murray and Erik Rosenthal</i>	
Logic for Reasoning about Components of Persuasive Actions	201
<i>Katarzyna Budzynska, Magdalena Kacprzak, and Pawel Rembelski</i>	
A Hybrid Method of Indexing Multiple-Inheritance Hierarchies	211
<i>Jacek Lewandowski and Henryk Rybinski</i>	

Text Mining

Theme Extraction from Chinese Web Documents Based on Page Segmentation and Entropy	221
<i>Deqing Wang, Hui Zhang, and Gang Zhou</i>	
Topic-Based Hard Clustering of Documents Using Generative Models . . .	231
<i>Giovanni Ponti and Andrea Tagarelli</i>	
Boosting a Semantic Search Engine by Named Entities	241
<i>Annalina Caputo, Pierpaolo Basile, and Giovanni Semeraro</i>	
Detecting Temporal Trends of Technical Phrases by Using Importance Indices and Linear Regression	251
<i>Hidenao Abe and Shusaku Tsumoto</i>	

Applications of Intelligent Systems in Music

Detecting Emotions in Classical Music from MIDI Files	261
<i>Jacek Grekow and Zbigniew W. Raś</i>	
Mining Musical Patterns: Identification of Transposed Motives	271
<i>Fernando Berzal, Waldo Fajardo, Aída Jiménez, and Miguel Molina-Solana</i>	
Musical Instruments in Random Forest	281
<i>Miron Kursa, Witold Rudnicki, Alicja Wieczorkowska, Elżbieta Kubera, and Agnieszka Kubik-Komar</i>	
Application of Analysis of Variance to Assessment of Influence of Sound Feature Groups on Discrimination between Musical Instruments	291
<i>Alicja Wieczorkowska and Agnieszka Kubik-Komar</i>	

Information Processing

Alternative Formulas for Rating Prediction Using Collaborative Filtering	301
<i>Amar Saric, Mirsad Hadzikadic, and David Wilson</i>	
On Three Classes of Division Queries Involving Ordinal Preferences . . .	311
<i>Patrick Bosc, Olivier Pivert, and Olivier Soufflet</i>	
Analyses of Knowledge Creation Processes Based on Different Types of Monitored Data	321
<i>Ján Paralič, František Babič, Jozef Wagner, Ekaterina Simonenko, Nicolas Spyratos, and Tsuyoshi Sugibuchi</i>	

Intelligent Information Processing in Semantically Enriched Web	331
<i>Pavol Návrat, Mária Bielíková, Daniela Chudá, and Viera Rozínajová</i>	

Agents

Modeling Ant Activity by Means of Structured HMMs	341
<i>Guenael Cabanes, Dominique Fresnau, Ugo Galassi, and Attilio Giordana</i>	
Modern Approach for Building of Multi-Agent Systems	351
<i>Lukasz Chomatek and Aneta Poniszewska-Marańda</i>	
Relational Sequence Clustering for Aggregating Similar Agents	361
<i>Grazia Bombini, Nicola Di Mauro, Stefano Ferilli, and Floriana Esposito</i>	
FutureTrust Algorithm in Specific Factors on Mobile Agents	371
<i>Michał Wolski and Mieczysław Kłopotek</i>	

Machine Learning

Ensembles of Abstaining Classifiers Based on Rule Sets	382
<i>Jerzy Błaszczyński, Jerzy Stefanowski, and Magdalena Zając</i>	
Elicitation of Sugeno Integrals: A Version Space Learning Perspective . . .	392
<i>Henri Prade, Agnes Rico, and Mathieu Serrurier</i>	
Efficient MAP Inference for Statistical Relational Models through Hybrid Metaheuristics	402
<i>Marenglen Biba, Stefano Ferilli, and Floriana Esposito</i>	
Combining Time and Space Similarity for Small Size Learning under Concept Drift	412
<i>Indrė Žliobaitė</i>	
Similarity and Kernel Matrix Evaluation Based on Spatial Autocorrelation Analysis	422
<i>Vincent Pisetta and Djamel A. Zighed</i>	

Applications of Intelligent Systems

Job Offer Management: How Improve the Ranking of Candidates	431
<i>Rémy Kessler, Nicolas Béchet, Juan-Manuel Torres-Moreno, Mathieu Roche, and Marc El-Béze</i>	

Discovering Structured Event Logs from Unstructured Audit Trails for Workflow Mining	442
<i>Liqiang Geng, Scott Buffett, Bruce Hamilton, Xin Wang, Larry Korba, Hongyu Liu, and Yunli Wang</i>	
GIS-FLSolution: A Spatial Analysis Platform for Static and Transportation Facility Location Allocation Problem	453
<i>Wei Gu, Xin Wang, and Liqiang Geng</i>	
A CBR System for Knowing the Relationship between Flexibility and Operations Strategy	463
<i>Daniel Arias-Aranda, Juan L. Castro, Maria Navarro, and José M. Zurita</i>	
Semantic-Based Top-k Retrieval for Competence Management	473
<i>Umberto Straccia, Eufemia Tinelli, Simona Colucci, Tommaso Di Noia, and Eugenio Di Sciascio</i>	
A New Strategy Based on GRASP to Solve a Macro Mine Planning	483
<i>María-Cristina Riff, Eridan Otto, and Xavier Bonnaire</i>	
Food Wholesales Prediction: What Is Your Baseline?	493
<i>Jorn Bakker and Mykola Pechenizkiy</i>	

Complex Data

A Distributed Immunization Strategy Based on Autonomy-Oriented Computing	503
<i>Jiming Liu, Chao Gao, and Ning Zhong</i>	
Discovering Relevant Cross-Graph Cliques in Dynamic Networks	513
<i>Loïc Cerf, Tran Bao Nhan Nguyen, and Jean-François Boulicaut</i>	
Statistical Characterization of a Computer Grid	523
<i>Lovro Ilijašić and Lorenza Saitta</i>	
On Social Networks Reduction	533
<i>Václav Snášel, Zdeněk Horák, Jana Kočibová, and Ajith Abraham</i>	
Networks Consolidation through Soft Computing	542
<i>Sami Habib, Paulvanna Nayaki Marimuthu, and Mohammad Taha</i>	
Lacking Labels in the Stream: Classifying Evolving Stream Data with Few Labels	552
<i>Clay Woolam, Mohammad M. Masud, and Latifur Khan</i>	
Novelty Detection from Evolving Complex Data Streams with Time Windows	563
<i>Michelangelo Ceci, Annalisa Appice, Corrado Loglisci, Costantina Caruso, Fabio Fumarola, and Donato Malerba</i>	

General AI

On Computational Creativity, ‘Inventing’ Theorem Proofs	573
<i>Marta Fraňová and Yves Kodratoff</i>	
Revisiting Constraint Models for Planning Problems	582
<i>Roman Barták and Daniel Toropila</i>	

Uncertainty

Interval-Valued Fuzzy Formal Concept Analysis	592
<i>Yassine Djouadi and Henri Prade</i>	
Application of Meta Sets to Character Recognition	602
<i>Bartłomiej Starosta</i>	
A General Framework for Revising Belief Bases Using Qualitative Jeffrey’s Rule	612
<i>Salem Benferhat, Didier Dubois, Henri Prade, and Mary-Anne Williams</i>	
Author Index	623

Randomization Methods for Assessing the Significance of Data Mining Results

Heikki Mannila^{1,2}

¹ Helsinki Institute for Information Technology HIIT

² University of Helsinki and Helsinki University of Technology

Heikki.Mannila@tkk.fi

Abstract. Data mining research has developed many algorithms for various analysis tasks on large and complex datasets. However, assessing the significance of data mining results has received less attention. Analytical methods are rarely available, and hence one has to use computationally intensive methods. Randomization approaches based on null models provide, at least in principle, a general approach that can be used to obtain empirical p-values for various types of data mining approaches. I review some of the recent work in this area, outlining some of the open questions and problems.

Dealing with Music in Intelligent Ways

Gerhard Widmer^{1,2}

¹ Department of Computational Perception, Johannes Kepler University Linz

² Austrian Research Institute for Artificial Intelligence, Vienna, Austria

gerhard.widmer@jku.at

<http://www.cp.jku.at/people/widmer>

Abstract. Music is not just a product of human creativity and a uniquely human means of expression, it is also a commodity of great commercial relevance. The rapid digitisation of the music market, with the global availability of ever larger amounts of music, is creating a need for musically intelligent computer systems, and lots of opportunities for exciting research.

The presentation gives an impression of latest research in the field of intelligent music processing and music information retrieval. Based (mostly) on our own recent work, I discuss what it means for a computer to be musically intelligent, describe some of the techniques that are being developed, and demonstrate how entirely new musical interfaces and devices become possible with such methods – devices that, in effect, will change the way we listen to, and interact with, music.

Intelligent Systems: From R.U.R. to ISMIS 2009 and beyond

Maria Zemankova

National Science Foundation, Arlington, VA 22230, USA
mzemanko@nsf.gov

Abstract. History of humans trying to understand and provide foundations for human reasoning or intelligence can be traced back to 11th century BC Babylonia reasoning in medical diagnostics and astronomy, Chinese tradition of thought that blossomed between 10th and 6th centuries BC and produced Analects of Confucius, 6th through 2nd century BC Indian philosophy and Hindu schools of thought, 4th century BC Greece where Aristotle gave birth to formal logic, and the quest continues into 21st century AD.

Humans have also been keen on creating “artificial life” or “intelligent systems”. The Old Testament that was created between the 12th and the 2nd century BC mentions a “servant” made from clay called Golem. However, you are now in Prague, where according to the legend the 16th century the chief Rabbi Loew constructed the Golem out of clay from the banks of Vltava (Moldau) river and brought it to life to protect the Jewish Ghetto. However, Golem grew it became increasingly violent, spreading fear, killing and eventually turning on its creator. It is believed that the first record of a “homunculus” (representation of a human being) appeared in alchemical literature in the 3rd century AD. A different branch of inquiry aimed at building “automata”, or self-operating artificial systems that have been made to resemble human or animal actions. These can be traced to 3rd century BC China, Greek Antikythera mechanism built about 150-100 BC, 8th century Muslim alchemist, Jabir ibn Hayyan (Geber) who in his coded Book of Stones included recipes for constructing artificial snakes, scorpions, and humans.

In 1921, a Czech writer Karel Capek gave birth to the word “robot” in his science fiction play “R.U.R.” (Rossum’s Universal Robots). Today, robots produce cars, explore the universe, search for victims in natural disasters, perform delicate surgeries, harvest crops, and cuddly robots with capabilities to talk are companions to elderly and children - as portrayed in another Czech science fiction from 1962 “Kybernetická Babicka” (Cybernetic Grandmother) by Jir Trnka - a puppet maker, illustrator, motion-picture animator and film director.

The quest to understand intelligence and create intelligent systems has a history of 32 centuries, while the International Symposium on Methodologies for Intelligent System that originated in 1986 has as of this year (2009) devoted 23 years to advancing the state of the art in our understanding intelligent behavior - both human and machine, and developing more effective intelligent machine systems or human-machine systems.

In the early years, ISMIS focused on areas of Approximate Reasoning, Expert Systems, Intelligent Databases, Knowledge Representation, Learning and Adaptive Systems, Logic for Artificial Intelligence, and Man-Machine Interaction. ISMIS 2009 called for papers in the areas of Active Media Human-Computer

Interaction, Autonomic and Evolutionary Computation, Digital Libraries, Intelligent Agent Technology, Intelligent Information Retrieval, Intelligent Information Systems, Intelligent Language Processing, Knowledge Representation and Integration, Knowledge Discovery and Data Mining, Knowledge Visualization, Logic for Artificial Intelligence, Music Information Retrieval, Soft Computing, Text Mining, Web Intelligence, Web Mining, and Web Services.

Have we solved the problems that are not in ISIMS 2009 focus? Where do we stand? What have we learned? What are we teaching? Where should we focus our attention in the years beyond 2009? This talk will outline open research problems, educational and other societal issues, and opportunities for cooperation in methodologies for intelligent systems.

The Art of Management and the Technology of Knowledge-Based Systems

Jozef Kelemen^{1,2} and Ivan Polášek³

¹ VSM College of Management, 851 04 Bratislava, Slovakia

² Inst. of Comp. Sci., Silesian University, 746 01 Opava, Czech Republic

³ Inst. of Informatics and Software Engineering, STU, Bratislava, Slovakia

jkelemen@vsm.sk, ipo@gratex.com

Abstract. Explicit knowledge is successfully transferrable into computers. As the consequence of this, we have today at hand various knowledge and expert systems. The talk provides a short overview of some basic steps towards the actual situation. Then it focuses to the role of management for effective dealing with knowledge, and to the role of special kind of knowledge – the knowledge of management. A new type of knowledge storing and processing technology, resulting in specific type of knowledge-based systems – the *Knowledge Managing Systems* – is proposed as a computer-based support for activities which form at least some part of the Art of Management.

Keywords: Workmanship, knowledge, knowledge-based systems, ontology, management, knowledge management, knowledge managing systems.

1 On the Structure of Workmanship

During the run of the historical time, the originally unstructured general knowledge has been structured according various profession, for instance, divided into several types supporting different capabilities and activities (like the common-sense knowledge, specialized experts' knowledge, craftsmen' skills, leaders' managing abilities) etc. One type of knowledge is in our times successfully transferrable into our present time computers. As the consequence of this possibility we have today at hand various knowledge-based and expert systems for improving the quality of many of our problem-solving, decision-making, diagnostic, planning, etc. activities.

The contribution provides a short overview of some basic steps towards the actual situation. Then it will focus to the role of management for effective dealing with knowledge, and to the role of special kind of knowledge – the knowledge of management.

A new type of knowledge storing and processing technology, resulting in specific type of knowledge-based systems – the *Knowledge Managing Systems* – will be then proposed as a useful, and perhaps also as an effective enough computer-based support for improving the quality of activities which form at least some part(s) of the Art of Management.

The workman's state of being a perfectly informed and skilful in his profession, in other words his *workmanship*, is a phenomenon which has been very closely related with the human knowledge for several thousands of years of human civilization.

However, the notion of workmanship covers something more than *to know* something only. From the perspective of applying of knowledge workmanship can be divided into two equally important parts (Drucker, 2007, Chapter 5): to professional *knowledge* and to craft *skills*. The third, for centuries hidden, capacity of all masters and craftsmen, it is their competency to *organize* the effective production and efficient sale of their goods. This activity has been in 20th century's economy denoted by the trendy word *management*.

In the next sections we analyze the structure of knowledge as well as that of the skill. We concentrate to their mutual interrelations, as well as to the relation of both of them to computing, in particular to their functions in knowledge (of knowledge-based) systems.

Moreover, we will show the way how the transfer of the knowledge processing to computerized *knowledge-based systems*, and the possibility to transfer (some of) skill to automated technological processes – to *robotic systems* in particular – prepares conditions for the growth of the role of managing knowledge – for *knowledge management*.

2 On the Structure of Knowledge and Skills, and the GOFAI

Expert's specialized knowledge and skills about some field of human activity – or, using an obsolescence, any *workmanship* – is internally complicated body of rules, taxonomies, uncertainties, conditionings, exclusions, motor skills, etc. To become an expert in the particular field supposes to be well and deeply oriented in all of these aspects. To accelerate such a preparation requires dividing the general knowledge into the smaller and simpler parts, to some “nuggets” of knowledge, or, in other words, to (a finite number of) *pieces of knowledge*. Complicated (and just mentioned) relations between these pieces of knowledge form then the whole knowledge on the given field of expertise.

The skill related to any workmanship can be also divided into simpler elementary parts. We will call these parts of skill as *operations*, because they are the basic “building blocks” of any complicated modes of operations based and closely related with any master knowledge.

In order to relate operations with pieces of knowledge, let us realize one among the basic moral from the traditional artificial intelligence research, one among the corner stones of the so called GOFAI (the Good, Old Fashioned Artificial Intelligence) as presented in several university course books, e.g. in (Winston, 1977): Let us characterize the knowledge (as well as the pieces of knowledge) by its three basic attributes – its declarability, its procedurality, and its associability.

The attribute of declarability means that any knowledge is expressible in certain rigorous form of syntactically correct symbol structure, so it is symbolically representable. The representability of knowledge is in fact the base of its use in any traditional GOFAI systems (expert systems, knowledge-based systems).

Another attribute of knowledge is the ability to connect each piece of knowledge by another pieces of knowledge in order to express (to represent) their contextual interrelatedness, or in other word, to associate knowledge pieces which some other semantically connected pieces. This is the attribute of *associability* of knowledge. It

such a way, the semantically interrelated, associated, pieces of knowledge forms a network-like structures usually called in GOFAI as associative or semantic net(work)s and today known as one among the conceptual base for creating and application of the so called *ontologies*.

The attribute of *procedurality* refers to the possibility to manipulate the pieces of knowledge. Such manipulations may transfer them into new object(s) – the new pieces of knowledge – with their new relations to their “origins” and, possibly, to other pieces of knowledge.

Because each piece of knowledge has to reflex all of the three inevitable attributes – its declarability, associability, and procedurality – it can also be associated with some basic elementary skill(s) of its use or application. For the same reason, *operation(s)* related to each piece of knowledge can be expressed by procedures dealing with this piece of knowledge. Note in this context that the equality between the number of pieces of knowledge (forming the general knowledge related to some mastership) and that of the operations (into which the expert skill is divided) is not accidental.

The emphasis put to one particular attribute leads to a particular type of knowledge representation structure. The *declarability* resulted in declarative representation schemes like rules successfully used perhaps first in the field of artificial intelligence by A. Newell and H. A. Simon (Newell, Simon, 1972) or formalisms and programming tools based on formal logics and automatic theorem proving, like the famous systems, in fact a declarative programming tool, PROLOG, etc. The *associability* has resulted – as we have mentioned already – in associative networks, and the *procedurality* in different types of procedural representation schemes, like Hewitt’s Planner, developed for experimentation in robotics at the former MIT AI Laboratory (Hewitt, 1972).

The effort to integrate the positive sides of all previously mentioned representation schemes, as well as to integrate them into a representational scheme of some other aspects of knowledge, like *uncertainties* (to the development of practically useful, and formalized enough methods of expressing formally and processing uncertainties the field of *computational intelligence* contributed significantly, e.g. by developing different fuzzy approaches) or *default values* (like different kinds of expectations usually related e.g. to stereotypical situations, and different types of commonsense knowledge) etc., led during the 70ties of the past century to different variations of schemes more or less similar to, but in basic principles almost identical with, the *frame representation* scheme as proposed in (Minsky, 1975) having different particular forms like *scenarios*, *scripts*, later *objects* used in the framework of the *object-oriented programming*, etc.

3 Knowledge Systems in Short

As we have already mentioned, some of the pieces of knowledge can be represented more straightforwardly in different representational formalism. Perhaps the most usual in the computerized *knowledge system* (*knowledge-based* or *expert* systems are expressions for denoting the same) are production rules. This representational scheme is usually considered as declarative one. An important consequence of the decision to use declarative representational schemes is the necessity to generalize the procedural parts of the represented knowledge, end include them in their universal form into the

general computational procedures forming the so called *inference engines* (a kind of interpreters of represented knowledge) of the knowledge systems. In the case of using more sophisticated representational schemes, however, for instance Minsky's frames, considerable part of the procedural knowledge can be expressed as part of the representation of the pieces of knowledge in frame systems, forming the representation of the whole general expert knowledge.

In any case, we are in present days faced with a situation when we have at hand sufficient representational schemes to represent (almost) all attributes of the required knowledge, and the knowledge systems are able to solve particular expert problems in the acceptable level of expertise; for more detail see e.g. (Stefik, 1995).

We can realize now that the knowledge representation as well as the exploitation of the knowledge – the use of necessary skill providing successful problem solving on the base of represented knowledge and using suitable inference engines – are activities executable using well programmed computers with a working *inference engine* and “stuffed” with a suitable amount and quality of input data (by *base of facts* describing the particular problem, and a *knowledge base* enable to knowledge systems to solve it). These activities are usually covered by the term *knowledge engineering*; for details and relations with knowledge management see e.g. (Schreiber et al., 2000).

The only part of the workmanship, unmentioned up to now In our consideration concerning the influence of information technologies to functioning of knowledge in production process, is the *management*. We will continue with it in the following Section.

4 Knowledge Management and Knowledge Managing Systems

The rapid progress in developing and application of information technology, esp. the progress in the fields like information systems development, and the growing field of application of knowledge-based systems in different areas of the research, industry and administration caused that the problems concerning the right management of knowledge becomes perhaps equally (but might be in some branches of professional activities more) important topic as the knowledge acquisition and knowledge-based problem-solving. The reason is obvious. Despite the enormous size and number of applications exploiting computerized knowledge, there is a wide area of knowledge that cannot be so easily captured and expressed in the form of pieces showing features of declarability, associability or procedurality to the degree that would allowed their (fully- or semi-) automated manipulation.

Knowledge management means in our context – similarly to (McElroy, 2003) – the large spectrum of activities connected with management of company's shared knowledge in the meaning of corporate knowledge decomposition, distribution, innovation, acquisition, accessibility, preservation, etc. The activities connected with satisfying of such requirements form the relatively traditional meaning of knowledge management. They are performed perhaps in all of the enterprises as the corporate knowledge is an important part of their function.

The direct use of the knowledge is often shifted to information technology – to knowledge-based (or knowledge-) systems. However, in connection with the use of knowledge-based systems some new kinds of problems appear. The solution of these kinds of problems requires some specific knowledge of management, and skills, too:

It is necessary to know how to organize the right conditions for effective and high quality knowledge acquisition process during the development of knowledge-bases of knowledge systems. It is necessary to organize the work of knowledge systems in the right way with respect to the requirements of users in different positions in the organization (often from the top management up to the product of technology engineers or technical support staff). The knowledge about why, when, and how to change the knowledge bases of knowledge systems used in the company or institution is necessary, etc. This is the second meaning of the knowledge management, specific for the enterprises which exploit the knowledge systems support.

Up to now, the transfer of activities specific for knowledge management to knowledge systems is rare. However, we see that this knowledge has – at least from the computational point of view – practically the same character as other knowledge already processed by computer-based knowledge storing and processing systems. The difference consists only in the problem space, but this fact has no crucial importance for an effort to develop specialized knowledge-based systems for storing and use of the knowledge necessary for (at least parts of) the activities important for the practice of knowledge management.

So, for the future, the discovering and developing technologies for construction and right use of computerized systems which will support knowledge management activities and will solve real problems of knowledge management practice – some kind of computerized *knowledge managing systems* (KM Systems, for short) – seems to be a promising field of research and engineering in computerized knowledge processing.

Because of the growing role of the knowledge management in the enterprises functioning, and the amount of specific knowledge and skills required from knowledge managers, it seems to be effective their specialized university level education, which provide not only the basics of the management in general, but complete their professional profile by knowledge and skills specific for the activities connected with managing knowledge in specific social and economic conditions of the *information and knowledge society*. For economic aspects of both see e.g. (Foray, 2004), and for first information on university education of knowledge management see (Kelemen, Hvorecký, 2008).

5 The *Gratex Knowledge Office* – An Example of a KM System

According to the common experience, up to circa 90% of all company information in the industrial countries exists in the present days in hard copies. According the same experience, from all of documents processed on daily basis, 90% are non-categorized and incomplete, while the average annual increase of the amount of documents in companies reaches circa 22%. So, e.g. the following complications can be observed:

Information and documentation are stored at several places which means that they are often incompact and disorganized, and often many of the documents are archived only in hard copies. Stored information is duplicate or not complete. Physical presence of persons participating in processes is required. In other words, *managers must be physically involved in decision-making processes and final approvals.*

Managerial information about resources is insufficient. So, to gain the information about availability and condition of available resources, more time is needed.

The fulfillment of assigned tasks is not controlled efficiently. After a period of time or after personnel changes, the *responsibilities and the extent to which the tasks were fulfilled are not clear.*

The allocation of assets and authorities to employees is not transparent. Allocated resources and the authority of the employees to carry out particular activities, i.e. use the resources, are not registered properly.

Decision-making competencies are not clearly personalized, decisions are not registered, and a *summary of impact of the decisions on the development of the company is missing.*

5.1 The Running Solution

Gratex Knowledge Office (GKO) is a knowledge based platform designed for the effective management of company knowledge base, control of a company's internal processes and project management. It can be used in a wide variety of companies with diverse specializations and eliminate some of the above mentioned troubles and insufficiencies. The architecture of the planned new modul of GKO combines the advantages of expert system developing environments (like Jess, Clips, for instance), datamining methods and indexing systems. A knowledge base in GKO is a searchable database which contains the accumulated knowledge of specialists in a particular field of documents interpretation and processing. The knowledge base should support canned answers, FAQs, customer interaction histories, solved problems, and self-help question and answer pairs. It should also facilitate easy integration of new expertise as it becomes available. The system work with a given document (its inference) starts with scanning ontologies and using the graph algorithms to find relevant documents. Then it is possible to fire prepared rules as well as discovered associative rules in the supporting expert system knowledge base. In this manner, dynamic ontologies increase the knowledge of the system whole system on how to deal with the given document, how to understand its contents and its relevance for other documents in the systems depository.

5.2 Main Ideas and Features

GKO.NET is now a user-defined document workflow and content management system based on the overall framework of *GKO*. Its variability, wide control options, and document distribution make a company's control system more effective and transparent. It enables an internal information sharing and management, based on predefined unified procedures and regulations. Information is transmitted via electronic documents of diverse types, saved on the central server. These types of documents register the development of internal processes in companies. *GKO.NET* enables also to define the roles and powers of employees transparently. It is an appropriate tool for global organization control, and effective teamwork. It represents the result of the further development, motivated by the effort to incorporate knowledge management aspects, of a previous similar tool developed by Gratex International, that won a prize in the *Microsoft Enterprise Solution of the Year 2001* competition.

GKO.NET offers the opportunities for effective management of key company processes, such as:

Standard processes (like business administration, decision-making processes, quality management, purchasing, sales, etc.).

Safety management (management of information systems and storages, assignment of system rights and accesses to individuals and groups, risk monitoring, monitoring of weaknesses, threats, and damages, etc.).

Document administration (registering, sharing, updating, backup and printing of various documents, and templates administration).

Human resources (hiring, profiles, qualification, trainings, and courses etc., availability information, as attendance, absences, sick leave, business trips, payroll administration, etc.).

Asset management (registration and categorization of assets, allocation of work tools to the employees).

Project management (definitions of project teams, scopes of delivery, deadlines, risks, assumptions, documentation, task planning, recording suggestions and changes, quality management, controlling etc.).

5.3 Structure of the System

Access of users to *GKO.NET* is simple, enabled through common network, or the Internet, and its implementation requires no significant changes in the existing infrastructure. The application is easy to adapt to customer specific needs. It provides for a

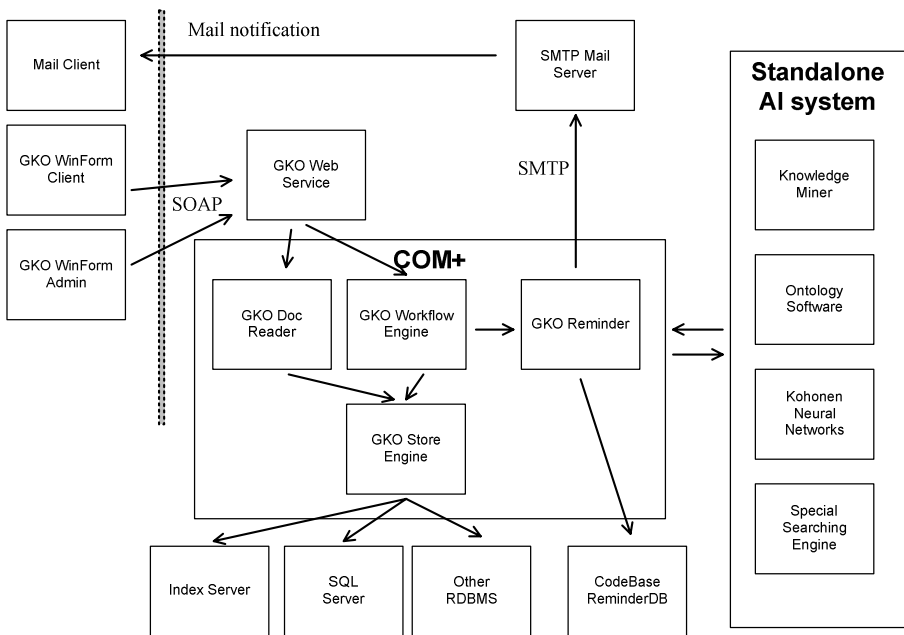


Fig. 1. The schematic view of the *GKO.NET*

flexible administration and specification of security rights and rules. It can be integrated with other systems. Multilevel data security and localization into selected languages are included. Thanks to its comfortable and easy-to-use graphical interface and low software and hardware requirements, *GKO.NET* is a convenient and affordable solution even for the smallest businesses. The overall scheme of the *GKO.NET* in the context of some other support systems is depicted in Figure 1. *Standalone AI system* communicates with business layer, which contains bussines logic (*Doc Reader, Workflow Engine, Store Engine, Reminder*).

5.4 How the Systems Works

System is supposed to manage existing documents of an institution in order to help the user to create new one. Figure 2 shows the process of extracting knowledge from various types of documents by using prepared ontologies and the *Knowledge System* (in the left side of the schema) and the process of creating new document and searching relevant knowledge from prepared knowledge base in the right part of the diagram. Process *Searching Relevant Knowledge* offers interesting information and knowledge as the parts of other documents for the new one. It needs extracted and indexed knowledge from existing documents, parsed by the ontology, and the knowledge software. The *New Document* item could be documentation, some agreements or a source code.

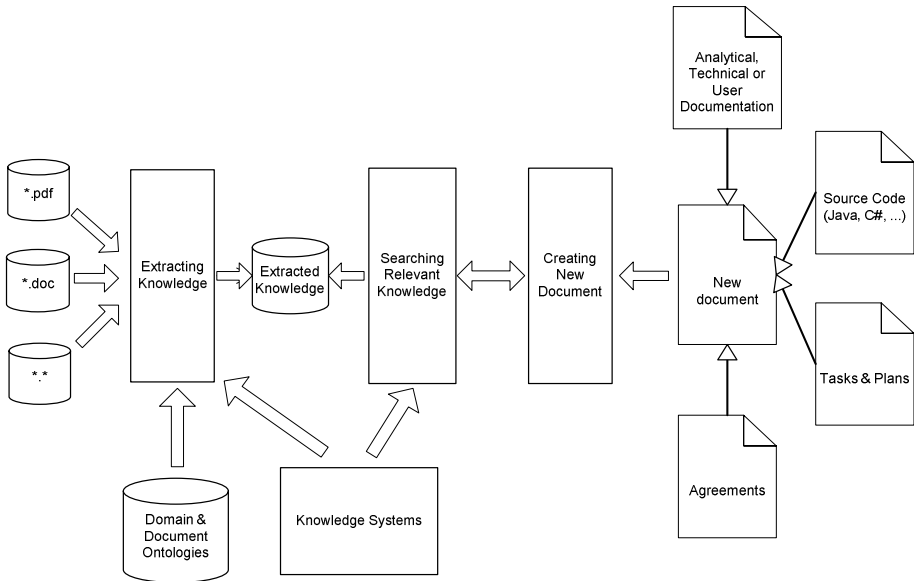


Fig. 2. The process of extracting knowledge and creating a new document

System needs to understand the structure of the workspace for searching the knowledge. Required structure is saved in the domain ontology. Thanks to its elements and the understanding of the knowledge management the system uses the appropriate rules to find other relevant documents.

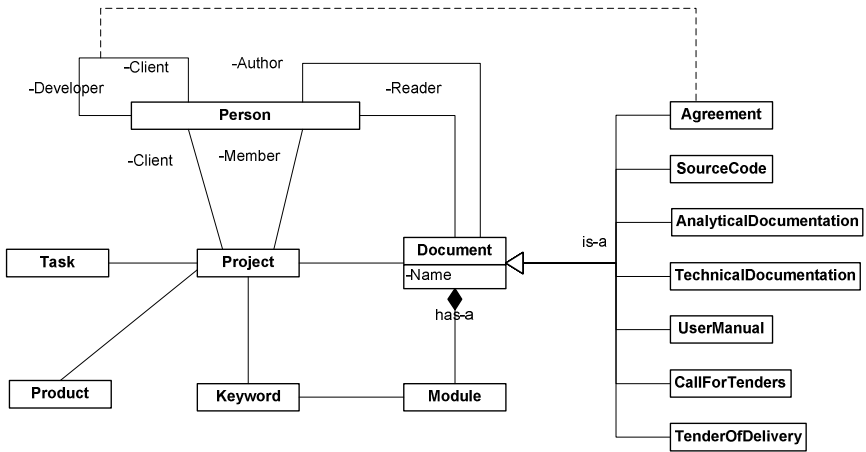


Fig. 3. A semantic model as a draft for a domain/document ontology

Figure 3 shows a draft of the semantic model as a basis for the domain and the document ontology. We can mention the relationship between a project, a document and some keywords. Therefore we could prepare searching knowledge with the keyword vectors using graph and clustering algorithms, the *Self Organizing Maps*, and mining associative rules.

Also we can find the relation between users, authors and the documents and prepared corresponding rules about the user history and authors of different artifacts for the knowledge system. Except the title and the authors, document contains modules (chapters, appendices, classes, packages, paragraphs). It is specialized to the independent types with their proper ontology (source code is quite different from the analytical document). In addition to the visual model of the ontology for the designers we need also the XML code for the parser.

An interesting feature of the system is the ability to serve in a many types of companies. *GKO* serves e.g. in Arca-capital (www.arcacapital.com/sk/), in Hornex (www.hornex.sk), Milking (www.milking.sk), in Elvea (www.elvea.sk), in Gratex International (in all the internal and economic processes of the organization), and also in Gratex SLA (in service level agreement) for Allianz. Our today experience is that at the beginning of the integration and implementation processes it seems to be important to identify needful entities with their state spaces and report criteria, data joins, and user filters. This is the way how to create the model of the company and monitor its life. For all users it is a need to create their own *entities* with *attributes* and their *State Spaces* with the special activities in the states. This is the first step to implement a particular knowledge management into a company. The next step is to implement input and output forms with their rules, triggers and criteria to report the actual situation in the company.

6 A General Conclusion

The historical process of distribution of activities related with the use of knowledge in expert activities leads to professional specialization of human beings in workshops first, then to specialization of activities in larger enterprises, like factories, the time of information technologies leads to the transferring some activities based on knowledge (some expert problem-solving activities) to computerized knowledge systems. In the just actual conditions of the beginning of 21st century when we have enough of knowledge about how to manage knowledge, but the task is more and more complicated and it is a real need to make the managing process faster and faster, the fulfilling of the requirements becomes to be a harder and harder job for human specialists. In such a situation, it is the time to start with research of possibilities and conditions, and then to developing of computerized knowledge managing systems.

Acknowledgement. Authors' research is partially supported by the Gratex International Corp., Bratislava, Slovakia. I. P.'s work was partially supported also by the Scientific Grant Agency of the Slovak Republic grant VG1/0508/09.

References

- [1] Drucker, P.F.: Management Challenges for the 21st Century. Elsevier, Boston (2007)
- [2] Foray, D.: The Economics of Knowledge. The MIT Press, Cambridge (2004)
- [3] Hewitt, C.: Description and Theoretical Analysis (Using Schema) of Planner (Memo No. 251). MIT Artificial Intelligence Laboratory, Cambridge (1972)
- [4] Kelemen, J., Hvorecký, J.: On knowledge, knowledge systems, and knowledge management. In: Proc.9th International Conference on Computational Intelligence and Informatics, pp. 27–35. Budapest Tech., Budapest (2008)
- [5] McElroy, M.W.: The New Knowledge Management. Elsevier, Amsterdam (2003)
- [6] Minsky, M.L.: A framework for representing knowledge. In: Winston, P.H. (ed.) The Psychology of Computer Vision. McGraw-Hill, New York (1975)
- [7] Newell, A., Simon, H.A.: Human Problem Solving. Prentice Hall, Englewood Cliffs (1972)
- [8] Schreiber, G., et al.: Knowledge Engineering and Management. The MIT Press, Cambridge (2000)
- [9] Stefik, M.: Introduction to Knowledge Systems. Morgan Kaufmann, San Francisco (1995)
- [10] Winston, P.H.: Artificial Intelligence. Addison-Wesley, Reading (1977)

Frequent Itemset Mining in Multirelational Databases

Aída Jiménez, Fernando Berzal, and Juan-Carlos Cubero

Dept. Computer Science and Artificial Intelligence,
ETSIIT - University of Granada, 18071, Granada, Spain
{aidajm, jc.cubero, fberzal}@decsai.ugr.es

Abstract. This paper proposes a new approach to mine multirelational databases. Our approach is based on the representation of a multirelational database as a set of trees. Tree mining techniques can then be applied to identify frequent patterns in this kind of databases. We propose two alternative schemes for representing a multirelational database as a set of trees. The frequent patterns that can be identified in such set of trees can be used as the basis for other multirelational data mining techniques, such as association rules, classification, or clustering.

1 Introduction

Data mining techniques have been developed to extract potentially useful information from databases. Classification, clustering, and association rules have been widely used. However, most existing techniques usually require all the interesting data to be in the same table.

Several alternatives have been proposed in the literature to handle with more than one table. There are algorithms that have been developed in order to explore tuples that, albeit in the same table, are somehow related [1] [2]. Other algorithms have been devised to extract information from multirelational databases, i.e., taking into account not only a single table but also the tables that are related to it [3]. For instance, these algorithms have been used for classification [4] and clustering [5] in multirelational databases.

In this paper, we propose two alternative representations for multirelational databases. Our representation schemes are based on trees, so that we can apply existing tree mining techniques to identify frequent patterns in multirelational databases. We also compare the proposed representation schemes in order to determine which one is better to use depending on the information we want to obtain from the database.

Our paper is organized as follows. We introduce some standard terms in Section 2. Section 3 presents two different schemes for representing multirelational databases using trees. We explain the kind of patterns that can be identified in the trees derived from a multirelational database in Section 4. We present some experimental results in Section 5 and, finally, we end our paper with some conclusions in Section 6.

2 Background

We will first review some basic concepts related to labeled trees before we address our multirelational data mining problem.

A **tree** is a connected and acyclic graph. A tree is rooted if its edges are directed and a special node, called root, can then be identified. The root is the node from which it is possible to reach all the other nodes in the tree. In contrast, a tree is said to be free if its edges have no direction, that is, when it is an undirected graph. A free tree, therefore, has no predefined root.

Rooted trees can be classified as **ordered trees**, when there is a predefined order within each set of sibling nodes, or **unordered trees**, when there is not such a predefined order among sibling nodes. textbfPartially-ordered trees contain both ordered and unordered sets of sibling nodes. They can be useful when the order within some sets of siblings is important but it is not necessary to establish an order relationship within all the sets of sibling nodes.

Figure 1 shows an example dataset with different kinds of rooted trees. In this figure, ordered sibling nodes are joined by an arc, while unordered sets of sibling nodes do not share an arc.

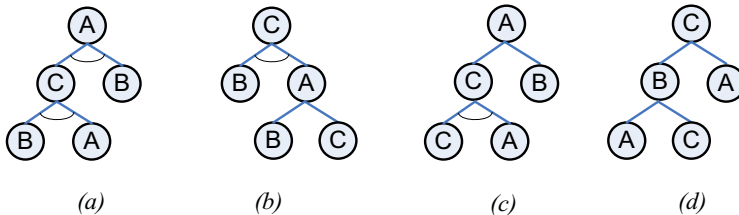


Fig. 1. Example dataset with different kinds of rooted trees (from left to right): (a) completely-ordered tree, (b) (c) partially-ordered trees, (d) completely-unordered tree

3 Tree-Based Multirelational Database Representation

Multirelational data mining techniques look for patterns that involve multiple relations (tables) from a relational database [3].

The schema of a multirelational database can be represented using an UML diagram [6]. Figure 2 shows the UML diagram for a multirelational database as well as its tables with some example tuples.

We will call target relation (or target table) to the main relation in the multirelational database. Let x be a relation, A be an attribute belonging to this relation, and a the value of the attribute. We will use the notation $x.A = a$ to represent the node which contains the value a for the attribute A in the relation x .

We present two different schemes for representing multirelational databases as sets of trees. The main idea behind both of them is building a tree from each tuple in the target table and following the links between tables (i.e. the foreign keys) to collect all the information related to each tuple in the target table. In both representation schemes, the root of all trees will be the name of the target table.

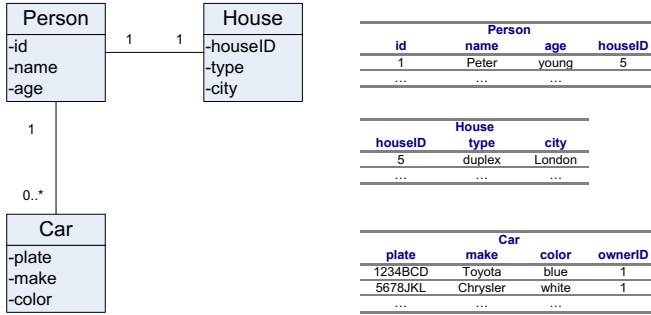


Fig. 2. MultiRelational database example

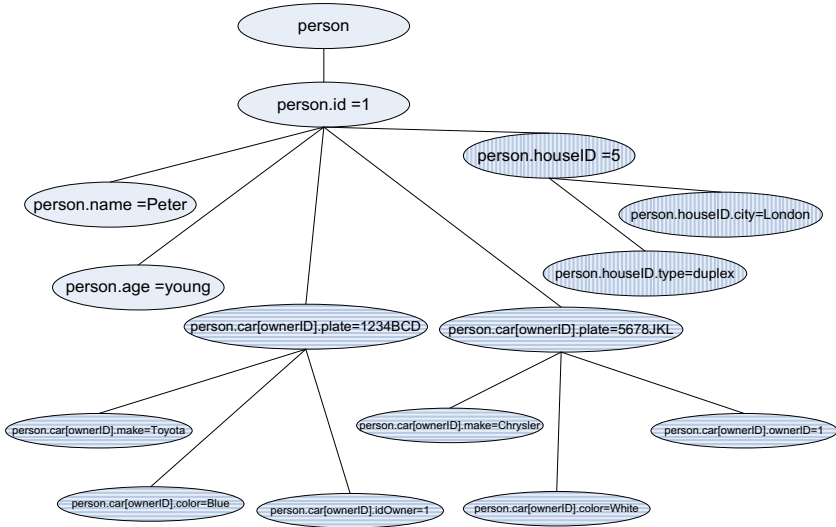
3.1 Key-Based Tree Representation

The key-based tree representation scheme represents all the attribute values within a tuple as children of its primary key.

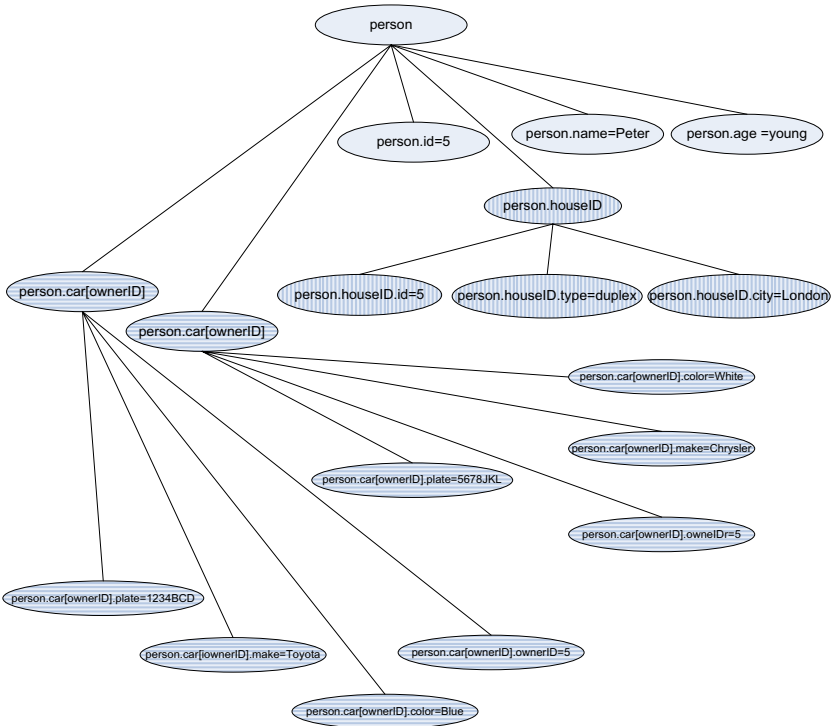
As a consequence, the root node of the tree representing a tuple in the target relation will have, as its unique child, the value of the primary key of the tuple that the tree represents. The children of this primary key node will be the remaining attribute values in the tuple.

The tree is then built by exploring the relationships between the target relation and the others relations in the database. Let x be the target table, $person$ in the example shown in Figure 2, and $x.K_x = k$ the node that represent the primary key, i.e. $person.id$. We can find two possible scenarios:

- When we have a one-to-one or many-to-one relationship between two relations, x and y , an attribute A of table x will be a foreign key pointing to the primary key of the table y . The attributes B of table y will be the children of $x.A = a$ in the tree and they will be represented as $x.A.B = b$. In our example, the relationship between Person and House is one-to-one and its representation is depicted in the Figure 3 a) using nodes with vertical lines in their background.
- When we have a one-to-many relationship, an attribute B of table y is a foreign key that refers to the primary key of table x . Many tuples in y may point to the same tuple in x . In this case, for each tuple in y that points to the same tuple in x , we create a new child of the x primary key node with the name of both tables, x and y , the attribute B that points to our target table, and the primary key of y with its value k_y using the notation $x.y[B].K_y = k_y$. This node will have, as children nodes, the remaining attributes in y using the notation $x.y[B].C = c$. In our example, the one-to-many relationship between Person and Car is shown in the nodes shaded with horizontal lines in Figure 3 a).



a) Key-based tree representation.



b) Object-based tree representation.

Fig. 3. Tree-based representation alternatives for the multirelational database shown in Figure 2

3.2 Object-Based Tree Representation

The object-based representation uses intermediate nodes as roots of the subtrees derived from the data in each table. In this representation, all the attribute values within a tuple will be in the same tree level.

If we have a single table, all the attribute values within the tuple (including the primary key) will be children of the root node in the tree representing the tuple.

- The case of one-to-one and many-to-one relationships is now addressed by adding the attributes of y as children of the node $x.A$ using the notation $x.A.B = b$. The nodes shaded with vertical lines illustrate this in Figure 3(b).
- When another table y has a foreign key B that refers to the target table x , a new node is built for each tuple in y that points to the same tuple in x . This node is labeled with the name of both tables and the attribute B involved in the relation, i.e. $x.y[B]$. Its children are all the attribute values of the tuple in y , i.e. $x.y[B].C = c$. The nodes with horizontal lines in Figure 3(b) show an example of this kind of relationship.

It should be noted that the object-based tree representation generates trees with more nodes than the key-based one. However, the tree depth is typically lower in the object-based tree representation than in the key-based one.

3.3 Deriving Trees from a Multirelational Database

Once we have presented two alternative schemes for the tree-based representation of multirelational databases, we have to establish how we traverse the relationships between the relations in our database to build the tree. In particular, we have to consider if it is interesting to go back through using a relationship that we have already represented in the tree.

In Figure 3, when we have represented the information about Peter and his cars, it is not necessary to go back through the person-car relationship because we would again obtain the information we already have in the tree.

However, if the target table were car, we would first represent the information of the Toyota car. Next, we would traverse the car-person relationship to obtain the information about the owner of the car (Peter). Finally, we would go back through the person-car relationship to represent all the cars that Peter owns.

Therefore, if we go through a one-to-many relationship from the relation with single cardinality, it is not necessary to go back. However, if we start from the table with multiple cardinality, we can go back through the same relationship to obtain more information.

4 Identifying Frequent Patterns in Multirelational Databases

The use of a tree-based representation for multirelational databases lets us apply tree mining techniques to identify the frequent patterns that are present in the

multirelational database. Several algorithms have been devised to identify tree patterns, including TreeMiner [7], SLEUTH [8], and POTMiner [9].

Different kinds of subtrees can be defined depending on the way we define the matching function between the pattern and the tree it derives from [10] (see Figure 4):

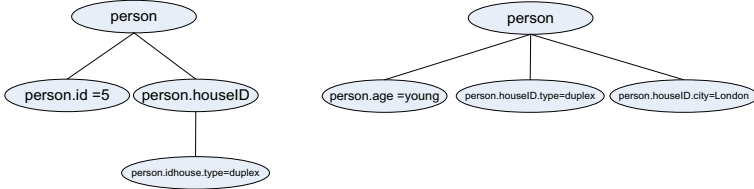


Fig. 4. An induced subtree (*left*) and an embedded subtree (*right*) from the tree shown in Figure 3 (*b*)

- A **bottom-up subtree** T' of T (with root v) can be obtained by taking one vertex v from T with all its descendants and their corresponding edges.
- An **induced subtree** T' can be obtained from a tree T by repeatedly removing leaf nodes from a bottom-up subtree of T .
- An **embedded subtree** T' can be obtained from a tree T by repeatedly removing nodes, provided that ancestor relationships among the vertices of T are not broken.

4.1 Induced and Embedded Patterns

Induced and embedded patterns give us different information about the multirelational database.

Induced patterns preserve the relationships among the nodes in the tree-based representation of the multirelational database. Induced patterns describe the database in great detail, in the sense that the patterns preserve the structure of the original trees in the database. However, identifying large patterns is often necessary to obtain useful information from the multirelational database. Unfortunately, this might involve a heavy computational effort.

If we use embedded patterns, some of the relationships among the nodes in the original trees are not preserved. In other words, we could obtain the same pattern from different tree-structures in the database. On the other hand, embedded patterns are typically smaller than the induced patterns required to represent the same information.

For example, Figure 5 shows some patterns identified from the object-based tree representation of the multirelational database in Figure 2. The induced pattern shown on the left tells us that some people in our database have a Toyota *and* a white car, while the induced pattern on the right of the Figure tells us that people in our database have a Toyota *that* is also white. In the case of the embedded pattern shown in the same Figure illustrates that some people

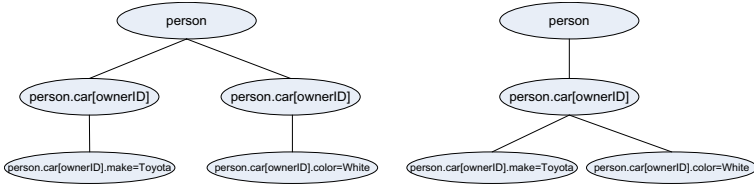
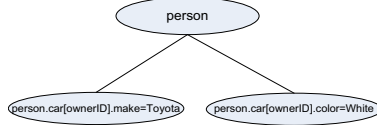
*Induced patterns**Embedded pattern*

Fig. 5. Embedded and induced patterns identified in the object-based tree representation of the multirelational database in Figure 2

have a Toyota car and a white car, but we do not know if it is the same car (a white Toyota) of they own two cars (a Toyota and a white car).

4.2 Key-Based and Object-Based Patterns

The key-based and the object-based tree representation schemes for multirelational databases also provide us different kinds of patterns.

Since intermediate primary key nodes from the target relation are not frequent in the tree database and induced patterns preserve all the nodes as in the original tree, no induced patterns starting at, or including, a primary key node from the target relation will be identified using the key-based representation scheme. However, it is possible to identify induced patterns starting at other key nodes because they may be frequent in the database.

When we use the object-based representation, induced patterns can be obtained with information about the target table. Therefore, the object-based representation is our only choice if we are interested in patterns that preserve the original structure of the trees in the database, i.e. induced patterns.

On the contrary, using the object-based representation to discover embedded patterns is useful only if we are interested in patterns like the one shown in Figure 6. That pattern indicates that Londoners with two cars are frequent, without any references to the car features. It should be noted that this kind of pattern cannot be identified using the key-based representation, since they always involve attribute values. In the object-based representation, however, the presence of intermediate nodes increases the number of identified patterns. Therefore, we should only resort to the object-based representation when we are interested in patterns like the one in Figure 6. Otherwise, the key-based representation provides faster results.

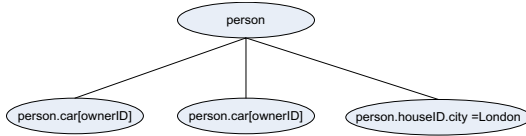


Fig. 6. An embedded subtree from the object-based tree in Figure 3(b)

5 Experimental Results

Once we have described our two tree representation schemes for multirelational databases, we discuss the experimental results we have obtained using both representation schemes. We have used the POTMiner algorithm [9] to identify induced and embedded subtrees from the trees representing the mutagenesis multirelational database, which can be downloaded from:

<http://www.ews.uiuc.edu/~xyin1/files/crossmine.html>.

Representation	Induced Subtrees					
	Support = 20%		Support = 10%		Support = 5%	
	Patterns	Time(ms)	Patterns	Time(ms)	Patterns	Time(ms)
Key-based	29	3040	29	3096	63	4315
Object-Based	4308	314378	6891	531263	14363	797126

Representation	Embedded Subtrees					
	Support = 20%		Support = 10%		Support = 5%	
	Patterns	Time(ms)	Patterns	Time(ms)	Patterns	Time(ms)
Key-based	4886	131185	8159	190484	16950	280758
Object-Based	19862	1783031	31143	1749505	63915	3034299

Fig. 7. Number of patterns and POTMiner execution time corresponding to the identification of induced and embedded patterns in the mutagenesis multirelational database using different minimum support thresholds

In our experiments, we have identified induced and embedded patterns including up to four nodes, i.e. MaxSize=4.

Figure 7 shows the number of discovered patterns and POTMiner execution time using different minimum support thresholds for the mutagenesis dataset, for both the key-based and the object-based tree representation schemes.

The number of discovered patterns using the object-based tree representation is larger than the number of identified patterns using the key-based representation. This is mainly due to the use of intermediate nodes, which are usually frequent, to represent objects.

It should also be noted that the number of induced patterns obtained from the key-based representation of the database is very small. This is due to the use of primary keys as internal nodes within the trees.

Figure 8 shows a comparison of the time required to identify induced and embedded patterns using both the key-based and the object-based representation schemes.

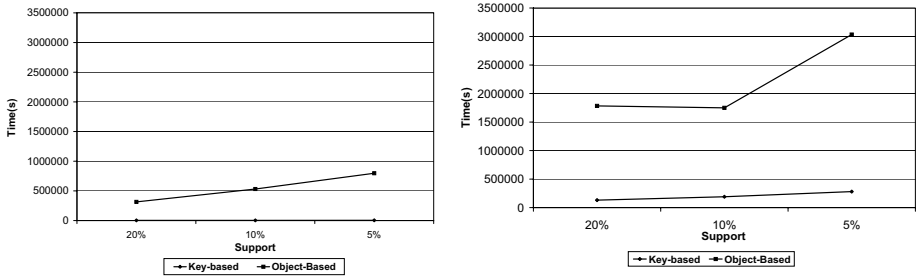


Fig. 8. POTMiner execution times when identifying induced patterns (left) and embedded patterns (right) in the mutagenesis database

Our algorithm execution time is proportional to the number of considered patterns. The execution time for discovering induced patterns is lower than the time needed to discover embedded patterns because a lower number of induced patterns are identified. Likewise, the object-based representation requires more execution time because a greater number of patterns are considered.

Albeit not shown in the Figure, it should also be noted that our algorithm is also linear with respect to the number of trees in the database.

6 Conclusions

This paper proposes a new approach to mine multirelational databases. Our approach is based on trees and we have designed two alternative tree representation schemes for multirelational databases. The main idea behind both of them is to build a tree representing each tuple in the target table by following the existing foreign keys that connect tables in the multirelational database.

The key-based representation scheme uses primary keys as intermediate nodes in the trees representing each tuple in the target relation. In contrast, the object-based representation scheme uses generic references as intermediate nodes to include new tuples in the trees.

We have identified frequent patterns in the trees representing the multirelational database and we have studied the differences that result from identifying induced or embedded patterns in the key-based or in the object-based representation scheme. These frequent patterns can be used to extract association rules from the multirelational database, build multirelational classification models, or develop multirelational clustering techniques.

Our experiments with an actual database show that our approach is feasible in practise. The discovery of induced patterns combined with the object-based representation scheme is often enough to describe a multirelational database in great detail. Embedded patterns, when used with the key-based representation scheme, let us reach data that is farther from the target table, although they might not preserve the structure of the original database trees.

Acknowledgements

Work partially supported by research project TIN2006-07262.

References

1. Tung, A.K.H., Lu, H., Han, J., Feng, L.: Efficient mining of intertransaction association rules. *IEEE Transaction on Knowledge and Data Engineering* 15(1), 43–56 (2003)
2. Lee, A.J.T., Wang, C.S.: An efficient algorithm for mining frequent inter-transaction patterns. *Inf. Sci.* 177(17), 3453–3476 (2007)
3. Džeroski, S.: Multi-relational data mining: An introduction. *SIGKDD Explorations Newsletter* 5(1), 1–16 (2003)
4. Yin, X., Han, J., Yang, J., Yu, P.S.: CrossMine: efficient classification across multiple database relations. In: *International Conference on Data Engineering*, pp. 399–410 (2004)
5. Yin, X., Han, J., Yu, P.S.: Cross-relational clustering with user’s guidance. In: *Knowledge Discovery and Data Mining*, pp. 344–353 (2005)
6. Booch, G., Rumbaugh, J., Jacobson, I.: *Unified Modeling Language User Guide*, 2nd edn. The Addison-Wesley Object Technology Series. Addison-Wesley Professional, Reading (2005)
7. Zaki, M.J.: Efficiently mining frequent trees in a forest: Algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering* 17(8), 1021–1035 (2005)
8. Zaki, M.J.: Efficiently mining frequent embedded unordered trees. *Fundamenta Informaticae* 66(1-2), 33–52 (2005)
9. Jimenez, A., Berzal, F., Cubero, J.C.: Mining induced and embedded subtrees in ordered, unordered, and partially-ordered trees. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) *Foundations of Intelligent Systems. LNCS (LNAI)*, vol. 4994, pp. 111–120. Springer, Heidelberg (2008)
10. Chi, Y., Muntz, R.R., Nijssen, S., Kok, J.N.: Frequent subtree mining - an overview. *Fundamenta Informaticae* 66(1-2), 161–198 (2005)

A Multiple Scanning Strategy for Entropy Based Discretization

Jerzy W. Grzymala-Busse^{1,2}

¹ Department of Electrical Engineering and Computer Science,
University of Kansas, Lawrence, KS 66045, USA

² Institute of Computer Science,
Polish Academy of Sciences, 01-237 Warsaw, Poland
jerzy@ku.edu

Abstract. We present results of experiments performed on 14 data sets with numerical attributes using a novel technique of discretization called multiple scanning. Multiple scanning is based on scanning all attributes of the data set many times, during each scan the best cut-points are found for all attributes. Results of our experiments show that multiple scanning enhances successfully, in terms of the error rate, an ordinary discretization technique based on conditional entropy.

Keywords: Rough sets, multiple scanning, entropy based discretization, LEM2 rule induction algorithm, LERS data mining system.

1 Introduction

Mining data sets with numerical attributes requires a special technique called discretization, i.e., converting numerical values into intervals [7]. Discretization is usually performed before the main process of knowledge acquisition. There are many techniques of discretization, however discretization based on conditional entropy is considered to be one of the most successful techniques [3,4,5,7,9,10,11,13,14]. Therefore, in this paper, we selected for our experiments a method of discretization based on conditional entropy [3,5].

Let us start from some fundamental ideas of discretization. For a numerical attribute a with an interval $[a, b]$ as a range, a partition of the range into k intervals

$$\{[a_0, a_1), [a_1, a_2), \dots, [a_{k-2}, a_{k-1}), [a_{k-1}, a_k]\},$$

where $a_0 = a$, $a_k = b$, and $a_i < a_{i+1}$ for $i = 0, 1, \dots, k-1$, defines a discretization of a . The numbers a_1, a_2, \dots, a_{k-1} are called *cut-points*. Our discretization system denotes such intervals as $a_0..a_1, a_1..a_2, \dots, a_{k-1}..a_k$.

Discretization methods in which attributes are processed one at a time are called *local* [3,7] (or *static* [4]). On the other hand, if all attributes are considered in selection of the best cut-point, the method is called *global* [3,7] (or *dynamic* [4]). Additionally, if information about the expert's classification of cases is taken into account during the process of discretization, the method is called *supervised* [4].

2 Entropy Based Discretization

An entropy of a variable v (attribute or decision) with values v_1, v_2, \dots, v_n is defined by the following formula

$$E_v(U) = - \sum_{i=1}^n p(v_i) \cdot \log p(v_i),$$

where U is the set of all cases in a data set and $p(v_i)$ is a probability (relative frequency) of value v_i in the set U , $i = 0, 1, \dots, n$. All logarithms in this paper are binary.

Table 1. An example of a data set with numerical attributes

Case	Attributes			Decision
	A	B	C	D
1	0.3	-0.2	12.2	a
2	0.3	0.4	12.2	b
3	0.3	0.4	14.6	c
4	1.1	-0.2	14.6	c
5	1.1	1.8	14.6	c
6	1.5	1.8	14.6	d
7	1.5	1.8	20.2	e

A conditional entropy of the decision d given an attribute a is

$$E(d|a) = - \sum_{j=1}^m p(a_j) \cdot \sum_{i=1}^n p(d_i|a_j) \cdot \log p(d_i|a_j),$$

where a_1, a_2, \dots, a_m are all values of a and d_1, d_2, \dots, d_n are all values of d . There are two fundamental criteria of quality based on entropy. The first is an *information gain* associated with an attribute a , denoted by $I(a)$, and equal to

$$E_d(U) - E(d|a)$$

the second is *information gain ratio*, for simplicity called *gain ratio*, defined by

$$\frac{I(a)}{E_a(U)}.$$

For a cut-point q for an attribute a the conditional entropy, defined by a cut-point q that splits the set U of all cases into two sets, S_1 and S_2 is defined as follows

$$E_a(q, U) = \frac{|S_1|}{|U|} E_a(S_1) + \frac{|S_2|}{|U|} E_a(S_2),$$

where $|X|$ denotes the cardinality of the set X . The cut-point q for which the conditional entropy $E_a(q, U)$ has the smallest value is selected as the best cut-point.

2.1 Starting from One Attribute

We will discuss two basic discretization techniques based on entropy. The first discretization technique is called *starting from one attribute*. Initially, we identify the best attribute (i.e., the attribute with the largest information gain or the attribute with the largest gain ratio). For the best attribute, we are looking for the best cut-point, i.e., the cut-point with the smallest conditional entropy. The best cut-point divides the data set into two smaller data sets, S_1 and S_2 . We apply the same strategy for both smaller data sets separately. However, we need to take into account that discretization of one of the smaller data sets may affect the other. We will illustrate this method by discretizing the data set from Table 1. We will use the information gain as the criterion to select the best attribute.

The conditional entropy $E(D|A)$ is

$$\frac{3}{7}(3)(-\frac{1}{3} \cdot \log \frac{1}{3}) + \frac{2}{7} \cdot 0 + \frac{2}{7}(2)(-\frac{1}{2} \cdot \log \frac{1}{2}) = 0.965.$$

Similarly, the conditional entropies $E(D|B) = 1.250$ and $E(D|C) = 0.749$. The minimal conditional entropy is associated with attribute C . The next question is what is the best cut-point for attribute C . This attribute has two potential cut-points (averages between sorted values of the attribute C): 13.4 and 17.4. The conditional entropy $E_C(13.4, U)$ is

$$\frac{2}{7}(2)(-\frac{1}{2} \cdot \log \frac{1}{2}) + \frac{5}{7}(-\frac{3}{5} \cdot \log \frac{3}{5} - \frac{1}{5}(2) \cdot \log \frac{1}{5}) = 1.265,$$

similarly, the conditional entropy $E_C(17.4, U) = 1.536$. Thus we will select the cut-point 13.4. Obviously, the current discretization of attribute C into two intervals 12.2..13.4 and 13.4..20.2 is not sufficient, since if we will use only discretized attribute C our data set will be inconsistent, i.e., there will be conflicting cases. The current discretization partitions Table 1 into two subtables, Tables 2 and 3.

Table 2. The first subtable of Table 1

Case	Attributes			Decision
	A	B	C	D
1	0.3	-0.2	12.2	a
2	0.3	0.4	12.2	b

Table 3. The second subtable of Table 1

Case	Attributes			Decision
	A	B	C	
3	0.3	0.4	14.6	c
4	1.1	-0.2	14.6	c
5	1.1	1.8	14.6	c
6	1.5	1.8	14.6	d
7	1.5	1.8	20.2	e

It is also obvious that for Table 2 the only attribute that may be discretized is B , with the cut-point equal to 0.1. Table 4 presents the current situation: discretized are attributes B and C , with cut-points 0.1 and 13.4, respectively.

Table 4. Table 1 with discretized attributes B and C once

Case	Attributes		Decision
	B	C	
1	-0.2..0.1	12.2..13.4	a
2	0.1..1.8	12.2..13.4	b
3	0.1..1.8	13.4..20.2	c
4	-0.2..0.1	13.4..20.2	c
5	0.1..1.8	13.4..20.2	c
6	0.1..1.8	13.4..20.2	d
7	0.1..1.8	13.4..20.2	e

Table 4 is not consistent, so Table 1 needs further discretization. However, by analysis of Table 4 we may easily discover that all what we need to do is to distinguish cases 3 and 5 from cases 6 and 7 and that cases 3 and 4 do not need to be distinguished. Thus, our next table to be discretized is presented as Table 5 (note that Table 5 is simpler than Table 3). We will continue discretization by recursion. Our final choice of cut-points is 1.3 for A , 0.1 for B , and 13.4 and 17.4 for C .

2.2 Multiple Scanning Strategy

The second discretization technique is based on scanning the set of attributes some fixed number of times and selecting for each attribute the best cut-point during each scan. After such scanning, if the discretized decision table needs more discretization, the first technique (starting from one attribute) is used. We will illustrate this technique by scanning all attributes, A , B , and C once. First

Table 5. Table that still need discretization

Case	Attributes			Decision
	A	B	C	D
3	0.3	0.4	14.6	c
5	1.1	1.8	14.6	c
6	1.5	1.8	14.6	d
7	1.5	1.8	20.2	e

we are searching for the best cut-point for attributes A , B , and C . The best cut-points are 1.3, 1.1, and 13.4, respectively. The discretized table is presented as Table 6.

Table 6. Table 1 discretized by scanning all attributes once

Case	Attributes			Decision
	A	B	C	D
1	0.3..1.3	-0.2..1.1	12.2..13.4	a
2	0.3..1.3	-0.2..1.1	12.2..13.4	b
3	0.3..1.3	-0.2..1.1	13.4..20.2	c
4	0.3..1.3	-0.2..1.1	13.4..20.2	c
5	0.3..1.3	1.1..1.8	13.4..20.2	c
6	1.3..1.5	1.1..1.8	13.4..20.2	d
7	1.3..1.5	1.1..1.8	13.4..20.2	e

Table 6 is not consistent, we need to distinguish cases 1 and 2, and, separately, cases 6 and 7. Therefore we need to use *starting from one attribute* technique for two tables, first with two cases, 1 and 2, and second with also two cases, 6 and 7. As a result we will select cut-points 0.1 and 17.4 for attributes B and C , respectively.

2.3 Stopping Condition for Discretization

In experiments discussed in this paper, the stopping condition was the level of consistency [3], based on rough set theory introduced by Z. Pawlak in [12]. Let U denote the set of all cases of the data set. Let P denote a nonempty subset of the set of all variables, i.e., attributes and a decision. Obviously, set P defines an equivalence relation φ on U , where two cases x and y from U belong to the same equivalence class of φ if and only if both x and y are characterized by the same values of each variable from P . The set of all equivalence classes of φ , i.e., a partition on U , will be denoted by P^* .

Equivalence classes of φ are called *elementary sets* of P . Any finite union of elementary sets of P is called a *definable set* in P . Let X be any subset of U . In general, X is not a definable set in P . However, set X may be approximated by two definable sets in P , the first one is called a *lower approximation of X in P* , denoted by $\underline{P}X$ and defined as follows

$$\bigcup\{Y \in P^* \mid Y \subseteq X\}.$$

The second set is called an *upper approximation of X in P* , denoted by $\overline{P}X$ and defined as follows

$$\bigcup\{Y \in P^* \mid Y \cap X \neq \emptyset\}.$$

The lower approximation of X in P is the greatest definable set in P , contained in X . The upper approximation of X in P is the least definable set in P containing X . A *rough set of X* is the family of all subsets of U having the same lower and the same upper approximations of X .

A *level of consistency* [3], denoted L_c , is defined as follows

$$L_c = \frac{\sum_{X \in \{d\}^*} |\underline{A}X|}{|U|}.$$

Practically, the requested level of consistency for discretization is 100%, i.e., we want the discretized data set to be *consistent*.

2.4 Interval Merging

The next step of discretization was merging intervals, to reduce their number and, at the same time, preserve consistency. Merging of intervals begins from *safe merging*, where, for each attribute, neighboring intervals labeled by the same decision value are replaced by their union. The next step of merging intervals was based on checking every pair of neighboring intervals whether their merging will result in preserving consistency. If so, intervals are merged permanently. If not, they are marked as un-mergeable. Obviously, the order in which pairs of intervals are selected affects the final outcome. In our experiments, we selected two neighboring intervals with the smallest total conditional entropy, taking all attributes into account. Using interval merging we may eliminate the cut-point 1.1 for attribute B , computed as a result of scanning Table 1 once.

2.5 Rule Induction: LEM2 Algorithm

The data system LERS (Learning from Examples based on Rough Sets) [6] induces rules from incomplete data, i.e., data with missing attribute values, from data with numerical attributes, and from inconsistent data, i.e., data with conflicting cases. Two cases are conflicting when they are characterized by the same values of all attributes, but they belong to different concepts (classes). LERS uses rough set theory to compute lower and upper approximations for concepts involved in conflicts with other concepts [12].

Rules induced from the lower approximation of the concept *certainly* describe the concept, hence such rules are called *certain*. On the other hand, rules induced from the upper approximation of the concept describe the concept *possibly*, so these rules are called *possible*.

The LEM2 algorithm (Learning from Examples Module, version 2) of LERS is most frequently used for rule induction. LEM2 explores the search space of attribute-value pairs. Its input data set is a lower or upper approximation of a concept, so its input data set is always consistent. In general, LEM2 computes a local covering and then converts it into a rule set [26]. Recently, a new, improved version of LEM2, called MLEM2, was developed [8].

Table 7. Data sets

Data set	Number of		
	cases	attributes	concepts
Australian	690	14	2
Bankruptcy	66	5	2
Bupa	345	6	2
Connectionist Bench	208	60	2
Echocardiogram	74	7	2
Ecoli	336	8	8
Glass	214	9	6
Image Segmentation	210	19	7
Ionosphere	351	34	2
Iris	150	4	3
Pima	768	8	2
Wave	512	21	3
Wine	178	13	3
Yeast	1484	8	9

3 Experiments

Our experiments were conducted on 14 data sets, summarized in Table 7. All of these data sets, with the exception of *bankruptcy*, are available on the University of California at Irvine *Machine Learning Repository*. The bankruptcy data set is a well-known data set used by E. Altman to predict a bankruptcy of companies.

Every discretization method was applied to every data set, with the level of consistency equal to 100%. For a choice of the best attribute, we used gain ratio. Rule sets were induced using the LEM2 algorithm of the LERS data mining system.

Table 8 presents results of ten-fold cross validation, for all 14 data sets, using increasing number of scans. Obviously, for any data set, after some fixed number of scans, an error rate is stable (constant). For example, for *Australian* data set, the error rate will be 15.65% for the scan number 4, 5, etc. Thus, and data set from Table 8 is characterized by two error rates: minimal and stable. For a given data set, the smallest error rate from Table 8 will be called *minimal* and the last entry in the row that corresponds to the data set will be called *stable*. For example, for the *Australian* data set, the minimal error rate is 14.93% and the stable error rate is 15.65%. For some data sets (e.g., for *bankruptcy*), minimal and stable error rates are identical.

Table 8. Error rates for discretized data sets

Data set	Error rate for scan number						
	0	1	2	3	4	5	6
Australian	34.49	15.22	14.93	15.65			
Bankruptcy	3.03	9.09	1.52				
Bupa	31.30	29.28	30.14	26.67			
Connectionist Bench	29.33	27.88					
Echocardiogram	24.32	16.22					
Ecoli	19.64	20.54	18.75	20.83	21.43	20.54	20.83
Glass	24.77	34.58	20.56	25.70	24.77	25.70	26.64
Image Segmentation	29.52	19.52	16.19	17.14			
Ionosphere	10.83	6.27	9.69	7.12			
Iris	5.33	2.67	4.67				
Pima	27.21	26.04	25.65	26.30	26.82	26.69	26.43
Wave	27.10	19.53	20.70	19.53	24.77	19.53	
Wine	11.24	2.81					
Yeast	56.74	50.47	48.99	48.92	51.28	52.83	

It is clear from Table 8 that the minimal error rate is never associated with 0 scans (i.e., with the method *starting from one attribute*). Using the Wilcoxon matched-pairs signed-ranks test, we conclude that the following two statements are statistically highly significant (i.e., the significance level is equal to 1% for a two-tail test):

- the minimal error rate is associated with scanning the entire attribute set at least once,
- the stable error rate is smaller than the error rate associated with the *starting from one attribute* discretization technique.

Additionally, effects of scanning during discretization are presented in Table 9. Note that some data sets, e.g., *Australian*, have binary attributes. For such data

Table 9. Number of intervals for scanning data set *bankruptcy*

Attribute	Number of scans					
	0		1		2	
	before merging	after merging	before merging	after merging	before merging	after merging
a1	9	9	4	3	3	2
a2	1	1	2	2	3	2
a3	2	2	2	1	3	2
a4	1	1	2	2	2	2
a5	1	1	2	2	2	1

sets, scanning will not change the number of intervals for binary attributes. We selected *bankruptcy* data set not only because its all attributes are numerical with real numbers as values but also since it has only five attributes. For 0 scans, i.e., for *starting from one attribute*, it is clear that attribute *a1* was selected as the best and that during discretization eight cut-points were selected. After the single scan, the same attribute was selected as the best attribute, hence two additional cut-points were selected for *a1*. With two scans, for the first three attributes two cut-points were selected, as expected, for the last two attributes, *a4* and *a5*, only single cut-points were found since the discretized table was already consistent, second cut-points for *a4* and *a5* would be redundant.

4 Conclusions

Our paper presents results of experiments in which scanning was used during discretization of 14 data sets with numerical attributes. Our discretization techniques were combined with rule induction using the LEM2 rule induction algorithm. As a result, we conclude that results of discretization based on scanning the attribute set at least once are significantly better (with a significance level of 1%, two-tailed test) than the results of discretization based on starting from one attribute. Thus, we proved that there exists an additional technique for improving discretization.

References

1. Blajdo, P., Grzymala-Busse, J.W., Hippe, Z.S., Knap, M., Mroczek, T., Piatek, L.: A comparison of six approaches to discretization—A rough set perspective. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 31–38. Springer, Heidelberg (2008)
2. Chan, C.C., Grzymala-Busse, J.W.: On the attribute redundancy and the learning programs ID3, PRISM, and LEM2. Department of Computer Science, University of Kansas, TR-91-14 (1991)

3. Chmielewski, M.R., Grzymala-Busse, J.W.: Global discretization of continuous attributes as preprocessing for machine learning. *Int. Journal of Approximate Reasoning* 15, 319–331 (1996)
4. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: 12th International Conference on Machine Learning, pp. 194–202. Morgan Kaufmann Publishers, San Francisco (1995)
5. Fayyad, U.M., Irani, K.B.: On the handling of continuous-valued attributes in decision tree generation. *Machine Learning* 8, 87–102 (1992)
6. Grzymala-Busse, J.W.: A new version of the rule induction system LERS. *Fundamenta Informaticae* 31, 27–39 (1997)
7. Grzymala-Busse, J.W.: Discretization of numerical attributes. In: Klösgen, W., Zytkow, J. (eds.) *Handbook of Data Mining and Knowledge Discovery*, pp. 218–225. Oxford University Press, New York (2002)
8. Grzymala-Busse, J.W.: MLEM2: A new algorithm for rule induction from imperfect data. In: 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, pp. 243–250. ESIA Annecy, France (2002)
9. Grzymala-Busse, J.W., Stefanowski, J.: Three discretization methods for rule induction. *Int. Journal of Intelligent Systems* 16, 29–38 (2001)
10. Liu, H., Hussain, F., Tan, C.L., Dash, M.: Discretization: An enabling technique. *Data Mining and Knowledge Discovery* 6, 393–423 (2002)
11. Nguyen, H.S., Nguyen, S.H.: Discretization methods for data mining. In: Polkowski, L., Skowron, A. (eds.) *Rough Sets in Knowledge Discovery*, pp. 451–482. Physica-Verlag, Heidelberg (1998)
12. Pawlak, Z.: Rough Sets. *International Journal of Computer and Information Sciences* 11, 341–356 (1982)
13. Pensa, R.G., Leschi, C., Besson, J., Boulicaut, J.F.: Assessment of discretization techniques for relevant pattern discovery from gene expression data. In: Proc. of the 4th ACM SIGKDD Workshop on Data Mining in Bioinformatics, pp. 24–30 (2004)
14. Stefanowski, J.: Handling continuous attributes in discovery of strong decision rules. In: Polkowski, L., Skowron, A. (eds.) *RSCTC 1998. LNCS (LNAI)*, vol. 1424, pp. 394–401. Springer, Heidelberg (1998)

Fast Subgroup Discovery for Continuous Target Concepts

Martin Atzmueller and Florian Lemmerich

University of Würzburg,
Department of Computer Science VI
Am Hubland, 97074 Würzburg, Germany
{atzmueller, lemmerich}@informatik.uni-wuerzburg.de

Abstract. Subgroup discovery is a flexible data mining method for a broad range of applications. It considers a given property of interest (target concept), and aims to discover interesting subgroups with respect to this concept. In this paper, we especially focus on the handling of continuous target variables and describe an approach for fast and efficient subgroup discovery for such target concepts. We propose novel formalizations of effective pruning strategies for reducing the search space, and we present the SD-Map* algorithm that enables fast subgroup discovery for continuous target concepts. The approach is evaluated using real-world data from the industrial domain.

1 Introduction

Subgroup discovery is a general knowledge discovery and data mining method that can be customized for various application scenarios. Prominent examples include knowledge discovery in medical and technical domains, e.g., [1,2,3,4]. Subgroup discovery is an undirected method for the identification of groups of individuals that deviate from the norm considering a certain property of interest [5], that is, a certain target concept. For example, the risk of coronary heart disease (target variable) is significantly higher in the subgroup of smokers with a positive family history than in the general population.

In this context, continuous target concepts have received increasing attention recently, e.g., [4,6], especially regarding industrial applications. For example, we could investigate whether certain combinations of factors cause an increased repair and/or scrap rate in a manufacturing scenario.

Many existing approaches, e.g., [2,5] require the discretization of continuous target attributes – with a significant loss of information. Therefore, this paper describes methods for fast and effective subgroup discovery for continuous target concepts and presents an efficient algorithm for this purpose. We first focus on pruning strategies for reducing the search space utilizing optimistic estimate functions for obtaining upper bounds for the possible quality of the discovered patterns. Specifically, we focus on the recently introduced notion of *tight optimistic estimate* functions for the case of continuous target concepts. Additionally, we show how an efficient method for subgroup discovery can be combined with the proposed pruning strategy, and present the SD-Map* algorithm as a novel adaptation of the efficient SD-Map [7] algorithm.

The rest of the paper is organized as follows: Section 2 provides the basics on subgroup discovery. After that, Section 3 introduces the challenges of handling continuous target concepts and proposes novel implementations of the tight optimistic estimate functions. Next, we propose the SD-Map* algorithm as an adaptation of the efficient SD-Map subgroup discovery method, and discuss related work. Section 4 provides an evaluation of the approach using real-world data from an exemplary industrial application. Finally, Section 5 concludes with a summary of the presented work and provides pointers for future work.

2 Preliminaries

In the following, we first introduce the necessary notions concerning the used knowledge representation, before we introduce subgroup discovery and its implementation using continuous target concepts.

2.1 Subgroup Discovery

The main application areas of subgroup discovery are exploration and descriptive induction to obtain an overview of the relations between a (dependent) target variable and a set of explaining (independent) variables. Then, the goal is to uncover properties of the selected target population of individuals featuring the given target property of interest. Specifically, these interesting subgroups should have the most unusual (distributional) characteristics with respect to the concept of interest given by the target variable [5].

A subgroup discovery task mainly relies on the following four main properties: the target variable, the subgroup description language, the quality function, and the discovery strategy. Since the search space is exponential concerning all the possible selectors of a subgroup description efficient discovery methods are necessary, e.g., beam-search or the exhaustive SD-Map algorithm [7].

First, let us introduce some basic notions: Let Ω_A denote the set of all attributes. For each attribute $a \in \Omega_A$ a range $dom(a)$ of values is defined. Let CB be the case base (data set) containing all available cases (instances). A case $c \in CB$ is given by the n-tuple $c = ((a_1 = v_1), (a_2 = v_2), \dots, (a_n = v_n))$ of $n = |\Omega_A|$ attribute values, $v_i \in dom(a_i)$ for each a_i .

The subgroup description language specifies the individuals belonging to the subgroup. For a commonly applied single-relational propositional language a subgroup description can be defined as follows:

Definition 1 (Subgroup Description). *A subgroup description $sd(s) = \{e_1, \dots, e_n\}$ of the subgroup s is defined by the conjunction of a set of selection expressions (selectors). The individual selectors $e_i = (a_i, V_i)$ are selections on domains of attributes, $a_i \in \Omega_A, V_i \subseteq dom(a_i)$. We define Ω_E as the set of all selection expressions and Ω_{sd} as the set of all possible subgroup descriptions.*

A subgroup s described by $sd(s)$ is given by all cases $c \in CB$ covered by the subgroup description $sd(s)$. A subgroup s' is called a *refinement* of s , if $sd(s) \subset sd(s')$.

2.2 Subgroup Quality Functions

A quality function measures the interestingness of the subgroup and is used to rank these. Typical quality criteria include the difference in the distribution of the target variable concerning the subgroup and the general population, and the subgroup size.

Definition 2 (Quality Function). *Given a particular target variable $t \in \Omega_E$, a quality function $q : \Omega_{sd} \times \Omega_E \rightarrow R$ is used in order to evaluate a subgroup description $sd \in \Omega_{sd}$, and to rank the discovered subgroups during search.*

For binary target variables, examples for quality functions are given by

$$q_{WRACC} = \frac{n}{N} \cdot (p - p_0), \quad q_{PS} = n \cdot (p - p_0), \quad q_{LIFT} = \frac{p}{p_0}, n \geq \mathcal{T}_n$$

where p is the relative frequency of the target variable in the subgroup, p_0 is the relative frequency of the target variable in the total population, $N = |CB|$ is the size of the total population, n denotes the size of the subgroup, and \mathcal{T}_n specifies a minimal size constraint for the subgroup. As discussed in [2] q_{WRACC} (weighted relative accuracy) trades off the increase in the target share p vs. the generality (n) of the subgroup. The Piatetsky-Shapiro, e.g., [9] quality function q_{PS} is a variation of q_{WRACC} without considering the size of the total population. Finally, the quality function q_{LIFT} focuses on the decrease/increase of the target share.

The main difference between the different types of target variables is given by disjoint sets of applicable quality functions, due to the different parameters that can be applied for estimating the subgroup quality: Target *shares* are only applicable for binary or categorical attributes, while continuous target variables require averages/aggregations of values, e.g., the *mean*. As equivalents to the quality functions for binary targets discussed above, we consider the functions *Continuous Piatetsky-Shapiro* (q_{CPS}), *Continuous LIFT* (q_{CLIFT}), and *Continuous Weighted Relative Accuracy* (q_{CWRACC}):

$$q_{CWRACC} = \frac{n}{N} \cdot (m - m_0), \quad q_{CPS} = n \cdot (m - m_0), \quad q_{CLIFT} = \frac{m}{m_0}, n \geq \mathcal{T}_n$$

where n and N denote the size of the subgroup and the size of the total population as defined above, respectively, and m specifies the mean of the target variable within the subgroup; m_0 specifies the mean of the target variable in the total population.

The *CN2-SD* algorithm [2], is a prominent example of an heuristic subgroup discovery algorithm that applies a beam-search strategy. The adaption of such an algorithm is rather simple, as in each step the quality values of the subgroup hypotheses contained in the beam are directly updated from the case base. Instead of determining the target share(s) of (binary) target variables, simply the mean values of the cases contained in the subgroup m and (once) for the total population m_0 need to be obtained. It is easy to see that the continuous case subsumes the binary one as a special case: Computing the averages includes computing the target shares – by considering the values 1 and 0 for a *true/false* target concept, respectively. The ordinal case can be captured by mapping the ordinal values to continuous values and normalizing these if necessary.

3 Adapting Subgroup Discovery for Continuous Target Concepts

In the following, we show how to efficiently adapt exhaustive subgroup discovery for continuous target concepts. We discuss tight optimistic estimate quality functions [9], and we introduce novel formalizations for the case of continuous target variables. Additionally, we describe how the efficient subgroup discovery algorithm SD-Map [7] can be combined with pruning measures using tight optimistic estimate functions resulting in the novel SD-Map* algorithm.

3.1 Tight Optimistic Estimates

The basic principle of optimistic estimates [9] is to safely prune parts of the search space. This idea relies on the intuition that if the k best hypotheses so far have already been obtained, and the optimistic estimate of the current subgroup is below the quality of the worst subgroup contained in the k best, then the current branch of the search tree can be safely pruned. More formally, an optimistic estimate oe of a quality function qf is a function such that $s' \subseteq s \Rightarrow oe(s) > qf(s')$, i.e., that no refinement of subgroup s can exceed the quality $oe(s)$. An optimistic estimate is considered *tight* if for any database and any subgroup s , there exists a subset $s' \subseteq s$, such that $oe(s) = qf(s')$. While this definition requires the existence of a subset of s , there is not necessarily a subgroup description, that describes s' , cf., [9].

For binary targets the determination of such a best subset is relatively simple using any quality function that is constructed according to the axioms postulated in [8]. The best subset is always given by the set of all cases, for which the target concept is true.

We introduce the following notation: $n(s) = |\{c \in s\}|$ specifies the size of subgroup s , $tp(s) = |\{c \in s | t(c) = true\}|$ the number of positive examples in s ; $t(c)$ denotes the value of the target variable in case c and $p(s) = \frac{tp(s)}{n(s)}$ is the target share of the subgroup.

Theorem 1. *For each subgroup s with $p > p_0$ and for each boolean quality function q for which the axioms postulated in [8] apply: $s' \subseteq s \Rightarrow q(s') \leq q(s^*)$, where $s^* = \{c \in s | t(c) = true\}$*

Proof. We first show, that $q(s) \leq q(s^*)$. This means, that the quality of any subgroup with a positive quality is always lower or equal to the quality of the subset of examples, that only contains the positive examples of s . We apply the third axiom of [8]: “ $q(s)$ monotonically decreases in n , when $p = c/n$ with a fixed constant c .”: As fixed constant c we consider the number of positive examples tp , as $p = c/n \Leftrightarrow c = p \cdot n$ and $n(s) \cdot p(s) = tp(s) = tp(s^*) = n(s^*) \cdot p(s^*)$. So, the quality function monotonically decreases in n . As $n(s) > n(s^*)$ we conclude: $q(s) \leq q(s^*)$.

For arbitrary $s' \subseteq s$ we now need to consider two cases: If $s^* \subseteq s'$, then $q(s') \leq q(s^*)$ as shown above. If $s^* \not\subseteq s'$, then there exists a subset $s'' = \{c \in s | t(c) = true\}$, that contains only the positive examples of s' . The above proof then implies: $q(s') \leq q(s'')$. On the other hand $s'' \subseteq s^*$ is true, as s'' only consists of positive examples of s . We now apply the fourth axiom of [8]: “ $q(s)$ monotonically increases in n when $p > p_0$ is fixed.”: As $p(s'') = p(s^*) = 1$ it follows that $q(s'') \leq q(s^*)$. Thus, $q(s') \leq q(s'') \leq q(s^*)$, proving the theorem. \square

In contrast to the binary case, the continuous one is more challenging, since the best refinement of a subgroup is dependent on the used quality function: Given an average target value of $m_0 = 50$ and the subgroup s containing cases with values 20, 80 and 90. Then, for the quality function lift with $\mathcal{T}_n = 1$ the subset with the best quality contains only the case with value 90. On the other hand for the Pietatsky-Shapiro quality function, it contains two cases with the values 80 and 90, respectively.

Theorem 2. For the Pietatsky-Shapiro quality function $q_{CPS}(s) = n \cdot (m - m_0)$ the tight optimistic estimate for any subgroup s is given by

$$oe(s) = \sum_{c \in s, t(c) > 0} (t(c) - m_0).$$

Proof. We reformulate the Pietatsky-Shapiro quality function:

$$\begin{aligned} q_{CPS}(s) &= n \cdot (m - m_0) \\ &= n \cdot \left(\frac{\sum_{c \in s} t(c)}{n} - \frac{n \cdot m_0}{n} \right) \\ &= \sum_{c \in s} t(c) - n \cdot m_0 \\ &= \sum_{c \in s} (t(c) - m_0) \end{aligned}$$

For all subsets $s' \subseteq s$, this sum reaches its maximum for the subgroup s^* , that contains all cases with larger target values than the average of the population, since it contains only positive summands, but no negatives. The quality of s^* is given by $q_{CPS}(s^*) = oe(s)$ using the above formula. As no other subset of s can exceed this quality the $oe(s)$ is an optimistic estimate. Since for any given subgroup the estimate is reached by one of its subsets, the estimate is tight. \square

Please note, that the tight optimistic estimate for the binary case provided in [9], i.e., $np(1 - p_0)$, can be seen as special case of this formula, considering $t(c) = 1$ for *true* target concepts and $t(c) = 0$ for *false* target concepts:

$$\begin{aligned} oe(s) &= \sum_{c \in s, t(c) > 0} (t(c) - m_0) \\ &= \sum_{c \in s, t(c) = 1} (1 - p_0) \\ &= np(1 - p_0). \end{aligned}$$

Theorem 3. Considering the quality function Weighted Relative Accuracy $q_{CWRACC}(s) = \frac{n}{N} \cdot (m - m_0)$ the tight optimistic estimate for any subgroup s is given by

$$oe(s) = \frac{1}{N} \sum_{c \in s, t(c) > 0} (t(c) - m_0),$$

where $t(c)$ is the value of the target variable for the case c .

Proof. q_{CWRACC} differs by the factor $\frac{1}{N}$ from the Pietatsky-Shapiro function. The population size can be considered as a constant, so the proof proceeds analogously. \square

Theorem 4. *For the quality function Lift with a minimum subgroup size T_n the optimistic estimate is given by $oe(s) = \sum_{i=1}^{T_n} (v_i - m_0)$, where v_i is the value i -th highest in the subgroup in respect of the target variable.*

Proof. Since the size of the subgroup is not relevant for these quality functions, the best possible subset is always the subset with the highest average of the target attribute with size k . The quality of this subset is given by the above formula. \square

3.2 Efficient Subgroup Discovery with SD-Map*

SD-Map [7] is based on the efficient FP-growth [10] algorithm for mining frequent patterns. As a special data structure, the frequent pattern tree or FP-tree is used which is implemented as an extended prefix-tree-structure that stores count information about the frequent patterns. FP-growth applies a divide and conquer method, first mining frequent patterns containing one selector and then recursively mining patterns of size 1 conditioned on the occurrence of a (prefix) 1-selector. For the recursive step, a conditional FP-tree is constructed, given the conditional pattern base of a frequent selector (node). Due to the limited space we refer to Han et al. [10] for more details.

SD-Map utilizes the FP-tree structure built in one database pass to efficiently compute quality functions for all subgroups. For the binary case, an FP-tree node stores the subgroup size and the true positive count of the respective subgroup description. In the case of a continuous target variable, we consider the sum of values of the target variable, enabling us to compute the respective quality functions value accordingly. Therefore, all the necessary information is locally available in the FP-tree structure. Please note, that the adaptations for numeric target variables includes the case of a binary variable as a special case, where the value of the target variable is 1, if the target concept is *true* and 0, otherwise.

SD-Map* extends SD-Map by including (optional) pruning strategies and utilizes quality functions with tight optimistic estimates for this purpose: For embedding (tight) optimistic estimate pruning into the SD-Map algorithm, we basically only need to consider three options for pruning and reordering/sorting according to the current (tight) optimistic estimates: (1) **Pruning:** In the recursive step when building a conditional FP-tree, we omit a (conditioned) branch, if the optimistic estimate for the conditioning selector is below the threshold given by the k best subgroup qualities. (2) **Pruning:** When building a (conditional) frequent pattern tree, we can omit all the nodes with an optimistic estimate below the mentioned quality threshold. (3) **Reordering/Sorting:** During the iteration on the currently active selector queue when processing a (conditional) FP-tree, we can dynamically reorder the selectors that have not been evaluated so far by their optimistic estimate value. In this way, we evaluate the *more promising* selectors first. This heuristic can help to obtain higher values for the pruning threshold early in the process, a way to prune more often earlier. Additionally, this step implements a modified depth-first search guided by the current optimistic estimates.

To efficiently compute the (tight) optimistic estimates we store additional information in the nodes of the FP-Tree, depending on the used quality function. For example,

for the Piatetsky-Shapiro quality function we add the value $\max(0, t(c) - p_0)$ for each case c to a field in the respective node during the construction of the FP-Tree. This field can also be propagated recursively – analogously to the sum of target values when building the conditional trees. It directly reflects the optimistic estimate of each node and can be immediately evaluated whenever it is needed for pruning.

3.3 Related Work and Discussion

In this paper, we propose novel formalizations of tight optimistic estimates for numeric quality functions. Additionally, we present the SD-Map* algorithm that enables efficient subgroup discovery for continuous target concepts. By utilizing these novel quality functions, SD-Map* shows a significant decrease in the number of examined states of the search space, and therefore also a significant reduction concerning the runtime and space requirements of the algorithm, as shown in the evaluation in Section 4. The techniques were implemented in the data mining environment VIKAMINE, and evaluation showed its benefit for the intended application in the industrial domain.

Handling numeric target concepts in the context of subgroup discovery has been first discussed by Kloesgen [8,11] in the EXPLORA system. Kloesgen applied both heuristic and exhaustive subgroup discovery strategies without pruning. An improvement was proposed by Wrobel [5], presenting optimistic estimate functions for binary target variables. Recently, Grosskreutz et al. [9] introduced tight optimistic estimate quality functions as a further improvement on optimistic estimate quality functions for binary and nominal target variables.

Jorge et al. [4] introduced an approach for subgroup discovery with continuous target concepts applying special visualization techniques for the interactive discovery step. In contrast to the presented approach, the methods focus on a different rule representation, i.e., distribution rules, not on deviations of the target concept averages in the subgroup vs. the total population. Furthermore, an adapted standard algorithm for discovering frequent sets is applied, so there are no pruning options for enabling a more efficient discovery process.

Grosskreutz et al. [12] proposed the DpSubgroup algorithm that also incorporates tight optimistic estimate pruning. While their algorithm is somehow similar to the SD-Map algorithm, since also a frequent pattern tree is used for efficiently obtaining the subgroup counts, the DpSubgroup algorithm focuses on binary and categorical target concepts only, and lacks the efficient propagation method of SD-Map* when computing the tight optimistic estimates in the FP-tree. In contrast, SD-Map* is applicable for binary, categorical, and continuous target concepts. Additionally, DpSubgroup uses an explicit depth-first search step for evaluating the subgroup hypotheses while this step is implicitly included in the divide-and-conquer frequent pattern growth method of SD-Map* directly (that is, by the reordering/sorting optimization).

4 Evaluation

In the following, we outline the application scenario of the presented approach. After that, we present an evaluation using real-world data, and show, how the proposed approach helps to prune the search space significantly.

4.1 Application Scenario

The development of the presented techniques was mainly motivated by industrial applications in the service support and in the manufacturing domain that required fast responsiveness for interactive scenarios. In general, industrial applications of subgroup discovery often require the utilization of continuous parameters, for example, certain measurements of machines or production conditions. Then, the target concepts can often not be analyzed sufficiently using the standard techniques for binary/nominal subgroup discovery, since the discretization of the variables causes a loss of information. As a consequence, the interpretation of the results is often difficult.

Therefore, the presented techniques provide a robust alternative: In our applications, one important goal was the identification of subgroups (as combination of certain factors) that cause a significant increase/decrease in certain parameters, for example, the number of service requests for a certain technical component, or the fault/repair rate of a certain manufactured product. For a comprehensive analysis, the presented pruning technique was important to enable a semi-automatic involvement of the domain experts in order to effectively contribute in a discovery session.

4.2 Results

As discussed above, we provide an evaluation using exemplary real-world data from the industrial domain using the adapted SD-Map algorithm with pruning and the 'standard' SD-Map algorithm as a reference, using the Piatetsky-Shapiro quality function. The applied data set contains about 20 attributes and a relatively large number of cases. Figure 1 shows the number of hypotheses that were considered during the discovery process. As discussed above, the complexity grows exponentially with the number of attributes and attribute values. It is easy to see, that the optimistic estimate pruning approach shows a significant decrease in the number of hypotheses considered, and thus also in the runtime of the algorithm. The 'full' optimistic estimate pruning approach using dynamic reordering strategy (optimistic estimate pruning and dynamic sorting of the FP-Tree-header nodes) also shows a improvement by a factor of two compared to the approach using only optimistic estimate pruning.

As a further benchmark, Figure 2 shows results of applying the approach on the credit-g data set from the UCI [13] repository. We considered the target concept *credit amount*, and provided the nominal attributes of the data set subsequently. The results confirm the observation for the industrial data, and the decrease of considered hypotheses/running time is even more significant. Similar to the industrial data set, the 'full' pruning/sort strategy of SD-Map* shows slight 'variations' with respect to the number of considered hypotheses (cf., lines for 9 and 12 attributes of credit-g): This can be explained by the fact that the ordering strategy can yield better subgroup qualities earlier in the process.

These results clearly indicate the benefit and broad applicability of the approach: The pruning strategies enable fast (automatic) subgroup discovery for continuous target concepts, that can then be the starting point for further analysis.

#Attributes	w/o OE-Pruning	w/OE-Pruning	w/OE-Sort-Pruning
2	23	23	23
3	83	61	50
4	117	57	41
5	157	67	48
6	390	101	55
7	464	109	50
8	544	112	43
9	1546	131	48
10	2763	181	69
11	4333	170	84
12	4486	173	87
13	7567	219	110
14	9123	275	155
15	14057	325	186
16	20112	432	240
17	29708	548	335
18	43517	583	390
19	58139	612	381
20	65307	800	469

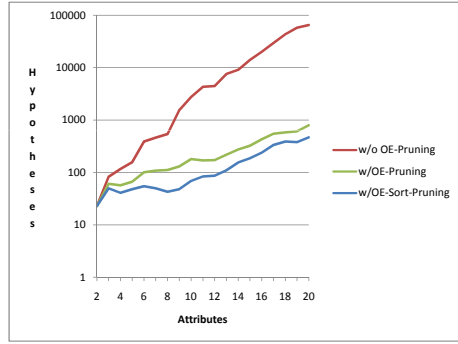


Fig. 1. Evaluation: Industrial Data. The x-axis of the graph shows the number of attributes provided to the discovery method, the y-axis shows the resulting number of considered hypotheses. The columns of the table indicate the number of hypotheses without pruning (*w/o OE-Pruning*), with optimistic-estimate pruning and no sorting (*w/OE-Pruning*), and with the full optimistic-estimate/sorting strategy (*w/OE-Sort-Pruning*).

#Attributes	w/o OE-Pruning	w/OE-Pruning	w/OE-Sort-Pruning
2	27	27	27
3	215	119	68
4	652	130	102
5	2019	181	158
6	4756	238	170
7	7342	240	177
8	16779	260	168
9	28770	347	212
10	50015	411	307
11	93626	448	343
12	168818	500	212
13	224087	558	229

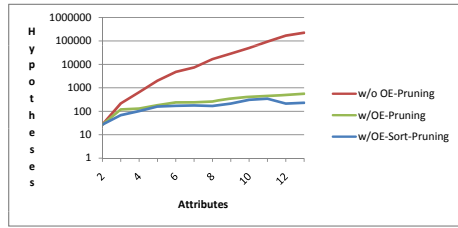


Fig. 2. Evaluation: Credit-G Data Set. The x-axis of the table shows the number of attributes provided to the discovery method, the y-axis shows the resulting number of considered hypotheses. The columns of the table indicate the number of hypotheses without pruning (*w/o OE-Pruning*), with optimistic-estimate pruning and no sorting (*w/OE-Pruning*), and with the full optimistic-estimate/sorting strategy (*w/OE-Sort-Pruning*).

5 Conclusions

In this paper, we have presented techniques for fast subgroup discovery with continuous target concepts of interest: These feature novel formalizations of tight optimistic estimate quality functions and the SD-Map* algorithm for enabling efficient subgroup discovery for continuous target concepts. The applied pruning techniques for safely removing areas of the search space are based on utilizing these tight optimistic estimate functions for continuous target concepts.

The evaluation of the approach was performed using real-world data from industrial applications and showed significant improvements concerning the efficiency of the subgroup discovery approach since large areas of the search space could be safely pruned in the experiments. This enables a seamless application of the algorithm even in rather interactive contexts.

For future work, we aim to assess a combination of tight (continuous) optimistic estimates with sampling techniques and methods for distributed subgroup discovery in order to optimize the efficiency of the subgroup discovery method even more. Another promising direction for future research is given by effective visualization techniques for continuous target concepts.

Acknowledgements

This work has been partially supported by the German Research Council (DFG) under grant Pu 129/8-2.

References

1. Gamberger, D., Lavrac, N.: Expert-Guided Subgroup Discovery: Methodology and Application. *Journal of Artificial Intelligence Research* 17, 501–527 (2002)
2. Lavrac, N., Kavsek, B., Flach, P., Todorovski, L.: Subgroup Discovery with CN2-SD. *Journal of Machine Learning Research* 5, 153–188 (2004)
3. Atzmueller, M., Puppe, F., Buscher, H.P.: Exploiting Background Knowledge for Knowledge-Intensive Subgroup Discovery. In: *Proc. 19th Intl. Joint Conference on Artificial Intelligence (IJCAI 2005)*, Edinburgh, Scotland, pp. 647–652 (2005)
4. Jorge, A.M., Pereira, F., Azevedo, P.J.: Visual interactive subgroup discovery with numerical properties of interest (ISI, ISIProc). In: Todorovski, L., Lavrač, N., Jantke, K.P. (eds.) *DS 2006. LNCS (LNAI)*, vol. 4265, pp. 301–305. Springer, Heidelberg (2006)
5. Wrobel, S.: An Algorithm for Multi-Relational Discovery of Subgroups. In: Komorowski, J., Żytkow, J.M. (eds.) *PKDD 1997. LNCS*, vol. 1263, pp. 78–87. Springer, Heidelberg (1997)
6. Aumann, Y., Lindell, Y.: A Statistical Theory for Quantitative Association Rules. *Journal of Intelligent Information Systems* 20(3), 255–283 (2003)
7. Atzmueller, M., Puppe, F.: SD-Map – A Fast Algorithm for Exhaustive Subgroup Discovery. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) *PKDD 2006. LNCS (LNAI)*, vol. 4213, pp. 6–17. Springer, Heidelberg (2006)
8. Klösgen, W.: Explora: A Multipattern and Multistrategy Discovery Assistant. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.) *Advances in Knowledge Discovery and Data Mining*, pp. 249–271. AAAI Press, Menlo Park (1996)
9. Grosskreutz, H., Rüping, S., Wrobel, S.: Tight optimistic estimates for fast subgroup discovery. In: Daelemans, W., Goethals, B., Morik, K. (eds.) *ECML PKDD 2008, Part I. LNCS (LNAI)*, vol. 5211, pp. 440–456. Springer, Heidelberg (2008)
10. Han, J., Pei, J., Yin, Y.: Mining Frequent Patterns Without Candidate Generation. In: Chen, W., Naughton, J., Bernstein, P.A. (eds.) *2000 ACM SIGMOD Intl. Conference on Management of Data*, pp. 1–12. ACM Press, New York (2000)
11. Klösgen, W.: Applications and Research Problems of Subgroup Mining. In: Raś, Z.W., Skowron, A. (eds.) *ISMIS 1999. LNCS*, vol. 1609, pp. 1–15. Springer, Heidelberg (1999)
12. Grosskreutz, H., Rüping, S., Shaabani, N., Wrobel, S.: Optimistic estimate pruning strategies for fast exhaustive subgroup discovery. Technical report, Fraunhofer Institute IAIS (2008), <http://publica.fraunhofer.de/eprints/urn:nbn:de:0011-n-723406.pdf>
13. Newman, D., Hettich, S., Blake, C., Merz, C.: *UCI Repository of Machine Learning Databases* (1998), <http://www.ics.uci.edu/~mllearn/mlrepository.html>

Discovering Emerging Graph Patterns from Chemicals

Guillaume Poezevara, Bertrand Cuissart, and Bruno Crémilleux

Laboratoire GREYC-CNRS UMR 6072
Université de Caen Basse-Normandie, France
Forename.Lastname@info.unicaen.fr
<http://www.greyc.unicaen.fr/>

Abstract. Emerging patterns are patterns of a great interest for characterizing classes. This task remains a challenge, especially with graph data. In this paper, we propose a method to mine the whole set of frequent emerging graph patterns, given a frequency threshold and an emergence threshold. Our results are achieved thanks to a change of the description of the initial problem so that we are able to design a process combining efficient algorithmic and data mining methods. Experiments on a real-world database composed of chemicals show the feasibility and the efficiency of our approach.

Keywords: Data mining, emerging patterns, subgraph isomorphism, chemical information.

1 Introduction

Discovering knowledge from large amounts of data and data mining methods are useful in a lot of domains such as chemoinformatics. One of the goals in chemoinformatics is to establish relationships between chemicals (or molecules) and a given activity (e.g., toxicity). Such a relationship may be characterized by *patterns* associating atoms and chemical bonds. A difficulty of the task is the number of potential patterns which is very large. By reducing the number of extracted patterns to those of a potential interest given by the user, the constraint-based pattern mining [12] provides efficient methods. A very useful constraint is the emerging constraint [5]: emerging patterns (EPs) are patterns whose frequency strongly varies between two classes (the frequency of a pattern P is the number of examples in the database supporting P). EPs enable us to characterize classes (e.g., toxic versus non-toxic chemicals) in a quantitative and qualitative way. EPs are at the origin of various works such as powerful classifiers [9]. From an applicative point of view, we can quote various works on the characterization of biochemical properties or medical data [10].

Even if a lot of progress has recently been made in the constraint-based pattern mining, mining EPs remains difficult because the anti-monotone property which is at the core of powerful pruning techniques in data mining [11] cannot be applied. As EPs are linked to the pattern frequency, naive approaches for

mining EPs extract frequent patterns in a class and infrequent patterns in the set of the other classes because the frequency and infrequency constraints satisfy (anti-)monotone properties and therefore there are techniques to mine such a combination of constraints. Unfortunately, such an approach only extract a subset of the whole set of EPs. That it is why some techniques use handlings of borders but it is very expensive [5]. In the context of patterns made of items (i.e., database objects are described by items), an efficient method based on a prefix-freeness operator leading to interval pruning was proposed [13,14]. More generally, most of the works on EPs are devoted to the itemset area and there are very few attempts in areas such as chemoinformatics where chemicals are graphs [24]. These last two works are based on a combination of monotone and anti-monotone constraints and do not extract the whole collection of EPs. Mining patterns in a graph dataset is a much more challenging task than mining patterns in itemsets.

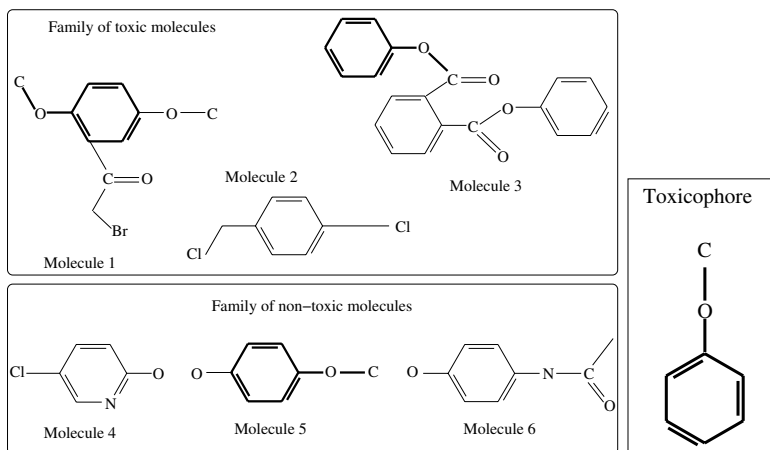
In this paper, we tackle this challenge of mining emerging graph patterns. Our main contribution is to propose a method mining all frequent emerging graph patterns. This result is achieved by a change of the description of the initial problem in order to be able to use efficient algorithmic and data mining methods (see Section 3). In particular, all frequent connected emerging subgraphs are produced; they correspond to the patterns of cardinality 1. These subgraphs are useful because they are the most understandable subgraphs from the chemical point of view. The patterns of cardinality greater than one capture the emerging power of associations of connected subgraphs. A great feature of our method is to be able to extract *all* frequent emerging graph patterns (given a frequency threshold and an emergence threshold) and not only particular EPs. Finally, we present a case study on a chemical database provided by the Environnement Protection Agency. This experiment shows the feasibility of our approach and suggests promising chemical investigations on the discovery of toxicophores.

This paper is organized as follows. Section 2 outlines preliminary definitions and related work. Our method for mining all frequent emerging graph patterns is described in Section 3. Experiments showing the efficiency of our approach and results on the chemical dataset are given in Section 4.

2 Context and Motivations

2.1 Notation and Definitions

Graph terminology. In this text, we consider simple labeled graphs. We recall here some important notions related to these graphs. A *graph* $G(V, E)$ consists of two sets V and E . An element of V is called a *vertex* of G . An element of E is called an *edge* of G , an edge corresponds to a pair of vertices. Two edges are *adjacent* if they share a common vertex. A *walk* is a sequence of edges such that two consecutive edges are adjacent. A graph G is *connected* if any two of its vertices are linked by a walk. Two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ are *isomorphic* if there exists a bijection $\psi : V_1 \rightarrow V_2$ such that for every $u_1, v_1 \in V_1$, $\{u_1, v_1\} \in E_1$ if and only if $\{\psi(u_1), \psi(v_1)\} \in E_2$; ψ is called an



(a) Molecules excerpted from the EPAFHM database [6] (b) Graph G_0

Fig. 1. Examples of molecules

isomorphism. Given two graphs $G'(V', E')$ and $G(V, E)$, G' is a *subgraph* of G if (a) V' is a subset of V and E' is a subset of E or if (b) G' is isomorphic to a subgraph of G . Given a family of graphs \mathcal{D} and a frequency threshold $f_{\mathcal{D}}$, a graph G is a *frequent subgraph* (of $(\mathcal{D}, f_{\mathcal{D}})$) if G is a subgraph of at least $f_{\mathcal{D}}$ graphs of \mathcal{D} ; a *frequent connected subgraph* is a frequent subgraph that is connected.

Graphs encountered in the text carry information by the meaning of *labellings* of the vertices and of the edges. The labellings do not affect the previous definitions, except that an isomorphism has to preserve the labels. A molecular graph is a labelled graph that depicts a chemical structure: a vertex represents an atom, an edge represents a chemical bond. Fig. 1(a) displays *molecular graphs*. The graph G_0 (see Fig. 1(b)) is (isomorphic to) a subgraph of molecules 1, 3 and 5 in Fig. 1(a) and therefore its frequency is 3. Assuming now that \mathcal{D} is partitioned into two subsets (or classes) \mathcal{D}_1 and \mathcal{D}_2 . For instance, in Fig. 1(a), its top part \mathcal{D}_1 gathers toxic molecules and its bottom part \mathcal{D}_2 non-toxic molecules. With a frequency threshold of 2 both for \mathcal{D}_1 and \mathcal{D}_2 , G_0 is a frequent graph in \mathcal{D}_1 but it is infrequent in \mathcal{D}_2 .

The problem of mining all the frequent connected subgraphs of $(\mathcal{D}, f_{\mathcal{D}})$ is called the *discovery of the Frequent Connected SubGraphs (FCSG)*. It relies on multiple subgraph isomorphism. Given a couple of graphs (G', G) , the problem of deciding if G' is isomorphic to a subgraph of G is named the *Subgraph Isomorphism Problem (SI)*. *SI* is NP-complete [7, p. 64]. The problem remains NP-complete if we restrict the input to connected graphs. Consequently, the discovery of the *FCSGs* is NP-Complete. The labellings do not change the class of complexity of *SI* and the discovery of the *FCSGs*.

In the following, we will need to compute the frequency of a set of graphs \mathcal{G} (i.e. a *graph pattern*). $\mathcal{F}(\mathcal{G}, \mathcal{D})$ denotes the graphs of \mathcal{D} that include every graph of \mathcal{G} as a subgraph ($\mathcal{F}(\mathcal{G}, \mathcal{D}) = \{G_{\mathcal{D}} \in \mathcal{D} : \forall G \in \mathcal{G}, G \text{ is a subgraph of } G_{\mathcal{D}}\}$).

For example, the graph pattern made of G_0 and the graph $c=0$ named G_1 has a frequency of 2 in \mathcal{D} (it is a subgraph of molecules 1 and 3). In this paper, a graph pattern is composed of *connected* graphs.

Emerging Graph Pattern (EGP). As introduced earlier, an emerging graph pattern \mathcal{G} is a set of graphs whose frequency increases significantly from one subset (or class) to another. The capture of contrast brought by \mathcal{G} from \mathcal{D}_2 to \mathcal{D}_1 is measured by its *growth rate* $GR_{\mathcal{D}_1}(\mathcal{G})$ defined as:

$$\begin{cases} 0, & \text{if } \mathcal{F}(\mathcal{G}, \mathcal{D}_1) = \emptyset \text{ and } \mathcal{F}(\mathcal{G}, \mathcal{D}_2) = \emptyset \\ \infty, & \text{if } \mathcal{F}(\mathcal{G}, \mathcal{D}_1) \neq \emptyset \text{ and } \mathcal{F}(\mathcal{G}, \mathcal{D}_2) = \emptyset \\ \frac{|\mathcal{D}_2| \times |\mathcal{F}(\mathcal{G}, \mathcal{D}_1)|}{|\mathcal{D}_1| \times |\mathcal{F}(\mathcal{G}, \mathcal{D}_2)|}, & \text{otherwise (}| \cdot | \text{ denotes the cardinality of a set)} \end{cases}$$

Therefore, the definition of an EGP is given by:

Definition 1 (Emerging Graph Pattern). *Let \mathcal{D} be a set of graphs partitioned into two subsets \mathcal{D}_1 and \mathcal{D}_2 . Given a growth threshold ρ , a set of connected graphs \mathcal{G} is an emerging graph pattern from \mathcal{D}_2 to \mathcal{D}_1 if $GR_{\mathcal{D}_1}(\mathcal{G}) \geq \rho$*

We can now provide the terms of the problem of mining the whole set of frequent EGPs:

Definition 2 (Frequent Emerging Graph Pattern Extraction (FEGPE))

Input: *let \mathcal{D} be a set of graphs partitioned into two subsets \mathcal{D}_1 and \mathcal{D}_2 , $f_{\mathcal{D}_1}$ a frequency threshold in \mathcal{D}_1 and ρ a growth threshold*

Output: *the set of the frequent emerging graph patterns with their growth rate from \mathcal{D}_2 to \mathcal{D}_1 according to $f_{\mathcal{D}_1}$ and ρ .*

The *length* of a graph pattern denotes its cardinality. Note that the set of frequent EGPs of length 1 from \mathcal{D}_2 to \mathcal{D}_1 corresponds to the set of frequent emerging connected graphs from \mathcal{D}_2 to \mathcal{D}_1 .

For the sake of simplicity, the definitions are given with only with two classes but all the results hold with more than two classes (it is enough to consider that $\mathcal{D}_2 = \mathcal{D} \setminus \mathcal{D}_1$, as usual in the EP area [5]). Following our example, with $f_{\mathcal{D}_1} = 2$ and $\rho = 2$, the FEGPE problem provides 273 intervals [14] (see Section 3) condensing the frequent emerging graph patterns including G_0 ($GR_1(G_0) = 2$) and the graph pattern \mathcal{G} made of G_0 and G_1 ($GR_1(\mathcal{G}) = \infty$).

2.2 Related Work: Extraction of Discriminative Subgraphs

Several methods have been designed for discovering subgraphs that are correlated to a given class.

Molfea [8] uses a levelwise algorithm [11] enabling the extraction of *linear subgraphs (chains)* which are frequent in a set of “positive” graphs and infrequent in a set of “negative” graphs. However, the restriction to linear subgraphs disables a direct extraction of the graphs containing a branching point or a cycle.

Moss [12] is a program dedicated to mine molecular substructure; it can be extended to find the *discriminative fragments*. Given two frequency thresholds f_M and f_m , a discriminative fragment corresponds to a connected subgraph whose frequency is above f_M in a set of “positive” graphs and is below f_m in a set of “negative” graphs. This definition differs from the usual notion of emergence which is based on the growth rate as introduced in the previous section. Indeed, mining all discriminative fragments according to the thresholds f_M and f_m do not ensure extracting all EPs having a growth rate higher than f_M/f_m or another given growth rate threshold. At the contrary, we will see that our approach follows the usual notion of emergence.

Another work has been dedicated to the discovery of the *contrast subgraphs* [15]. A contrast subgraph is a graph that appears in the set of the “positive” graphs but never in the set of the “negative” graphs. Although this notion is very interesting, it requires a lot of computation. To the best of our knowledge, the calculus is limited to one “positive” graph and the mining of a graph exceeding 20 vertices brings up a significant challenge. Furthermore, contrast subgraphs correspond to *jumping emerging patterns* (i.e., EPs with a growth rate equals ∞) and therefore are a specific case of the general framework of EPs.

3 Mining Frequent Emerging Graph Patterns

This section presents our method for mining frequent emerging graph patterns. We start by giving the key ideas and the three steps of our method.

Outline. Let \mathcal{D} be a set of graphs partitioned into two subsets \mathcal{D}_1 and \mathcal{D}_2 . Our main idea is to change the description of the initial problem in order to be able to use efficient algorithmic and data mining methods. Briefly speaking, for mining the whole set of the frequent EGPs from \mathcal{D}_2 to \mathcal{D}_1 , we start by only extracting the frequent connected subgraphs in \mathcal{D}_1 . Then, by using a subgraph isomorphism method, both molecules of \mathcal{D}_1 and \mathcal{D}_2 are described with the frequent connected subgraphs as new features. This change of description of the problem brings a twofold advantage. First, as EGPs can only stem from these new features, it is enough in \mathcal{D}_2 to focus on candidate patterns made of these features. It strongly reduces the number of candidate patterns and this is precious especially in \mathcal{D}_2 because we have in this dataset to deal with graphs with very low frequency. Second, it enables us to set the problem in an itemset context from which we can reuse efficient results on the emerging constraint. Finally, we solve the *FEGPE* problem described in Section 2.1.

Main steps of our method. Fig. 2 depicts the three main steps of our method:

- 1) extracting the frequent connected subgraphs in \mathcal{D}_1 according to the frequency threshold $f_{\mathcal{D}_1}$. This is the *FCSG* problem.
- 2) for each graph $G_{\mathcal{D}}$ of \mathcal{D} and for each connected graph G resulting from 1), we successively test if G is a subgraph of $G_{\mathcal{D}}$. For that purpose, we have to solve multiple *SI* problems. For that task, we use our own implementation of J.R. Ullmann’s algorithm [16]. Then the dataset can be recoded such that

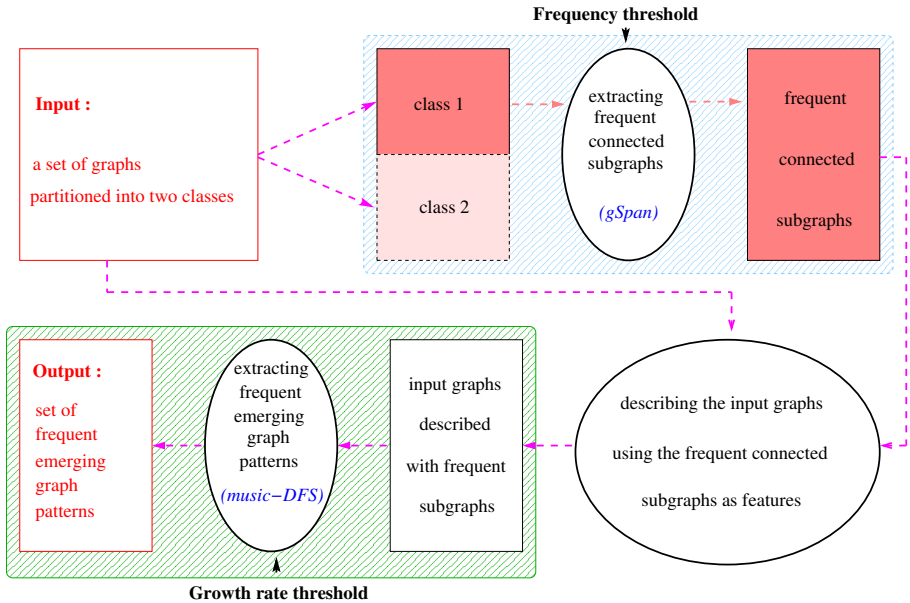


Fig. 2. The three steps of the process: extracting FCSGs (*up*), describing the input graphs using the FCSGs as features (*down-right*) and extracting frequent EGPs (*down-left*)

each row is a graph G of \mathcal{D} and each column indicates if a frequent connected graph issued from 1) is present or not in G .

- 3) the problem is then described by items (presence or absence of each frequent connected graph) and we are able to use an efficient method (i.e., MUSIC-DFS) based on itemsets to discover the frequent emerging graph patterns.

There are several methods to solve the *FCSG* problem. The algorithms are classified into two families: the *Apriori-Based* algorithms and the *Pattern-Growth-Based* algorithms. The two families have been compared for mining sets of chemical graphs [3]: the Apriori-Based algorithms spend less time while the Pattern-Growth-Based algorithms consume less memory. A comparison of four Pattern-Growth-Based algorithms has been conducted in [18]. For mining a set of chemical graphs, *gSpan* [19] consumes less memory and runs faster than the other ones. For these reasons, we have chosen *gSpan* for extracting frequent connected subgraphs. Moreover, *gSpan* is available on <http://illimine.cs.uiuc.edu/> under GNU GPL License Version 2¹.

Frequent emerging graph patterns are mined by using MUSIC-DFS². This tool offers a set of syntactic and aggregate primitives to specify a broad spectrum of constraints in a flexible way, for data described by items [14]. Then MUSIC-DFS mines soundly and completely all the patterns satisfying a given set of input

¹ <http://www.gnu.org/licenses/gpl/html>

² <http://www.info.univ-tours.fr/~soulet/music-dfs/music-dfs.html>

Table 1. Excerpt from the EPAFHM database: 395 molecules partitioned into two subsets according to the measure of $LC50$

Class	Subset	Toxicity	$LC50$ measure	Number of molecules
1	toxic	\mathcal{D}_1	$LC50 \leq 10mg/l$	223
2	non-toxic	\mathcal{D}_2	$100mg/l \leq LC50$	172

constraints. The efficiency of MUSIC-DFS lies in its depth-first search strategy and a safe pruning of the pattern space by pushing the constraints. The constraints are applied as early as possible. The pruning conditions are based on intervals. Here, an *interval* denominates a set of patterns that include a same prefix-free pattern P and that are included in the prefix-closure of P (see [14] for more details). Whenever it is computed that all the patterns included in an interval simultaneously satisfy (or not) the constraint, the interval is positively (negatively) pruned without enumerating all its patterns [14]. The output of MUSIC-DFS enumerates the intervals satisfying the constraint. Such an interval condensed representation improves the output legibility and each pattern appears in only one interval. In our context, this tool enables us to use the emerging and frequency constraints.

Our approach ensures to produce the whole set of frequent EGPs because the *FCSG* step extracts all the connected subgraphs and MUSIC-DFS is complete and correct for the pattern mining step.

4 Experiments on Chemical Data

Experiments are presented according to the three steps of our method. They show the feasibility of our approach and provide quantitative results.

The dataset gathers molecules stored in *EPA Fathead Minnow Acute Toxicity Database* [6] (EPAFHM). It has been generated by the Environment Protection Agency (EPA) of the United-States, it has been used to elaborate expert systems predicting the toxicity of chemicals [17]. From EPAFHM, we have selected the molecules classified as toxic and non-toxic, toxicity being established according to the measure of $LC50$. The resulting set \mathcal{D} contains 395 molecules (Table 1) and it is partitioned into two subsets: \mathcal{D}_1 contains toxic molecules (223 molecules) and \mathcal{D}_2 contains non-toxic molecules (172 molecules). Experiments were conducted on a computer running *Linux* operating system with a dual processor at 2.83 GHz and a RAM of 1.9 Gio.

Extraction of the FCSGs. As already said, we use gSpan to extract the set of FCSGs. The input set of graphs is the set \mathcal{D}_1 . The frequency threshold $f_{\mathcal{D}_1}$ varies from 1 % to 10 % with a step of 1 %. For each calculation, we measure the number of frequent connected subgraphs extracted and the computing time. Results are displayed in Fig. 3.

First, as expected, the number of extracted subgraphs decreases exponentially as the frequency increases. It takes 8 seconds to extract 49438 subgraphs ($f_{\mathcal{D}_1} =$

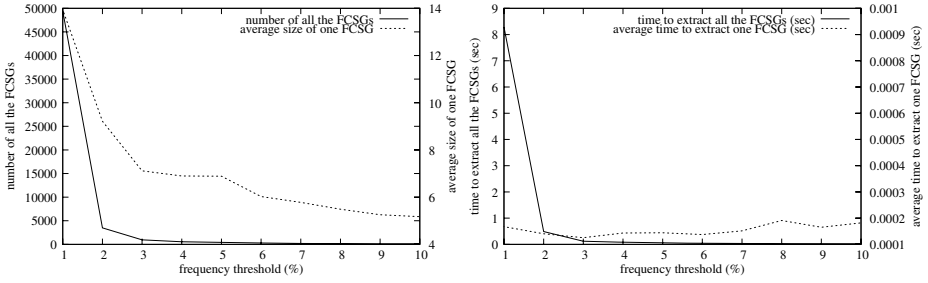


Fig. 3. Extraction of FCSGs according to the frequency threshold

Table 2. Measures on SI according to frequency threshold

frequency threshold	1	2	3	4	5	6	7	8	9	10
number of descriptors	49438	3492	956	558	416	291	198	157	121	110
avg time to describe a graph (sec)	139	7.49	2.04	1.18	0.870	0.612	0.420	0.329	0.255	0.250
avg time per isomorphism (sec)	$2.27 \cdot 10^{-3}$	$2.11 \cdot 10^{-3}$	$2.09 \cdot 10^{-3}$	$2.12 \cdot 10^{-3}$	$2.10 \cdot 10^{-3}$	$2.09 \cdot 10^{-3}$	$2.12 \cdot 10^{-3}$	$2.13 \cdot 10^{-3}$	$2.14 \cdot 10^{-3}$	$2.82 \cdot 10^{-3}$

1%) and 1 second to extract 110 subgraphs ($f_{\mathcal{D}_1} = 10\%$). Second, the computing time is strongly related to the number of extracted subgraphs: the average time to extract one FCSG varies from $1 \cdot 10^{-4}$ second to $2 \cdot 10^{-4}$ second. Third, the average size of an extracted subgraph decreases as frequency increases: from a average size of 14 vertices ($f_{\mathcal{D}_1} = 1\%$) to an average size of 5 vertices ($f_{\mathcal{D}_1} = 10\%$).

Description of the Graphs Using Subgraphs as Features. As in the previous experiment, $f_{\mathcal{D}_1}$ varies from 1% to 10% with a step of 1%. For each frequency threshold, the set of descriptors is the set of FCSGs extracted from \mathcal{D}_1 . Each graph of \mathcal{D} is then recoded according to these descriptors, one SI is required for recoding each descriptor. We count the number of successful SIs and we measure the computing time. Results are displayed on Table 2.

Obviously, the number of descriptors decreases as the frequency threshold increases (see the previous experiment). The average computing time to describe a graph decreases as the the number of descriptors decreases: it takes 140 seconds with 49438 descriptors ($f_{\mathcal{D}_1} = 1\%$) and 0.250 second with 110 descriptors ($f_{\mathcal{D}_1} = 10\%$). As the size of each descriptor remains small, the average time per isomorphism is stable (around $2 \cdot 10^{-3}$ second). Consequently, the average computing time to describe a graph is strongly correlated to the number of descriptors.

Extraction of Frequent EGPs. The input dataset is the set \mathcal{D} recoded with the FCSGs extracted from \mathcal{D}_1 under a frequency threshold of 5% (416 descriptors). The growth rate threshold ρ varies from 2 to 20 with a step of 1. For each

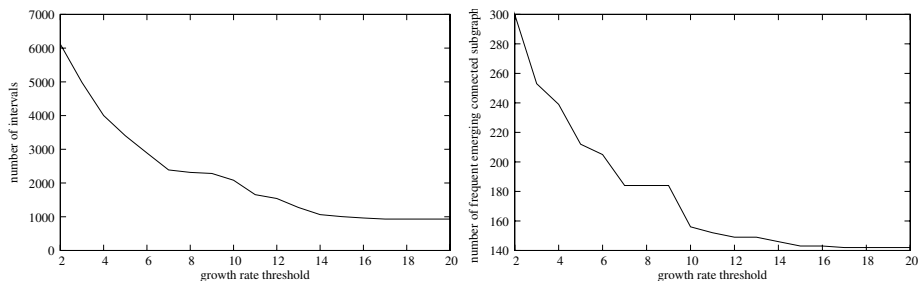


Fig. 4. Extraction of the frequent EGPs according to the growth rate threshold

value of ρ , we indicate the number of intervals mined by MUSIC-DFS. We also indicate the number of frequent emerging connected subgraphs as these graphs are of particular interest in chemoinformatics. Recall that the set of frequent emerging connected subgraphs corresponds to the set of frequent EGPs of length 1. Computing time is not provided because it is always very fast (MUSIC-DFS always extracts all emerging graph patterns in less than one second). Results are displayed on Fig 4.

The number of intervals decreases as ρ increases: it varies from 6000 ($\rho = 2$) to 1000 ($\rho = 20$). The number of emerging graphs decreases as growth rate threshold ρ increases: it varies from 300 ($\rho = 2$) to 140 ($\rho = 20$). Interestingly, such experiments depict the speed of decreasing of the number of patterns according to the growth rate threshold. We are also able to extract the jumping emerging patterns (see Section 2.2, page 49). An EGP is a jumping emerging pattern if it is always extracted whatever the value of the growth rate is. For this experiment, there are 635 jumping emerging patterns, 69 of them are of length 1. These latter are of particular interest for the toxicologist.

An overall result of these experiments is to show that mining emerging graph patterns from real-world chemical dataset is feasible. About use of EGPs in chemoinformatics, these patterns are currently used by the toxicologists in the search for discovering toxicophores because these latter are strongly present in toxic molecules and may be responsible of their toxicity. However, toxicity does not rely on the sole presence of a toxicophore (a pattern of length one). Indeed many toxicophores could be inhibited by a neighboring fragment. Although our tool underlines these interactions, we still don't know how to explain the patterns of length greater than one.

We are now processing a bigger dataset excerpted from the *Registry of Toxic Effects of Chemical Substances* (<http://www.cchst.ca/products/rtecs/>). This set contains more than 10 000 molecular graphs, along with their toxicity measures. The EGPs resulting from this experiment will provide valuable information for toxicologists.

5 Conclusion and Future Work

In this paper, we have investigated the notion of emerging graphs and we have proposed a method to mine emerging graph patterns. A strength of our approach

is to extract *all* frequent emerging graph patterns (given thresholds of frequency and emerging) and not only particular emerging patterns. In the particular case of patterns of length 1, all frequent connected emerging subgraphs are produced. Our results are achieved thanks to a change of the description of the initial problem so that we are able to design a process combining efficient algorithmic and data mining methods. Experiments on a real-world database composed of chemicals have shown the feasibility and the efficiency of our approach. Further work is to better investigate the use of such patterns in cheminformatics, especially for discovering toxicophores. A lot of data can be modeled by graphs and, obviously, emerging graph patterns may be used for instance in text mining or gene regulation networks.

Acknowledgments. The authors would like to thank Arnaud Soulet for very fruitful discussions and the MUSIC-DFS prototype and the CERMN lab for its invaluable help about the data and the chemical knowledge. This work is partly supported by the ANR (French Research National Agency) funded project Bingo2 ANR-07-MDCO-014 and the Region Basse-Normandie (INNOTOX2 project).

References

1. Borgelt, C., Berthold, M.R.: Mining molecular fragments: Finding relevant substructures of molecules. In: Proceedings of the IEEE International Conference on Data Mining (ICDM 2002), pp. 51–58 (2002)
2. Borgelt, C., Meinel, T., Berthold, M.: Moss: a program for molecular substructure mining. In: Workshop Open Source Data Mining Software, pp. 6–15. ACM Press, New York (2005)
3. Cook, D.J., Holder, L.B.: Mining Graph Data. John Wiley & Sons, Chichester (2006)
4. De Raedt, L., Kramer, S.: The levelwise version space algorithm and its application to molecular fragment finding. In: IJCAI 2001, pp. 853–862 (2001)
5. Dong, G., Li, J.: Efficient mining of emerging patterns: discovering trends and differences. In: Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD 1999), pp. 43–52. ACM Press, New York (1999)
6. EPAFHM. Mid continent ecology division (environnement protection agency), fathead minnow, http://www.epa.gov/med/Prods_Pubs/fathead_minnow.htm
7. Garey, M.R., Johnson, D.S.: Computers and Intractability. Freeman and Company, New York (1979)
8. Kramer, S., Raedt, L.D., Helma, C.: Molecular feature mining in HIV data. In: KDD, pp. 136–143 (2001)
9. Li, J., Dong, G., Ramamohanarao, K.: Making use of the most expressive jumping emerging patterns for classification. Knowledge and Information Systems 3(2), 131–145 (2001)
10. Li, J., Wong, L.: Emerging patterns and gene expression data. Genome Informatics 12, 3–13 (2001)
11. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery 1(3), 241–258 (1997)

12. Ng, R.T., Lakshmanan, V.S., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained associations rules. In: Proceedings of ACM SIGMOD 1998, pp. 13–24. ACM Press, New York (1998)
13. Soulet, A., Crémilleux, B.: Mining constraint-based patterns using automatic relaxation. *Intelligent Data Analysis* 13(1), 1–25 (2009)
14. Soulet, A., Kléma, J., Crémilleux, B.: Efficient Mining under Rich Constraints Derived from Various Datasets. In: Džeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, pp. 223–239. Springer, Heidelberg (2007)
15. Ting, R.M.H., Bailey, J.: Mining minimal contrast subgraph patterns. In: Ghosh, J., Lambert, D., Skillicorn, D.B., Srivastava, J. (eds.) SDM, pp. 638–642. SIAM, Philadelphia (2006)
16. Ullman, J.: An algorithm for subgraph isomorphism. *Journal of the ACM* 23, 31–42 (1976)
17. Veith, G., Greenwood, B., Hunter, R., Niemi, G., Regal, R.: On the intrinsic dimensionality of chemical structure space. *Chemosphere* 17(8), 1617–1644 (1988)
18. Wörlein, M., Meinel, T., Fischer, I., Philippsen, M.: A quantitative comparison of the subgraph miners mofa, gspan, FFSM, and gaston. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 392–403. Springer, Heidelberg (2005)
19. Yan, X., Han, J.: gspan: Graph-based substructure pattern mining. In: ICDM. LNCS, vol. 2394, pp. 721–724. IEEE Computer Society Press, Los Alamitos (2002)

Visualization of Trends Using RadViz

Lenka Nováková and Olga Štěpánková

Department of Cybernetics, Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2, 166 27 Prague 6, Czech Republic
{novakova,step}@labe.felk.cvut.cz

Abstract. Data mining is sometimes treating data consisting of items representing measurements of a single property taken in different time points. In this case data can be understood as a time series of one feature. It is no exception when the clue for evaluation of such data is related to their development trends as observed in several successive time points.

From the qualitative point of view one can distinguish 3 basic types of behavior between two neighboring time points: the value of the feature is stable (remains the same), it grows or it falls. This paper is concerned with identification of typical qualitative development patterns as they appear in the windows of given length in the considered time-stamped data and their utilization for specification of interesting subgroups.

Keywords: Time Series, Data Visualization, RadViz.

1 Introduction

The paper explains a novel approach to the search of typical qualitative patterns of development trends as they appear in windows of time stamped sequences of measurements of a single feature (e.g. weight of a person). Our solution is based on RadViz [2] 2D visualization of relevant data which is implemented in our SW tool. The applicability of the suggested solution is tested and demonstrated on the Stulong data set.

Our approach takes advantage of an interesting feature of RadViz visualization method [3] [4], namely its ability to depict some relational dependencies as studied in [7]. The Fig. 1 offers a clear example of a relational dependency among data that can be easily identified in a RadViz picture. This is a natural consequence of the RadViz mapping definition as briefly explained in the Section 2. Specific RadViz properties and modifications useful for analysis of time stamped data as well as design principles of our SW tool are treated in the Section 3. Some results obtained for Stulong data are reviewed in the Section 4. We conclude by pointing to several open questions in the last section where our plans for further development are outlined.

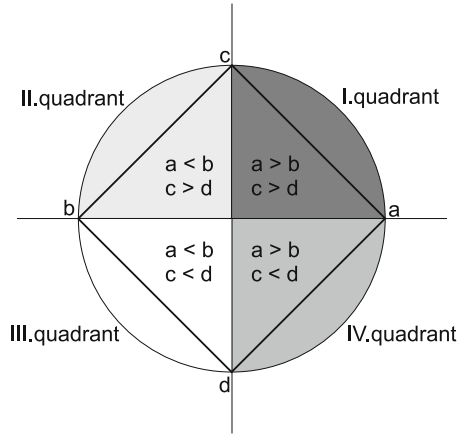


Fig. 1. The RadViz can show relations

2 Description of RadViz Mapping

RadViz is a mapping from n -dimensional space into a plane [2] [3] [4]. The point $y = [y_1, \dots, y_n]$ in an n -dimensional space is mapped to the point $u = [u_1, u_2]$ with following coordinates:

$$u_1 = \frac{\sum_{j=1}^n y_j \cos(\alpha_j)}{\sum_{j=1}^n y_j}$$

$$u_2 = \frac{\sum_{j=1}^n y_j \sin(\alpha_j)}{\sum_{j=1}^n y_j}$$

Let us consider the case depicted on the Fig. 1. Here, the angles corresponding to the anchors a, b, c and d are $\alpha = 0$, $\beta = \pi$, $\gamma = \frac{\pi}{2}$ and $\delta = \frac{3\pi}{2}$. After substituting the respective values into these equations we get the simplified equation for the point $u = [u_1, u_2]$:

$$u_1 = \frac{a - b}{a + b + c + d}$$

$$u_2 = \frac{c - d}{a + b + c + d}$$

All records from the data set, where the value of the attribute a is greater than that of b , are depicted in the first and the fourth quadrant of RadViz graph, see the Fig. 1. The data are divided by the vertical axis, which runs between anchors a and b and crosses the origin. On its right side there lie all points, where a is greater than b . On its left side, there lie the points where a is less than b . On the axis itself there are all the points for which a equals b . The

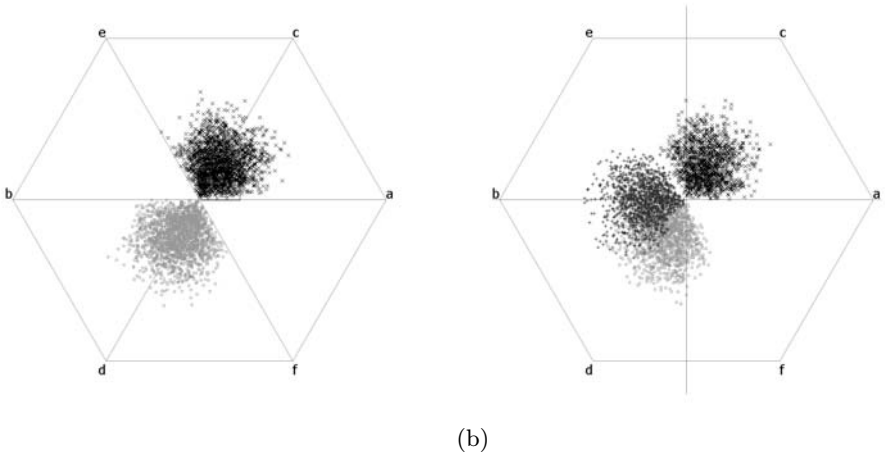


Fig. 2. Visualization of a relational dependency: a) $a < b \ \& \ c < d \ \& \ e < f$ in square points, $a > b \ \& \ c > d \ \& \ e > f$ in cross points color b) data from previous the image joined with those meeting the constraint ($a < b \ \& \ c < d \ \& \ e > f$)

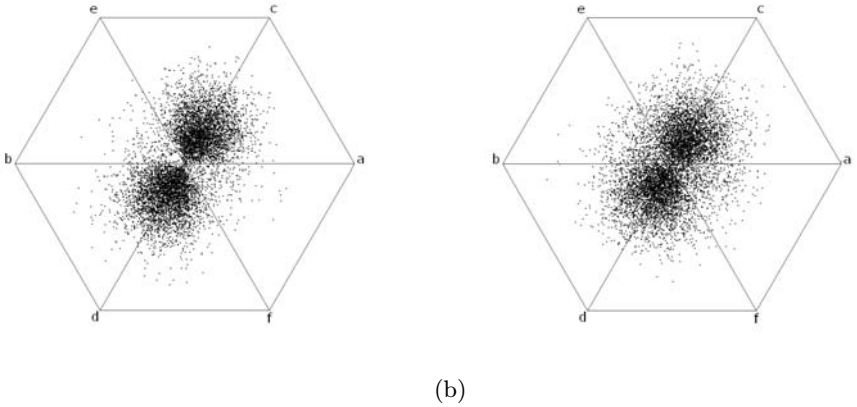


Fig. 3. Relation between three pairs: a) 10% noisy b) 30% noisy

same situation appears in the case of the attributes c and d . The Fig. 1 shows that given two pairs of attributes we can clearly distinguish those sections of the RadViz picture corresponding to various possible conjunctive combinations of their relations ($a < b \ \& \ c < d$, etc.). Each of the 9 possible non-overlapping sets of data described by upper mentioned constraints is mapped into a separate subset of the RadViz image (these sets represent its disjunctive partition).

Unfortunately, there is no straightforward generalization of the situation observed in the Fig. 1 as soon as there are considered sets of attribute pairs of

cardinality higher than 2. Even in this case we can identify sections in the RadViz image corresponding to specific patterns of qualitative behavior described by conjunction of relations between the values of pairs of the considered attributes, see for example Fig. 2.a.

We have to be aware of fact that the section corresponding to the data points meeting the condition $(a < b \ \& \ c < d \ \& \ e > f)$ has a large overlap with the section $(a < b \ \& \ c < d \ \& \ e < f)$. The images of the disjoint data subsets of our interest are no more disjoint and this causes problems in the interpretation of a RadViz image, see the Fig. 2.b - denote that the 3 depicted sets are of the same size. The next two figures Fig. 3.a and Fig. 3.b show the situation from Fig. 2.a with 10% and 30% of noise, respectively.

3 Visualization of Time Series in RadViz

Time series is a sequence of samples measured in fixed time points. Often, the data result from repeated measurements of a certain feature (e.g. blood pressure) for individual studied objects (e.g. patients). For simplicity let us assume that all the measured values appear in the interval $\langle 0, 1 \rangle$. If this is not the case the values have to be normalized using the global minimum and maximum for all the used attributes. Whenever the values of two attributes are the same they remain the same after normalization, of course. Such data are frequently visualized by a graph called “parallel coordinates”, where values corresponding to a single object are connected by a single broken line, see the Fig. 4.

Description of data through parallel coordinates provides valuable information if the size of the data set is small and the lines corresponding to individual objects can be easily distinguished e.g. by different colors. But as soon as the number of studied objects grows, the picture created in parallel coordinates provides no more evidence about the studied data. In general, one cannot see from the picture whether there is any dominant pattern appearing in the data or not, see the Fig. 9. Let us search for the qualitative development patterns appearing in the studied data. The value between two neighboring attributes can “grow”, “be stable” or “fall”. If we have measurements in $n+1$ consecutive time points, we can observe 3^n different shapes of the corresponding time series. In the case where we distinguish only behavior types “grow” or “fall” there exist 2^n shapes. Clearly, systematic search through such a space does not seem reasonable. Fortunately, it is not necessary to solve the task of finding large enough interesting subsets of time series data that are characterized by a single qualitative development trend (e.g. “grow”, “fall”, “grow”):

Suppose, we are studying data series of the length m described by attributes $[at_1, \dots, at_m]$ with the aim to find interesting subsets of data that are characterized by a qualitative development trend of the length lower than m and starting from the value of at_1 . If the lower bound for the cardinality of the target set is given we can apply Apriori-like approach and start by the search for the interesting subsets characterized by the trends among the initial three attributes $[at_1, at_2, at_3]$. Any interesting subset has to be a refinement (or more precisely

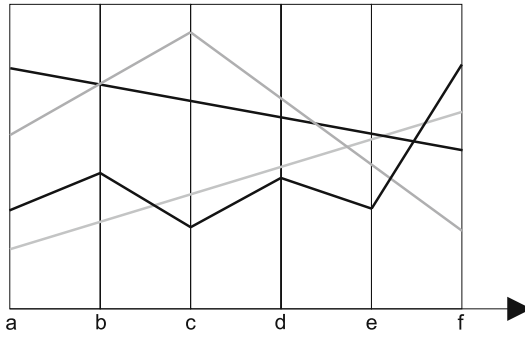


Fig. 4. Illustration example of time series

an extension towards further attributes) of an interesting subset which has been identified in the former step. In other words, we can focus our attention to the sets identified in the first step only and proceed by searching for their refinement using $at_4, , at_3$ etc. recursively.

We already know that RadViz can easily depict the relation “less than”. Let us utilize this feature to visualize time series. Let us denote the values corresponding to measurements in the time points 0, 1, 2 and 3 by attributes a, b, c and d . The trend between the attributes a and b from Fig. 4 is $b - a$ and its qualitative value can be naturally identified from its position in the RadViz picture without any need for additional computation provided the anchors a and b are situated opposite each other as in the Fig. II

Let us consider two pairs of successive attributes, the first pair is a, b and the second pair is b, c and a RadViz image with 4 dimensional anchors, where two anchors are bound to the attribute b . The corresponding anchors are denoted as b_1 and b_2) in the Fig. 5. We can replace this doubled dimensional anchor by a single dimensional anchor as shown in the the Fig. 5. The RadViz projections resulting from both the anchor settings are the same - compare for example the RadViz images at the Fig. 7.a and the Fig. 7.b of the Stulong data set treated in the following section with 4 and 3 dimensional anchors.

In the rest of the paper, we will work with the specific RadViz projections using three anchors only, that are identical to RadViz projections with two pairs of anchors. The radius of the dimensional anchor a is 1 and the angle $\alpha = 0$. The dimensional anchor for attribute b has radius $\sqrt{2}$ and angle is $\beta = \frac{3}{4}\pi$. This anchor lies further from center than the anchor for attributes a and c . The dimensional anchor for attribute c has radius 1 and its angle is $\gamma = \frac{3}{2}\pi$.

The equation for counting point $u = [u_1, u_2]$ of RadViz image are:

$$u_1 = \frac{a - b}{a + b + c}, \quad u_2 = \frac{b - c}{a + b + c}. \tag{1}$$

The design of anchors and the key how to interpret the trends as they appear in the RadViz image is shown in the Fig. 5.

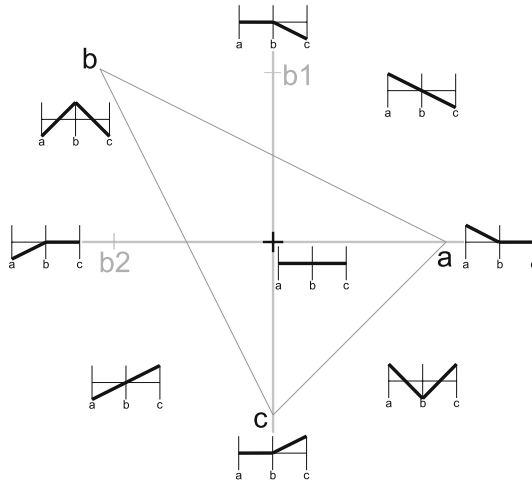


Fig. 5. How to detect trends in RadViz

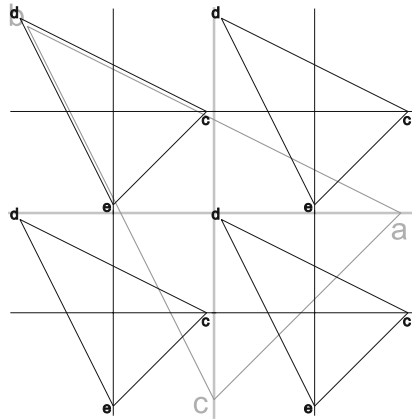


Fig. 6. Recursive expansion of RadViz image

Obviously, information about the development trends in the intervals from a to b and from b to c can be depicted in the orthogonal coordinates representing the values $(b - a)$ and $(c - b)$, too. We have decided to use RadViz visualization because it works directly with the data as they are (no need to create a new matrix with differences of neighboring attributes) and our SW tool implementing modification of RadViz offers number of additional features that proved helpful in the phase of data understanding and we can take its advantage e.g. when searching for frequent development trends.

This usage of RadViz can complement other visualization methods for example Parallel Coordinates as it can help the user to select group of data records

with some trends. This is impossible to be done directly in Parallel Coordinates. Eventuality, the RadViz graph can be expanded recursively as is suggested in the Fig. 6.

4 Identification of Trends in Stulong Data Set

The Stulong data set¹ comes from a truly longitudinal study aimed at primary prevention of atherosclerotic cardiovascular diseases (CVD). The study covers observation of more than 1400 middle aged men during 20 years. The intention of the project was to identify significant atherosclerosis risk factors, to follow development of these risk factors and their impact on the examined men health, especially with respect to atherosclerotic cardiovascular diseases.

Let us try to identify difference in time development of monitored physiological attributes between the two groups of patients classified with respect to occurrence of cardiovascular disease:

- patients who fell in by cardiovascular disease during the study (CVD = 1)
- patients who remained healthy during the study (CVD = 0)

The data from Stulong data set was preprocessed and there was proposed a new definition of derived attributes by combination of windowing and discretization [6]. These derived attributes were used for interesting sub-group discovery using association rules [9].

The RadViz method can help in the data understanding phase by depicting the first view to the trends appearing in the data. The Fig. 7.b shows the trends of systolic blood pressure in two neighboring intervals (between Syst0, Syst1 and between Syst1, Syst2). The CVD classification of the studied data offers an additional means for specification of an interesting subgroup. It is such a subgroup where the frequency of the classes is significantly different than that in the original set of data.

Our implementation of RadViz enables to explore the graph in detail. The user can interactively select by a rectangle the group of points and the system returns the table of all corresponding data records. Moreover, it summarizes the relevant information using a pie graph: while the inner pie gives the distribution of data in the full (original) data set, the outer pie describes the selected group, see the Fig. 8.a and Fig. 8.b.

The Fig. 8.a and Fig. 8.b show the difference between the whole set of patients and the selected group of the patients. The group of patients, who fall in

¹ The STULONG study was performed at the 2nd Dept. of Internal Medicine of the 1st Medical Faculty of the Charles University and General Faculty Hospital in Prague 2 (initially General Medical Faculty of the Charles University and Faculty Hospital I) managed by clinicians prof. MUDr. F.Boudik, DrSc., MUDr. M.Tomeckova, CSc. and doc. MUDr. J.Bultas, CSc. Most of the data was transferred into the electronic form by EuroMISE (European Centre for Medical Informatics, Statistics and Epidemiology) with support of European project [10].

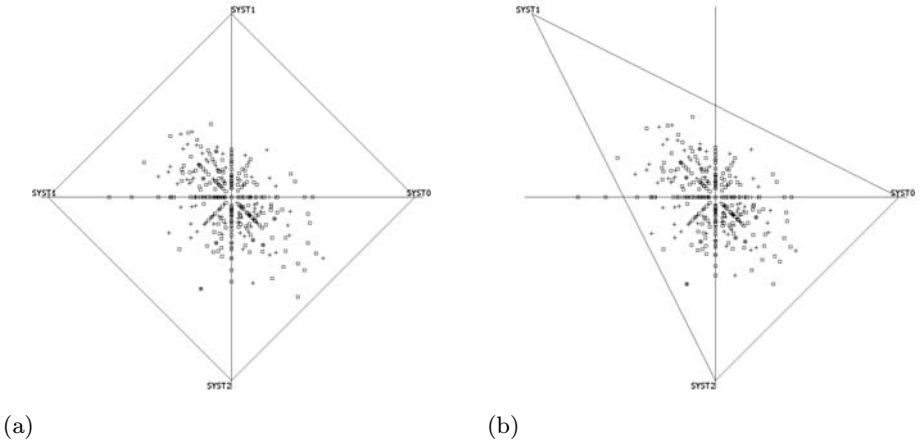


Fig. 7. Visualization of trends between two interval of systolic blood pressure (between Syst0, Syst1 and between Syst1, Syst2)

cardiovascular disease, are depicted in light grey color. The group of patients, who stay healthy, is depicted in dark grey color.

The difference between the whole group and patients with falling systolic blood pressure in both intervals are shown in the Fig. 8.a. This group fell in by cardiovascular disease less often than the whole group. On the other hand the Fig. 8.b shows the difference between the whole set and patients with growing systolic blood pressure in both intervals. This group has more than 1.5 times frequency that they fall in by cardiovascular disease. It is obvious that there is the difference between these two types of trends.

Using the suggested approach we were able to find quickly more interesting subgroups than those reported in [8]. One of such interesting subgroups is that

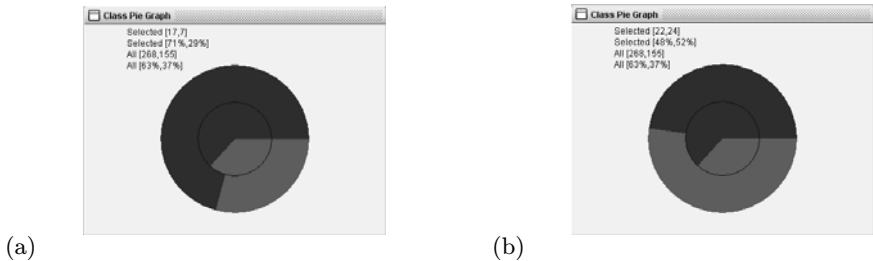


Fig. 8. Difference between CVD occurrences in the original data set and a subset specified through qualitative development trends: a) ($Syst0 > Syst1$ & $Syst1 > Syst2$), i.e. the trend of systolic blood pressure is “fall”-“fall” and b) ($Syst0 < Syst1$ & $Syst1 < Syst2$), i.e. the trend of systolic blood pressure is “grow”-“grow”

of patients, who have in the beginning stable diastolic blood pressure $Diast0 = Diast1$ and $Diast1 = Diast2$ followed by a jump up, i.e. $Diast2 < Diast3$. The group has 21 members (it is 5% of whole population) and there is a significant difference in the frequency of the patients with $CVD=0$ in this subgroup when compared to the original set, namely 27%.

5 Conclusion

The RadViz method provided significant help in search for qualitative patterns in the trends in the Stulong data set. Our experience proves that the RadViz visualization complements well the Parallel Coordinates [5] approach, see the Fig. 9: while Parallel Coordinates provide the global view to data, RadViz image can mediate more accurate local view.

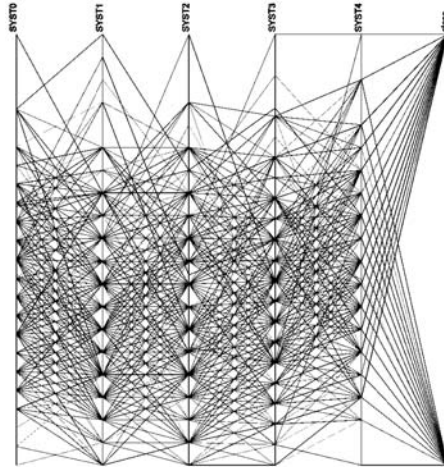


Fig. 9. Visualization of Stulong data set by Parallel Coordinates

Currently, we are finishing an implementation of a SW tool for data analysis that is extensively utilizing RadViz visualization. One of services it will offer is the systematic top-down automated search for frequently occurring qualitative trends. The tool applies the upper mentioned solution inspired by Apriori algorithm [1] to stop the recursive expansion, whenever there is not enough data in the expanded groups. Of course, it will ensure frequently used statistical evaluation to verify significance of the identified subsets. For practical applications it might be useful to introduce a possibility to work with fuzzy relations (e.g. “fuzzy equal”). This will be considered in further releases.

Acknowledgements. The presented research and development has been supported by the grant 1ET101210513 (Relational Machine Learning for Biomedical Data Analysis).

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast Discovery of Association Rules. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.) *Advances in Knowledge Discovery and Data Mining*, ch. 12, pp. 307–328. AAAI/MIT Press, Cambridge (1996)
2. Fayyad, U.M., Grinstein, G.G., Wierse, A.: *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann Publishers, San Francisco (2002)
3. Hoffman, P., Grinstein, G., Marx, K., Grosse, I., Stanley, E.: DNA Visual And Analytic Data Mining. In: *Proceedings of the IEEE Visualization 1997 Conference*, pp. 437–441 (1997)
4. Hoffman, P., Grinstein, G., Pinkney, D.: Dimensional anchors: a graphic primitive for multidimensional multivariate information visualizations. In: *Proceedings of the 1999 Workshop on New Paradigms in information Visualization and Manipulation*, pp. 9–16 (1999)
5. Inselberg, A.: The Plane with Parallel Coordinates. *Special Issue on Computational Geometry, The Visual Computer* 1, 69–91 (1985)
6. Kléma, J., Nováková, L., Karel, F., Štěpánková, O., Železný, F.: Sequential Data Mining: A Comparative Case Study in Development of Atherosclerosis Risk Factors. *IEEE Transactions on Systems, Man, and Cybernetics: Part C* 38(1), 3–15 (2008)
7. Nováková, L., Štěpánková, O.: Visualization of Some Relational Patterns for DM. In: *Cybernetics and Systems 2006, Vienna*, vol. 2, pp. 785–790 (2006)
8. Nováková, L., Karel, F., Aubrecht, P., Tomečková, M., Rauch, J., et al.: Trends in time windows as risk factors of cardiovascular disease. In: *Znalosti 2008*, pp. 148–159. Slovak University of Technology, Bratislava (2008) (in Czech)
9. Rauch, J., Šimůnek, M.: GUHA Method and Granular Computing. In: Hu, X., et al. (eds.) *Proceedings of IEEE conference Granular Computing*, pp. 630–635 (2005)
10. Project STULONG, WWW page, <http://euromise.vse.cz/stulong>

Action Rules Discovery Based on Tree Classifiers and Meta-actions

Zbigniew W. Ras^{1,2} and Agnieszka Dardzińska³

¹ Univ. of North Carolina, Dept. of Computer Science, Charlotte, NC, 28223, USA

² Polish-Japanese Institute of Information Technology, 02-008 Warsaw, Poland

³ Bialystok Technical Univ., Dept. of Computer Science, 15-351 Bialystok, Poland
ras@uncc.edu, adardzin@uncc.edu

Abstract. Action rules describe possible transitions of objects from one state to another with respect to a distinguished attribute. Early research on action rule discovery usually required the extraction of classification rules before constructing any action rule. Newest algorithms discover action rules directly from a decision system. To our knowledge, all these algorithms assume that all attributes are symbolic or require prior discretization of all numerical attributes. This paper presents a new approach for generating action rules from datasets with numerical attributes by incorporating a tree classifier and a pruning step based on meta-actions. Meta-actions are seen as a higher-level knowledge (provided by experts) about correlations between different attributes.

1 Introduction

An action rule is a rule extracted from an information system that describes a possible transition of objects from one state to another with respect to a distinguished attribute called a decision attribute [13]. Attributes used to describe objects are partitioned into stable and flexible. Values of flexible attributes can be changed. This change can be influenced and controlled by users. Action rules mining initially was based on comparing profiles of two groups of targeted objects - those that are desirable and those that are undesirable [13]. An action rule was defined as a term $[(\omega) \wedge (\alpha \rightarrow \beta)] \Rightarrow (\phi \rightarrow \psi)$, where ω is the header and it is a conjunction of fixed classification features shared by both groups, $(\alpha \rightarrow \beta)$ represents proposed changes in values of flexible features, and $(\phi \rightarrow \psi)$ is a desired effect of the action. The discovered knowledge provides an insight of how values of some attributes need to be changed so the undesirable objects can be shifted to a desirable group. For example, one would like to find a way to improve his or her salary from a low-income to a high-income. Action rules tell us what changes within flexible attributes are needed to achieve that goal.

Meta-actions are defined as actions which trigger changes of flexible attributes either directly or indirectly because of correlations among certain attributes in the system. Links between meta-actions and changes they trigger within the values of flexible attributes can be defined by an ontology [3] or by a mapping linking meta-actions with changes of attribute values used to describe objects in the decision system. In medical domain, taking a drug is a classical example of a meta-action. For instance, Lamivudine is used for treatment of chronic hepatitis B. It improves the seroconversion of e-antigen

positive hepatitis B and also improves histology staging of the liver but at the same time it can cause a number of other symptoms. This is why doctors have to order certain lab tests to check patient's response to that drug. Clearly, the cost of a drug is known.

The concept of an action rule was proposed in [13] and investigated further in [18] [19] [8] [14] [4] [10] [17]. Paper [6] was probably the first attempt towards formally introducing the problem of mining action rules without pre-existing classification rules. Authors explicitly formulated it as a search problem in a support-confidence-cost framework. The proposed algorithm has some similarity with Apriori [1]. Their definition of an action rule allows changes on stable attributes. Changing the value of an attribute, either stable or flexible, is linked with a cost [19]. In order to rule out action rules with undesired changes on attributes, authors assigned very high cost to such changes. However, that way, the cost of action rules discovery is getting unnecessarily increased. Also, they did not take into account the correlations between attribute values which are naturally linked with the cost of rules used either to accept or reject a rule.

Algorithm *ARED*, presented in [7], is based on Pawlak's model of an information system S [9]. The goal was to identify certain relationships between granules defined by the indiscernibility relation on its objects. Some of these relationships uniquely define action rules for S . Paper [11] presents a strategy for discovering action rules directly from the decision system. Action rules are built from atomic expressions following a strategy similar to *LEERS* [5]. In [19], authors introduced the cost of action rules and use it in the pruning step of the rule discovery process. Paper [20] introduced the notion of *action* as a domain-independent way to model the domain knowledge. Given a data set about actionable features and an utility measure, a pattern is actionable if it summarizes a population that can be acted upon towards a more promising population observed with a higher utility. Algorithms for mining actionable patterns (changes within flexible attributes) take into account only numerical attributes. The distinguished (decision) attribute is called utility. Each action A_i triggers changes of attribute values described by terms [$a \downarrow$], [$b \uparrow$], and [c (don't know)]. They are represented as an influence matrix built by an expert. While previous approaches used only features - mined directly from the decision system, authors in [20] define actions as its foreign concepts. Influence matrix shows the link between actions and changes of attribute values and the same shows correlations between some attributes, i.e. if [$a \downarrow$], then [$b \uparrow$]. Clearly, expert does not know correlations between classification attributes and the decision attribute. Such correlations can be seen as action rules and they are discovered from the decision system. So, the definition of an action rule in [20] only refers to the increase/decrease of values of numerical attribute and the process of constructing action rules does not take into consideration neither their cost nor stable attributes.

This paper extends the definition of action rules [12] to numerical attributes and presents a new approach for discovering them from decision systems by incorporating a tree classifier, cost of action rules [19], and the pruning step based on meta-actions.

2 Background and Objectives

In this section we introduce the notion of an information system, meta-action and give examples.

By an information system [9] we mean a triple $S = (X, A, V)$, where:

1. X is a nonempty, finite set of objects
2. A is a nonempty, finite set of attributes, i.e.
 $a : U \longrightarrow V_a$ is a function for any $a \in A$, where V_a is called the domain of a
3. $V = \bigcup\{V_a : a \in A\}$.

For example, Table 1 shows an information system S with a set of objects $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$, set of attributes $A = \{a, b, c, d\}$, and a set of their values $V = \{a_1, a_2, b_1, b_2, b_3, c_1, c_2, d_1, d_2, d_3\}$.

Table 1. Information System S

	a	b	c	d
x_1	a_1	b_1	c_1	d_1
x_2	a_2	b_1	c_2	d_1
x_3	a_2	b_2	c_2	d_1
x_4	a_2	b_1	c_1	d_1
x_5	a_2	b_3	c_2	d_1
x_6	a_1	b_1	c_2	d_2
x_7	a_1	b_2	c_2	d_1
x_8	a_1	b_2	c_1	d_3

An information system $S = (X, A, V)$ is called a decision system, if one of the attributes in A is distinguished and called the decision. The remaining attributes in A are classification attributes. Additionally, we assume that $A = A_{St} \cup A_{Fl} \cup \{d\}$, where attributes in A_{St} are called *stable* whereas in A_{Fl} are called *flexible*. Attribute d is the decision attribute. “Date of birth” is an example of a stable attribute. “Interest rate” for each customer account is an example of a flexible attribute.

By meta-actions associated with S we mean higher level concepts representing actions introduced in [20]. Meta-actions, when executed, are expected to trigger changes in values of some flexible attributes in S as described by influence matrix [20]. To give an example, let us assume that classification attributes in S describe teaching evaluations at some school and the decision attribute represents their overall score. Examples of classification attributes are: *Explain difficult concepts effectively*, *Stimulate student interest in the course*, *Provide sufficient feedback*. Then, examples of meta-actions associated with S will be: *Change the content of the course*, *Change the textbook of the course*, *Post all material on the Web*. The influence matrix [20] is used to describe the relationship between meta-actions and the expected changes within classification attributes. It should be mentioned here that expert knowledge concerning meta-actions involves only classification attributes. Now, if some of these attributes are correlated with the decision attribute, then any change in their values will cascade to the decision attribute through this correlation. The goal of an action rule discovery is to identify all correlations between classification attributes and the decision attribute.

In earlier works in [13] [18] [19] [7] [14], action rules are constructed from classification rules. This means that we use pre-existing classification rules to construct action

rules either from certain pairs of these rules or from a single classification rule. For instance, algorithm *ARAS* [14] generates sets of terms (built from values of attributes) around classification rules and constructs action rules directly from them. In [12] authors presented a strategy for extracting action rules directly from a decision system and without using pre-existing classification rules.

In the next section, we introduce the notion of action sets, action rules [12], the cost of an action rule, and the notion of an influence matrix (see [20]) associated with a set of meta-actions. The values stored in an influence matrix are action sets.

3 Action Rules and Meta-actions

Let $S = (X, A, V)$ is an information system, where $V = \bigcup\{V_a : a \in A\}$. First, we modify the notion of an atomic action set given in [11] so it may include numerical attributes.

By an *atomic action set* we mean any of the three expressions:

1. $(a, a_1 \rightarrow a_2)$, where a is a symbolic attribute and $a_1, a_2 \in V_a$,
2. $(a, [a_1, a_2] \uparrow [a_3, a_4])$, where a is a numerical attribute, $a_1 \leq a_2 < a_3 \leq a_4$, and $(\forall)[(1 \leq i \leq 4) \rightarrow (a_i \in V_a)]$,
3. $(a, [a_1, a_2] \downarrow [a_3, a_4])$ where a is a numerical attribute, $a_3 \leq a_4 < a_1 \leq a_2$, and $(\forall)[(1 \leq i \leq 4) \rightarrow (a_i \in V_a)]$.

If a is symbolic and $a_1 = a_2$, then a is called *stable* on a_1 . Instead of $(a, a_1 \rightarrow a_1)$, we often write (a, a_1) for any $a_1 \in V_a$. The term $(a, a_1 \rightarrow a_2)$ should be read as “*the value of attribute a is changed from a_1 to a_2* ”. The term $(a, [a_1, a_2] \uparrow [a_3, a_4])$ should be read as “*the value of attribute a from to the interval $[a_1, a_2]$ is increased and it belongs now to the interval $[a_3, a_4]$* ”. Similarly, the term $(a, [a_1, a_2] \downarrow [a_3, a_4])$ should be read as “*the value of attribute a from to the interval $[a_1, a_2]$ is decreased and it belongs now to the interval $[a_3, a_4]$* ”.

Also, a simplified version of the atomic action set will be used. It includes such expressions as: $(a, \uparrow [a_3, a_4])$, $(a, [a_1, a_2] \uparrow)$, $(a, \downarrow [a_3, a_4])$, and $(a, [a_1, a_2] \downarrow)$.

Any collection of atomic action sets is called a *candidate action set*. If a candidate action set does not contain two atomic action sets referring to the same attribute, then it is called an *action set*. Clearly, $\{(b, b_2), (b, [b_1, b_2] \uparrow [b_3, b_4])\}$ is an example of a candidate action set which is not an action set. By the domain of action set t , denoted by $Dom(t)$, we mean the set of all attribute names listed in t . For instance if $\{(a, a_2), (b, b_1 \rightarrow b_2)\}$ is the action set, then its domain is equal to $\{a, b\}$. By an *action term* we mean a conjunction of atomic action sets forming an action set. There is some similarity between atomic action sets and atomic expressions introduced in [15], [20].

Now, assume that M_1 is a meta-action triggering the action set $\{(a, a_2), (b, b_1 \rightarrow b_2)\}$ and M_2 is a meta-action triggering the atomic actions in $\{(a, a_2), (b, b_2 \rightarrow b_1)\}$. It means that M_1 and M_2 involve attributes a, b with attribute a remaining stable. The corresponding action terms are: $(a, a_2) \cdot (b, b_1 \rightarrow b_2)$ associated with M_1 and $(a, a_2) \cdot (b, b_2 \rightarrow b_1)$ associated with M_2 .

Consider a set of meta-actions $\{M_1, M_2, \dots, M_n\}$ associated with a decision system $S = (X, A \cup \{d\}, V)$. Each meta-action M_i may trigger changes of some attribute values for objects in S . We assume here that $A - \{d\} = \{A_1, A_2, \dots, A_m\}$. The influence

of a meta-action M_i on attribute A_j in S is represented by an atomic action set $E_{i,j}$. The influence of meta-actions $\{M_1, M_2, \dots, M_n\}$ on the classification attributes in S is described by the influence matrix $\{E_{i,j} : 1 \leq i \leq n \wedge 1 \leq j \leq m\}$. There is no expert knowledge about what is their correlation with the decision attribute in S .

By *action rule* in S we mean any expression $r = [t_1 \Rightarrow (d, d_1 \rightarrow d_2)]$, where t_1 is an action set in S and d is the decision attribute. The domain of action rule r is defined as $Dom(t_1) \cup \{d\}$.

Now, we give an example of action rules assuming that an information system S is represented by Table 1, a, c, d are flexible attributes and b is stable. Expressions (a, a_2) , (b, b_2) , $(c, c_1 \rightarrow c_2)$, $(d, d_1 \rightarrow d_2)$ are examples of atomic action sets. Expression $r = [(a, a_2) \cdot (c, c_1 \rightarrow c_2)] \Rightarrow (d, d_1 \rightarrow d_2)$ is an example of an action rule. The rule says that if value a_2 remains unchanged and value c will change from c_1 to c_2 , then it is expected that the value d will change from d_1 to d_2 . The domain $Dom(r)$ of action rule r is equal to $\{a, c, d\}$. For simplicity reason, our example does not cover numerical attributes and the same we do not consider such terms as $(a, [a_1, a_2] \downarrow [a_3, a_4])$ or $(a, [a_1, a_2] \uparrow [a_3, a_4])$ which are also constructed by hierarchical classifiers.

Standard interpretation N_S of action terms in $S = (X, A, V)$ is defined as follow:

1. If $(a, a_1 \rightarrow a_2)$ is an atomic action set, then
 $N_S((a, a_1 \rightarrow a_2)) = [\{x \in X : a(x) = a_1\}, \{x \in X : a(x) = a_2\}]$.
2. If $(a, [a_1, a_2] \downarrow [a_3, a_4])$ is an atomic action set, then
 $N_S((a, [a_1, a_2] \downarrow [a_3, a_4])) = [\{x \in X : a_1 \leq a(x) \leq a_2\}, \{x \in X : a_3 \leq a(x) \leq a_4\}]$.
3. If $(a, [a_1, a_2] \uparrow [a_3, a_4])$ is an atomic action set, then
 $N_S((a, [a_1, a_2] \uparrow [a_3, a_4])) = [\{x \in X : a_1 \leq a(x) \leq a_2\}, \{x \in X : a_3 \leq a(x) \leq a_4\}]$.
4. If $(a, \uparrow [a_3, a_4])$ is an atomic action set, then
 $N_S((a, \uparrow [a_3, a_4])) = [\{x \in X : a(x) < a_3\}, \{x \in X : a_3 \leq a(x) \leq a_4\}]$.
5. If $(a, [a_1, a_2] \uparrow)$ is an atomic action set, then
 $N_S((a, [a_1, a_2] \uparrow)) = [\{x \in X : a_1 \leq a(x) \leq a_2\}, \{x \in X : a_2 < a(x)\}]$.
6. If $(a, \downarrow [a_3, a_4])$ is an atomic action set, then
 $N_S((a, \downarrow [a_3, a_4])) = [\{x \in X : a(x) < a_3\}, \{x \in X : a_3 \leq a(x) \leq a_4\}]$.
7. If $(a, [a_1, a_2] \downarrow)$ is an atomic action set, then
 $N_S((a, [a_1, a_2] \downarrow)) = [\{x \in X : a_1 \leq a(x) \leq a_2\}, \{x \in X : a_2 < a(x)\}]$.
8. If $t_1 = t_2 \cdot t$ is an action term, t_2 is an atomic action set, $N_S(t_2) = [Z_1, Z_2]$, and $N_S(t) = [Y_1, Y_2]$, then $N_S(t_1) = [Z_1 \cap Y_1, Z_2 \cap Y_2]$.

If t is an action rule and $N_S(t) = \{Y_1, Y_2\}$, then the support of t in S is defined as $sup(t) = \min\{card(Y_1), card(Y_2)\}$.

Now, let $r = [t_1 \Rightarrow t_2]$ is an action rule, where $N_S(t_1) = [Y_1, Y_2]$, $N_S(t_2) = [Z_1, Z_2]$. Support and confidence of r are defined as:

1. $sup(r) = \min\{card(Y_1 \cap Z_1), card(Y_2 \cap Z_2)\}$.
2. $conf(r) = [\frac{card(Y_1 \cap Z_1)}{card(Y_1)}] \cdot [\frac{card(Y_2 \cap Z_2)}{card(Y_2)}]$.

The definition of a confidence requires that $card(Y_1) \neq 0$ and $card(Y_2) \neq 0$. Otherwise, the confidence of an action rule is undefined.

Coming back to the example of S given in Table 1, we can find a number of action rules associated with S . Let us take $r = [(b, b_1) \cdot (c, c_1 \rightarrow c_2)] \Rightarrow (d, d_1 \rightarrow d_2)$ as an example of action rule. Then,

$$\begin{aligned} N_S((b, b_1)) &= [\{x_1, x_2, x_4, x_6\}, \{x_1, x_2, x_4, x_6\}], \\ N_S((c, c_1 \rightarrow c_2)) &= [\{x_1, x_4, x_8\}, \{x_2, x_3, x_5, x_6, x_7\}], \\ N_S((d, d_1 \rightarrow d_2)) &= [\{x_1, x_2, x_3, x_4, x_5, x_7\}, \{x_6\}], \\ N_S((b, b_1) \cdot (c, c_1 \rightarrow c_2)) &= [\{x_1, x_4\}, \{x_2, x_6\}]. \end{aligned}$$

Clearly, $sup(r) = 1$ and $conf(r) = 1 \cdot 1 = 1/2$.

The notion of a cost associated with an action rule was introduced in [19]. Some changes of values of attributes are not expensive and easy to achieve but some of them might be very costly. So, with every atomic action set t , we associate the cost $cost_S(t)$ needed to achieve the change of attribute value recommended in t for objects in S . If all atomic actions sets listed in $t = t_1 \cdot t_2 \cdot t_3 \cdot \dots \cdot t_k$ are not correlated, then $cost_S(t) = \sum \{cost_S(t_i) : 1 \leq i \leq k\}$.

4 Action Rules Discovery

In this section we present the process of discovering action rules of acceptable cost from a decision system $S = (X, A \cup \{d\}, V)$ using a tree classifier and an influence matrix associated with meta-actions.

To reduce the number of values for numerical attributes in S we use a classical method based either on entropy or Gini index resulting in a hierarchical discretization. Classification attributes are partitioned into stable and flexible. Before we use any flexible attribute in the process of a decision tree construction, all stable attributes have to be used first. This way the decision table is split into a number of decision subtables leading to them from the root of the tree by uniquely defined paths built from stable attributes [18]. Each path defines a header in all action rules extracted from the corresponding subtable. Initial testing shows that the action rules built that way are more compact (have larger intervals) than action rules built with prior discretization of the decision table done for instance by Rough Sets Exploration System [16].

For instance, let us assume that the table assigned to the root of the tree in Fig. 1 has to be converted into its decision tree representation and that we are looking for action rules of which purpose is to change the property of objects from d_3 into d_1 . We also assume that both attributes in $\{a, b\}$ are stable and attributes in $\{c, e, f\}$ are flexible. Attribute $\{d\}$ is the decision attribute. Our goal is to split this table into sub-tables by taking a stable attribute with the largest entropy gain as the splitting one. In our example, we chose attribute a .

This process is recursively continued for all stable attributes (in our example only one stable attribute is left). The sub-tables corresponding to outgoing edges from the root node which are labelled by $a = 10$, $a = 3$ are removed because they do not contain decision value d_1 . After all stable attributes are used, we begin splitting recursively each of the latest sub-tables by taking a flexible attribute with the largest entropy gain as the splitting one. In our example, the following paths (from the root to a leaf) are finally

Stable: {a, b}
 Flexible: {c, e, f}
 Decision Feature: {d}
 Reclassification Direction : 3 → 1

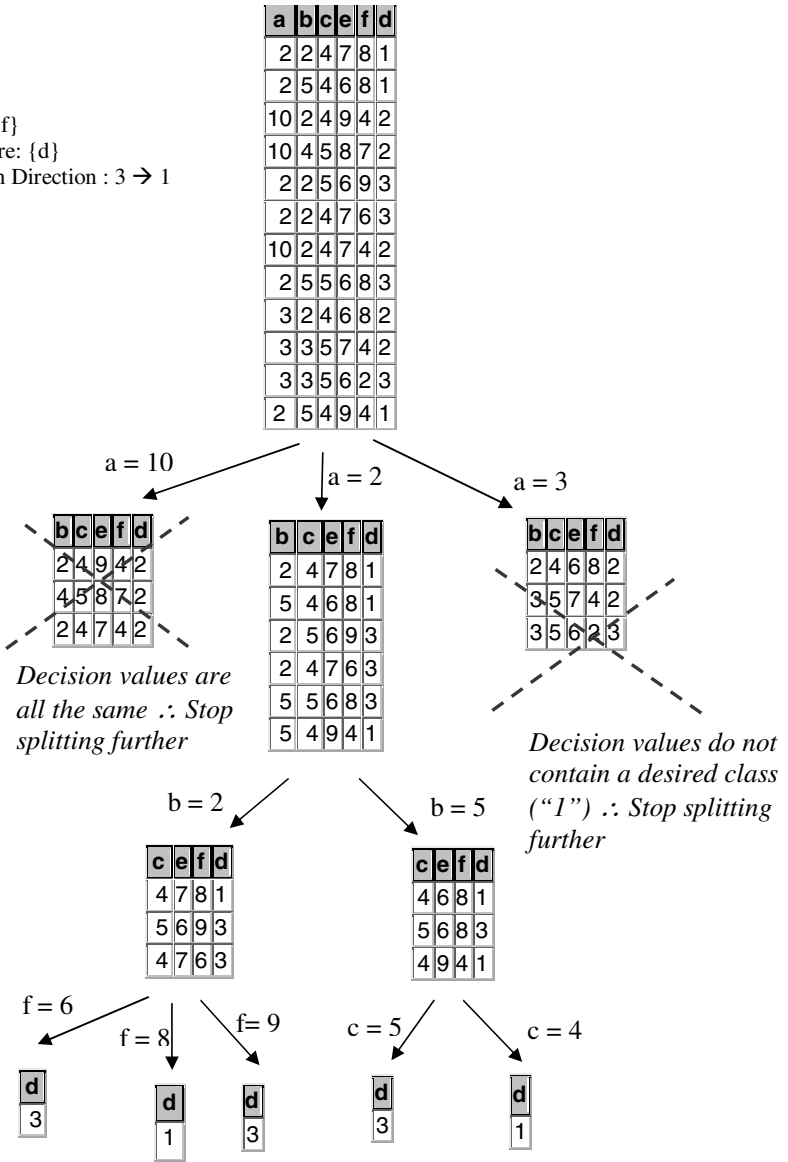


Fig. 1. Classification Tree Construction Process

built: $[[[a, 2) \wedge (b, 2)] \wedge (f, 6) \wedge (d, 3)], [[(a, 2) \wedge (b, 2)] \wedge (f, 8) \wedge (d, 1)], [[(a, 2) \wedge (b, 2)] \wedge (f, 9) \wedge (d, 3)], [[(a, 2) \wedge (b, 5)] \wedge (c, 5) \wedge (d, 3)], [[(a, 2) \wedge (b, 2)] \wedge (c, 4) \wedge (d, 1)].$

By examining the nodes right above the leaves of the resulting decision tree, we get the following candidate action rules (based on strategy similar to DEAR [18]):

- $[[[(a, 2) \wedge (b, 2)] \wedge (f, 6 \rightarrow 8)] \Rightarrow (d, 3 \rightarrow 1)],$
- $[[[(a, 2) \wedge (b, 2)] \wedge (f, 9 \rightarrow 8)] \Rightarrow (d, 3 \rightarrow 1)],$
- $[[[(a, 2) \wedge (b, 5)] \wedge (c, 5 \rightarrow 4)] \Rightarrow (d, 3 \rightarrow 1)].$

Now, we can calculate the cost of each of these candidate action rules and delete all with cost below a user specified threshold.

Influence matrix associated with S and a set of meta-actions is used to identify which remaining candidate action rules are valid with respect to meta-actions and hidden correlations between classification attributes and the decision attribute.

Assume that $S = (X, A \cup \{d\}, V)$ is a decision system, $A - \{d\} = A_1 \cup A_2 \cup \dots \cup A_m$, $\{M_1, M_2, \dots, M_n\}$ are meta-actions associated with S , $\{E_{i,j} : 1 \leq i \leq n, 1 \leq j \leq m\}$ is the influence matrix, and $r = [(A_{[i,1]}, a_{[i,1]} \rightarrow a_{[j,1]}) \cdot (A_{[i,2]}, a_{[i,2]} \rightarrow a_{[j,2]}) \cdot \dots \cdot (A_{[i,k]}, a_{[i,k]} \rightarrow a_{[j,k]})] \Rightarrow (d, d_i \rightarrow d_j)$ is a candidate action rule extracted from S . Also, we assume here that $A_{[i,j]}(M_i) = E_{i,j}$. Value $E_{i,j}$ is either an atomic action set or $NULL$. By meta-actions based decision system, we mean a triple consisting with S , meta-actions associated with S , and the influence matrix linking them.

We say that r is valid in S with respect to meta-action M_i , if the following condition holds:

$$\begin{aligned} &\text{if } (\exists p \leq k)[A_{[i,p]}(M_i) \text{ is defined}], \text{ then} \\ &(\forall p \leq k)[\text{if } A_{[i,p]}(M_i) \text{ is defined, then } (A_{[i,p]}, a_{[i,p]} \rightarrow a_{[j,p]}) = (A_{[i,p]}, E_{i,p})] \end{aligned}$$

We say that r is valid in S with respect to meta-actions $\{M_1, M_2, \dots, M_n\}$, if there is i , $1 \leq i \leq n$, such that r is valid in S with respect to meta-action M_i .

To give an example, assume that S is a decision system represented by Table 1 and $\{M_1, M_2, M_3, M_4, M_5, M_6\}$ is the set of meta-actions assigned to S with an influence matrix shown in Table 2. Clearly, each empty slot in Table 2 corresponds to $NULL$ value.

Table 2. Influence Matrix for S

	a	b	c
M_1		b_1	$c_2 \rightarrow c_1$
M_2	$a_2 \rightarrow a_1$	b_2	
M_3	$a_1 \rightarrow a_2$		$c_2 \rightarrow c_1$
M_4		b_1	$c_1 \rightarrow c_2$
M_5			$c_1 \rightarrow c_2$
M_6	$a_1 \rightarrow a_2$		$c_1 \rightarrow c_2$

In the example presented in previous section, two candidate action rules have been constructed:

$$\begin{aligned} r1 &= [[(b, b_1) \cdot (c, c_1 \rightarrow c_2)] \Rightarrow (d, d_1 \rightarrow d_2)] \text{ and} \\ r2 &= [(a, a_2 \rightarrow a_1) \Rightarrow (d, d_1 \rightarrow d_2)]. \end{aligned}$$

Clearly $r1$ is valid in S with respect to M_4 and M_5 . Also, $r2$ is valid in S with respect to M_1, M_4, M_5 because there is no overlap between the domain of action rule $r2$ and the

set of attributes influenced by any of these meta-actions. However, we can not say that r_2 is valid in S with respect to M_2 since b_2 is not listed in the classification part of r_2 .

Assume that $S = (X, A \cup \{d\}, V)$ is a decision system with meta-actions $\{M_1, M_2, \dots, M_n\}$ associated with S . Any candidate action rule extracted from S which is valid in a meta-actions based decision system is called action rule. So, the process of action rules discovery is simplified to checking the validity of candidate action rules.

5 Conclusion

New algorithm for action rules discovery from the data containing both symbolic and numerical attributes is presented. The method basically follows *C4.5* or *CART* with only one exception - before any flexible attribute is used as a splitting one, all stable attributes have to be processed first. Each path starting from the root and built from stable attributes defines a class of candidate action rules having the same heading. Meta-actions jointly with the influence matrix are used as a postprocessing tool in action rules discovery. Influence matrix shows the correlations among classification attributes triggered off by meta-actions. If the candidate actions rules are not on par with them, then they are not classified as action rules. However, the influence matrix may not show all the interactions between classification attributes, so still some of the resulting action rules may fail when tested on real data.

Acknowledgment

This work was supported by the Ministry of Science and Higher Education in Poland under Grant *N N519 404734*, by Bialystok Technical University under Grant *W/WI/2/09*, and by Polish-Japanese Institute of Information Technology under Grant *ST/SI/02/08*.

References

1. Agrawal, R., Srikant, R.: Fast algorithm for mining association rules. In: Proceeding of the Twentieth International Conference on VLDB, pp. 487–499 (1994)
2. Dardzińska, A., Raś, Z.: Extracting rules from incomplete decision systems. In: Foundations and Novel Approaches in Data Mining. Studies in Computational Intelligence, vol. 9, pp. 143–154. Springer, Heidelberg (2006)
3. Fensel, D.: Ontologies: a silver bullet for knowledge management and electronic commerce. Springer, Heidelberg (1998)
4. Greco, S., Matarazzo, B., Pappalardo, N., Slowiński, R.: Measuring expected effects of interventions based on decision rules. *J. Exp. Theor. Artif. Intell.* 17(1-2), 103–118 (2005)
5. Grzymala-Busse, J.: A new version of the rule induction system LERS. *Fundamenta Informaticae* 31(1), 27–39 (1997)
6. He, Z., Xu, X., Deng, S., Ma, R.: Mining action rules from scratch. *Expert Systems with Applications* 29(3), 691–699 (2005)
7. Im, S., Raś, Z.W.: Action rule extraction from a decision table: ARED, in Foundations of Intelligent Systems. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) Foundations of Intelligent Systems. LNCS (LNAI), vol. 4994, pp. 160–168. Springer, Heidelberg (2008)

8. Im, S., Raś, Z.W., Wasyluk, H.: Action Rule Discovery From Incomplete Data. *Knowledge and Information Systems Journal* (will appear, 2009)
9. Pawlak, Z.: Information systems - theoretical foundations. *Information Systems Journal* 6, 205–218 (1981)
10. Qiao, Y., Zhong, K., Wang, H.-A., Li, X.: Developing event-condition-action rules in real-time active database. In: *Proceedings of the 2007 ACM symposium on Applied computing*, pp. 511–516. ACM, New York (2007)
11. Raś, Z.W., Dardzińska, A.: Action rules discovery without pre-existing classification rules. In: Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) *RSCTC 2008*. LNCS (LNAI), vol. 5306, pp. 181–190. Springer, Heidelberg (2008)
12. Raś, Z.W., Dardzińska, A., Tsay, L.-S., Wasyluk, H.: Association Action Rules. In: *Proceedings of IEEE/ICDM Workshop on Mining Complex Data (MCD 2008)*, Pisa, Italy. IEEE Computer Society, Los Alamitos (2008)
13. Raś, Z.W., Wieczorkowska, A.: Action-Rules: How to increase profit of a company. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) *PKDD 2000*. LNCS (LNAI), vol. 1910, pp. 587–592. Springer, Heidelberg (2000)
14. Raś, Z., Wyrzykowska, E., Wasyluk, H.: ARAS: Action rules discovery based on agglomerative strategy. In: Raś, Z.W., Tsumoto, S., Zighed, D.A. (eds.) *MCD 2007*. LNCS (LNAI), vol. 4944, pp. 196–208. Springer, Heidelberg (2008)
15. Rauch, J.: Considerations on logical calculi for dealing with knowledge in data mining. In: *Advances in Data Management. Studies in Computational Intelligence*. LNCS, vol. 223. Springer, Heidelberg (will appear, 2009)
16. <http://logic.mimuw.edu.pl/~rses/>
17. Suzuki, E.: Discovering Action Rules that are Highly Achievable from Massive Data. In: *Proceedings of PAKDD 2009*. LNCS (LNAI), vol. 5476, pp. 713–722. Springer, Heidelberg (2009)
18. Tsay, L.-S., Raś, Z.W.: Action rules discovery system DEAR3. In: Esposito, F., Raś, Z.W., Malerba, D., Semeraro, G. (eds.) *ISMIS 2006*. LNCS (LNAI), vol. 4203, pp. 483–492. Springer, Heidelberg (2006)
19. Tzacheva, A., Tsay, L.S.: Tree-based construction of low-cost action rules. *Fundamenta Informaticae Journal* 86(1-2), 213–225 (2008)
20. Wang, K., Jiang, Y., Tuzhilin, A.: Mining Actionable Patterns by Role Models. In: *Proceedings of the 22nd International Conference on Data Engineering*, pp. 16–25. IEEE Computer Society, Los Alamitos (2006)

Action Rules and the GUHA Method: Preliminary Considerations and Results

Jan Rauch and Milan Šimůnek

Faculty of Informatics and Statistics, University of Economics, Prague*
nám W. Churchilla 4, 130 67 Prague, Czech Republic
rauch@vse.cz, simunek@vse.cz

Abstract. The paper presents an alternative approach to action rules. The presented approach is based on experience with the GUHA method and the LISp-Miner system. G-action rules are introduced. First experience with new GUHA procedure Ac4ft-Miner that mines for G-action rules is described.

1 Introduction

Action rules present a promising direction in data mining research. The notion of action rules was proposed in [4]. The basic idea of action rules is to suggest a way to re-classify objects (for instance customers) to a desired state. The action rules are based on stable and flexible attributes [4]. An example of a *stable attribute* is the date of birth. An example of a *flexible attribute* is an interest rate on a customer account that depends on a bank. Action rules suggest a way how to change the values of flexible attributes to get a desired state. An example is a suggestion to change the interest rate to decrease customer attrition. There are various approaches to mine action rules, see e.g. [4,5,7].

This paper introduces an approach to action rules based on the GUHA method and its implementation in the LISp-Miner system [10,11]. This approach offers new ways how to take an action to get an advantage.

The paper is organized as follows. The action rules are introduced in section 2. Important features of the GUHA method and the LISp-Miner system are summarized in section 3. Definition of action rules based on these features is given in section 4. We call such action rules *G-action rules*. First experience with a new GUHA procedure *Ac4ft-Miner* that mines for G-action rules is presented in section 5. Conclusions and description of further work are in section 6.

2 Action Rules

We start with action rules defined in [5]. There are various additional approaches to action rules, some of them are closely related to the approach introduced in

* The work described here has been supported by Grant No. 201/08/0802 of the Czech Science Foundation and by Grant No. ME913 of Ministry of Education, Youth and Sports, of the Czech Republic.

[5]. An overview paper related to action rules is [6], an example of another paper is [7].

Action rules are in [5] defined on the basis of an information system $\mathcal{S} = (\mathcal{U}, \mathcal{A})$ where \mathcal{U} is a nonempty, finite set of objects and \mathcal{A} is a nonempty, finite set of attributes. It means that each $A \in \mathcal{A}$ is a function $\mathcal{U} \rightarrow V_A$ where V_A is a domain of A . A special type of information system is called *decision table*. It is any information system $\mathcal{S} = (\mathcal{U}, \mathcal{A}_{St} \cup \mathcal{A}_{Fl} \cup \{D\})$ where $D \notin \mathcal{A}_{St} \cup \mathcal{A}_{Fl}$ is a distinguished attribute called a decision and the set \mathcal{A} of attributes is partitioned into stable conditions \mathcal{A}_{St} and flexible conditions \mathcal{A}_{Fl} . Action rule R in \mathcal{S} is an expression

$$(A_1 = \omega_1) \wedge \dots \wedge (A_q = \omega_q) \wedge (B_1, \alpha_1 \rightarrow \beta_1) \wedge \dots \wedge (B_p, \alpha_p \rightarrow \beta_p) \Rightarrow (D, k_1 \rightarrow k_2)$$

where $\{B_1, \dots, B_p\}$ are flexible attributes and $\{A_1, \dots, A_q\}$ are stable in \mathcal{S} . Moreover, it is assumed that $\omega_i \in Dom(A_i)$, $i = 1, \dots, q$ and $\alpha_i, \beta_i \in Dom(B_i)$, $i = 1, \dots, p$. The term $(A_i = \omega_i)$ means that the value of the attribute A_i is ω_i . The term $(B_j, \alpha_j \rightarrow \beta_j)$ means that the value of the attribute B_j has been changed from α_j to β_j , similarly for $(D, k_1 \rightarrow k_2)$.

The left hand side pattern of the above action rule is the set $P_L = V_L \cup \{k_1\}$ where $V_L = \{\omega_1, \dots, \omega_q, \alpha_1, \dots, \alpha_p\}$. The domain $Dom_{\mathcal{S}}(V_L)$ of P_L is a set of objects in \mathcal{S} that exactly match V_L . $Card[Dom_{\mathcal{S}}(V_L)]$ is the number of objects in $Dom_{\mathcal{S}}(V_L)$, $Card[Dom_{\mathcal{S}}(P_L)]$ is the number of objects that exactly match P_L , and $Card[\mathcal{U}]$ is the total number of objects in \mathcal{S} . The left support $supL(R)$ of the action rule R is defined as $supL(R) = Card[Dom_{\mathcal{S}}(P_L)]/Card[\mathcal{U}]$.

The right support $supR(R)$ of the action rule R is defined analogously i.e. $supR(R) = Card[Dom_{\mathcal{S}}(P_R)]/Card[\mathcal{U}]$ where $P_R = V_R \cup \{k_2\}$ and V_R is defined as $V_R = \{\omega_1, \dots, \omega_q, \beta_1, \dots, \beta_p\}$.

The *support of the action rule* R in \mathcal{S} is denoted by $Sup_{\mathcal{S}}(R)$ and it is the same as the left support $supL(R)$. The *confidence* of the action rule R in \mathcal{S} is denoted by $Conf_{\mathcal{S}}(R)$ and it is defined as

$$(Card[Dom_{\mathcal{S}}(P_L)]/Card[Dom_{\mathcal{S}}(V_L)]) * (Card[Dom_{\mathcal{S}}(P_R)]/Card[Dom_{\mathcal{S}}(V_R)]) .$$

An algorithm for mining of such action rules is described in [5] together with discussion of additional approaches.

3 The GUHA Method and the LISP-Miner System

GUHA is a method of exploratory data analysis developed since 1960's [2]. Its goal is to offer all interesting facts following from the analyzed data to the given problem. GUHA is realized by GUHA-procedures. Input of a GUHA-procedure consists of the analyzed data and of a simple definition of a usually very large set of relevant (i.e. potentially interesting) patterns. The procedure generates each particular pattern and tests if it is true in the analyzed data. The output of the procedure consists of all prime patterns. The pattern is prime if it is true in the analyzed data and if it does not immediately follow from the other more simple output patterns [3].

The most important GUHA procedure is the procedure ASSOC [3]. It mines for patterns that can be understood as association rules, they are however more general than the "classical" association rules defined in [1]. The probably most used implementation of the ASSOC procedure is the procedure *4ft-Miner*. It has various new important features and it mines also for conditional association rules [10].

Implementations of the procedure ASSOC are based on *representation of analyzed data by strings of bits* [8,10], the well known apriori algorithm [1] is not used. A system of modules for dealing with strings of bits was developed [13]. Their utilization leads to algorithm with the complexity linearly dependent on the number of rows of the analyzed data matrices [10]. These modules were used to implement five additional GUHA procedures, all of them are included in the LISp-Miner system [11]. This paper describes a new GUHA procedure *Ac4ft-Miner* that mines for patterns that can be understood as an enhancement of action rules introduced in Section 2.

All the GUHA procedures implemented in the LISp-Miner system deal with data matrices. An example of the data matrix is the data matrix \mathcal{M} shown in Fig. 1.

object	stable attributes			flexible attributes			examples of basic Boolean attributes			
	A_1	...	A_Q	B_1	...	B_P	D	$A_1(2)$	$B_1(9, 12)$	$D(5, 7, 9)$
o_1	6	...	4	12	...	9	7	0	1	1
o_2	13	...	2	9	...	5	3	0	1	0
\vdots	\vdots	\ddots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\vdots	\vdots
o_n	2	...	2	8	...	5	6	1	0	0

Fig. 1. Data matrix \mathcal{M} with stable and flexible attributes

Rows of a data matrix correspond to observed objects (e.g. clients of a bank), columns correspond to attributes describing properties of particular objects (e.g. date of birth or interest rate on a customer account), possible values of attributes are called categories. Data matrix \mathcal{M} in Fig. 1 has attributes $A_1, \dots, A_P, B_1, \dots, B_Q, D$. Each data matrix corresponds to an information system $\mathcal{S} = (\mathcal{U}, \mathcal{A})$ in a clear way. In the case of data matrix \mathcal{M} from the Fig. 1 it is $\mathcal{U} = \{o_1, \dots, o_n\}$ etc. If we consider also information on stable and flexible attributes as given in first row of Fig. 1 then the data matrix \mathcal{M} can be seen as a decision table $\mathcal{S} = (\mathcal{U}, \mathcal{A}_{St} \cup \mathcal{A}_{Fl} \cup \{D\})$ where $\mathcal{A}_{St} = \{A_1, \dots, A_P\}$ and $\mathcal{A}_{Fl} = \{B_1, \dots, B_Q\}$.

The procedure *Ac4ft-Miner* was derived from the procedure *4ft-Miner* that mines for association rules of the form $\varphi \approx \psi$ where φ and ψ are Boolean attributes. The Boolean attribute φ is called *antecedent* and ψ is called *succedent*. The association rule $\varphi \approx \psi$ means that φ and ψ are associated in the way given by the symbol \approx . The symbol \approx is called the *4ft-quantifier*. It corresponds to a condition concerning a four-fold contingency table of φ and ψ . Various types of

dependencies of φ and ψ can be expressed by 4ft-quantifiers. The rule $\varphi \approx \psi$ is *true in data matrix* \mathcal{M} if the condition corresponding to the 4ft-quantifier is satisfied in the four-fold contingency table of φ and ψ in \mathcal{M} , otherwise $\varphi \approx \psi$ is *false in data matrix* \mathcal{M} .

The four-fold contingency table of φ and ψ in data matrix \mathcal{M} is a quadruple $\langle a, b, c, d \rangle$ of natural numbers such that a is the number of rows of \mathcal{M} satisfying both φ and ψ , b is the number of rows of \mathcal{M} satisfying φ and not satisfying ψ , etc., see Table [1](#). The four-fold contingency table (the *4ft table*) of φ and ψ in \mathcal{M} is denoted by $4ft(\varphi, \psi, \mathcal{M})$.

Table 1. 4ft table $4ft(\varphi, \psi, \mathcal{M})$ of φ and ψ in \mathcal{M}

\mathcal{M}	ψ	$\neg\psi$
φ	a	b
$\neg\varphi$	c	d

There are 14 basic 4ft-quantifiers implemented in the 4ft-Miner procedure, it is possible to use also conjunctions of basic 4ft-quantifiers. A simple example is the 4ft-quantifier $\Rightarrow_{p,B}$ of *founded implication* [3](#) that is defined for $0 < p \leq 1$ and $B > 0$ by the condition $\frac{a}{a+b} \geq p \wedge a \geq B$. The association rule $\varphi \Rightarrow_{p,B} \psi$ means that at least $100p$ percent of rows of \mathcal{M} satisfying φ satisfy also ψ and that there are at least B rows of \mathcal{M} satisfying both φ and ψ .

The Boolean attributes φ and ψ are derived from the columns of data matrix \mathcal{M} . We assume there is a finite number of possible values for each column of \mathcal{M} . *Basic Boolean attributes* are created first. The basic Boolean attribute is an expression of the form $A(\kappa)$ where $\kappa \subset \{a_1, \dots, a_k\}$ and $\{a_1, \dots, a_k\}$ is the set of all possible values of the column A . The basic Boolean attribute $A(\kappa)$ is true in row o of \mathcal{M} if it is $a \in \alpha$ where a is the value of the attribute A in row o . The set κ is called a *coefficient* of basic Boolean attribute $A(\kappa)$. For example $\{9,12\}$ is a coefficient of the basic Boolean attribute $B_1(9,12)$. There are examples of values of basic Boolean attributes in Fig. [1](#); the value 1 means *true* and the value 0 means *false*. Boolean attributes φ and ψ are derived from basic Boolean attributes using propositional connectives \vee , \wedge and \neg in the usual way.

The input of the procedure *4ft-Miner* consists of the analyzed data matrix and of several parameters defining (usually very) large set of association rules to be verified. There are very fine tools to define this set [10](#), some of them are introduced in Sect. [5](#).

4 G-Action Rules

A G-action rule \mathcal{R} is an expression of the form $\varphi_{St} \wedge \Phi_{Chg} \approx^* \psi_{St} \wedge \Psi_{Chg}$ where φ_{St} is a Boolean attribute called *stable antecedent*, Φ_{Chg} is an expression called *change of antecedent*, ψ_{St} is a Boolean attribute called *stable succedent*, Ψ_{Chg} is an expression called *change of succedent*, and \approx^* is a symbol called

Ac4ft-quantifier. An example of the Ac4ft-quantifier is the symbol \Rightarrow_{p,B_1,B_2} . An example of the G-action rule is the expression

$$A_1(\kappa_1) \wedge A_2(\kappa_2) \wedge [B_1(\lambda_1) \rightarrow B_1(\lambda'_1)] \Rightarrow_{p_1 \rightarrow p_2, B_1, B_2} A_3(\kappa_3) \wedge [B_2(\lambda_2) \rightarrow B_2(\lambda'_2)] .$$

Both the *change of antecedent* Φ_{Chg} and the *change of succedent* Ψ_{Chg} are built from *changes of coefficient*. The *change of coefficient* is an expression $[Z(\kappa) \rightarrow Z(\kappa')]$ where both $Z(\kappa_1)$ and $Z(\kappa_1)$ are literals with coefficients κ and κ' respectively such that $\kappa \cap \kappa' = \emptyset$. The *change of Boolean attribute* is created from *changes of coefficient* and Boolean connectives in the same way as the Boolean attribute is created from the literals, see also Section 3. Both the *change of antecedent* and the *change of succedent* are *changes of Boolean attribute*.

If $\Lambda = [Z(\kappa) \rightarrow Z(\kappa')]$ is a change of coefficient, then an *initial state* $\mathcal{I}(\Lambda)$ of Λ is defined as $\mathcal{I}(\Lambda) = Z(\kappa)$ and a *final state* $\mathcal{F}(\Lambda)$ of Λ is defined as $\mathcal{F}(\Lambda) = Z(\kappa')$. If Φ is a change of Boolean attribute, then *initial state* $\mathcal{I}(\Phi)$ of Φ is a Boolean attribute that we get by replacing all changes of literals Λ occurring in Φ by their initial states $\mathcal{I}(\Lambda)$. The *final state* $\mathcal{F}(\Phi)$ of Φ is a Boolean attribute that we get by replacing all changes of literals Λ occurring in Φ by their final states $\mathcal{F}(\Lambda)$. Two examples: $\mathcal{I}(A_1(\kappa_1) \wedge A_2(\kappa_2) \wedge [B_1(\lambda_1) \rightarrow B_1(\lambda'_1)]) = A_1(\kappa_1) \wedge A_2(\kappa_2) \wedge B_1(\lambda_1)$ and $\mathcal{F}(A_3(\kappa_3) \wedge [B_2(\lambda_2) \rightarrow B_2(\lambda'_2)]) = A_3(\kappa_3) \wedge B_2(\lambda'_2)$.

The attributes used in Φ_{Chg} are called *independent attributes* of \mathcal{R} and the attributes used in Ψ_{Chg} are called *dependent attributes* of \mathcal{R} . The action rule $\mathcal{R} : \varphi_{St} \wedge \Phi_{Chg} \approx^* \psi_{St} \wedge \Psi_{Chg}$ says what happen with objects satisfying stable conditions φ_{St} and ψ_{St} when we change values of their flexible independent attributes in the way given by Φ_{Chg} – i.e. from the initial state characterized by the Boolean attribute $\mathcal{I}(\Phi_{Chg})$ into a final state characterized by the Boolean attribute $\mathcal{F}(\Phi_{Chg})$. The effect is described by two association rules \mathcal{R}_I and \mathcal{R}_F :

$$\mathcal{R}_I : \varphi_{St} \wedge \mathcal{I}(\Phi_{Chg}) \approx_I \psi_{St} \wedge \mathcal{I}(\Psi_{Chg}) \quad \mathcal{R}_F : \varphi_{St} \wedge \mathcal{F}(\Phi_{Chg}) \approx_F \psi_{St} \wedge \mathcal{F}(\Psi_{Chg}) .$$

The first rule \mathcal{R}_I characterizes the initial state. The second rule \mathcal{R}_F describes the final state induced by the change of the independent flexible attributes property (i.e. Boolean attribute) $\mathcal{I}(\Phi_{Chg})$ to $\mathcal{F}(\Phi_{Chg})$. If we denote $\varphi_{St} \wedge \mathcal{I}(\Phi_{Chg})$ as φ_I , $\psi_{St} \wedge \mathcal{I}(\Psi_{Chg})$ as ψ_I , $\varphi_{St} \wedge \mathcal{F}(\Phi_{Chg})$ as φ_F , and $\psi_{St} \wedge \mathcal{F}(\Psi_{Chg})$ as ψ_F then the rules \mathcal{R}_I and \mathcal{R}_F can be written as

$$\mathcal{R}_I : \quad \varphi_I \approx_I \psi_I \qquad \mathcal{R}_F : \quad \varphi_F \approx_F \psi_F .$$

The action rule \mathcal{R} makes possible to see the effect of the change Φ_{Chg} in three steps: (1) φ_F is true instead of φ_I . (2) The values of dependent flexible attributes are changed such that $\mathcal{F}(\Psi_{Chg})$ is true instead of $\mathcal{I}(\Psi_{Chg})$ and thus ψ_F is true instead of ψ_I . (3) The initial relation of $\varphi_I \approx_I \psi_I$ described by the 4ft-quantifier \approx_I is changed to the final relation $\varphi_F \approx_F \psi_F$ described by the 4ft-quantifier \approx_F .

The truthfulness of the G-action rule $\mathcal{R} : \varphi_{St} \wedge \Phi_{Chg} \approx^* \psi_{St} \wedge \Psi_{Chg}$ is defined on the basis of this interpretation of the change Φ_{Chg} . The G-action rule \mathcal{R} is true in the analyzed data matrix \mathcal{M} if the condition corresponding to the *Ac4ft-quantifier* \approx^* is satisfied in the data matrix \mathcal{M} . The sense of the rule \mathcal{R} is

expressed by the rules \mathcal{R}_I and \mathcal{R}_F described above. Thus the condition related to \approx^* and defining the truthfulness of \mathcal{R} is related to a way in which the rules \mathcal{R}_I and \mathcal{R}_F are evaluated. The rule \mathcal{R}_I is evaluated on the basis of 4ft-table $4ft(\varphi_I, \psi_I, \mathcal{M})$ of φ_I and ψ_I in \mathcal{M} and the rule \mathcal{R}_F is evaluated on the basis of 4ft-table $4ft(\varphi_F, \psi_F, \mathcal{M})$ of φ_F and ψ_F in \mathcal{M} , see Fig. 2.

\mathcal{M}	ψ_I	$\neg\psi_I$	\mathcal{M}	ψ_F	$\neg\psi_F$
φ_I	a_I	b_I	φ_F	a_F	b_F
$\neg\varphi_I$	c_I	d_I	$\neg\varphi_F$	c_F	d_F
4ft-table $4ft(\varphi_I, \psi_I, \mathcal{M})$			4ft-table $4ft(\varphi_F, \psi_F, \mathcal{M})$		

Fig. 2. The 4ft-tables $4ft(\varphi_I, \psi_I, \mathcal{M})$ and $4ft(\varphi_F, \psi_F, \mathcal{M})$

An example of Ac4ft-quantifier \approx^* is the Ac4ft-quantifier $\Rightarrow_{q, B_1, B_2}^{I>F}$ defined by the condition $\frac{a_I}{a_I + b_I} - \frac{a_F}{a_F + b_F} \geq q \wedge a_I \geq B_1 \wedge a_F \geq B_2$. It is assumed that $0 < q \leq 1$, $B_1 > 0$, and $B_2 > 0$.

If the action rule $\mathcal{R} = \varphi_{St} \wedge \Phi_{Chg} \Rightarrow_{p, B_1, B_2} \psi_{St} \wedge \Psi_{Chg}$ with Ac4ft-quantifier $\Rightarrow_{q, B_1, B_2}^{I>F}$ is true in the data matrix \mathcal{M} then its effect can be expressed by two association rules \mathcal{R}_I and \mathcal{R}_F with 4ft-quantifiers $\Rightarrow_{=, p+q, B_1}$ and $\Rightarrow_{=, p, B_2}$:

$$\mathcal{R}_I: \quad \varphi_I \Rightarrow_{=, p, B_1} \psi_I \qquad \mathcal{R}_F: \quad \varphi_F \Rightarrow_{=, p+q, B_2} \psi_F$$

where $\varphi_{St} \wedge \mathcal{I}(\Phi_{Chg})$ is denoted by φ_I , $\psi_{St} \wedge \mathcal{I}(\Psi_{Chg})$ by ψ_I , $\varphi_{St} \wedge \mathcal{F}(\Phi_{Chg})$ by φ_F , and $\psi_{St} \wedge \mathcal{F}(\Psi_{Chg})$ by ψ_F , see also above. The 4ft-quantifier $\Rightarrow_{=, p, B}$ is derived from the 4ft-quantifier $\Rightarrow_{p, B}$, $\Rightarrow_{=, p, B}$ is defined for $0 < p \leq 1$ and $B > 0$ by the condition $\frac{a}{a+b} = p \wedge a = B$, see Tab. 1. Remember that 4ft-quantifier $\Rightarrow_{p, B}$ is defined by the condition $\frac{a}{a+b} \geq p \wedge a \geq B$, see Sect. 3. The parameter p in $\Rightarrow_{=, p+q, B_1}$ and $\Rightarrow_{=, p, B_2}$ depends on the data matrix \mathcal{M} .

Informally speaking, the Ac4ft-quantifier $\Rightarrow_{q, B_1, B_2}^{I>F}$ expresses the fact that the confidence of the rule \mathcal{R}_F is smaller than the confidence of the rule \mathcal{R}_I . It is suitable when the truthfulness of the attribute ψ_{St} is undesirable. In the case when the truthfulness of the attribute ψ_{St} is desirable we can use Ac4ft-quantifier $\Rightarrow_{q, B_1, B_2}^{F>I}$ defined by the condition $\frac{a_F}{a_F + b_F} - \frac{a_I}{a_I + b_I} \geq q \wedge a_I \geq B_1 \wedge a_F \geq B_2$.

We use examples concerning the data set STULONG described at the website <http://euromise.vse.cz/challenge2004/> 1. Data set consists of four data matrices, we deal with data matrix *Entry* only. It concerns 1 417 patients, each row describes one patient. Data matrix has 64 columns corresponding to particular attributes – characteristics of patients. In a following example we use attributes *Height* (in cm), *BMI* (Body Mass Index), *Cholesterol* (in mg%).

¹ The study (STULONG) was realized at the 2nd Department of Medicine, 1st Faculty of Medicine of Charles University and Charles University Hospital (head. Prof. M. Aschermann, MD, SDr, FESC), under the supervision of Prof. F. Boudík, MD, ScD, with collaboration of M. Tomečková, MD, PhD and Ass. Prof. J. Bultas, MD, PhD. The data were transferred to the electronic form by the EuroMISE Centre of Charles University and Academy of Sciences (head. Prof. RNDr. J. Zvárová, DrSc).

An example of the action rule is the rule \mathcal{R}_1

$$Height\langle 163, 175 \rangle \wedge [BMI(> 30) \rightarrow BMI(27; 30)] \Longrightarrow_{0.119, 11, 12}^{I>F} Cholesterol(\geq 290)$$

that is described by two association rules \mathcal{R}_{1I} and \mathcal{R}_{1F} with 4ft-quantifiers $\Rightarrow_{=, 0.204, 11}$ and $\Rightarrow_{=, 0.085, 12}$:

$$\mathcal{R}_{1I} : Height\langle 163, 175 \rangle \wedge BMI(> 30) \Rightarrow_{=, 0.204, 11} Cholesterol(\geq 290)$$

$$\mathcal{R}_{1F} : Height\langle 163, 175 \rangle \wedge BMI(24; 27) \Rightarrow_{=, 0.085, 20} Cholesterol(\geq 290) .$$

The corresponding 4ft-tables $4ft(Height \wedge BMI_I, Chlst, STULONG)$ and $4ft(Height \wedge BMI_F, Chlst, STULONG)$ are in Fig. 3. We denote $Height\langle 163, 175 \rangle$ as $Height$, $BMI(> 30)$ as BMI_I , $BMI(24; 27)$ as BMI_F , and $Cholesterol(\geq 290)$ as $Chlst$.

STULONG	<i>Chlst</i>	$\neg Chlst$	STULONG	<i>Chlst</i>	$\neg Chlst$
$Height \wedge BMI_I$	11	43	$Height \wedge BMI_F$	12	130
$\neg(Height \wedge BMI_I)$	128	1235	$\neg(Height \wedge BMI_F)$	127	1148
$4ft(Height \wedge BMI_I, Chlst, STULONG)$			$4ft(Height \wedge BMI_F, Chlst, STULONG)$		

Fig. 3. 4ft-tables for association rules \mathcal{R}_{1I} and \mathcal{R}_{1F}

$Height$ is a stable attribute and BMI is a flexible attribute that can be influenced by the patient. The Boolean attribute $Cholesterol(\geq 290)$ means that the level of cholesterol is too high. Attribute $Cholesterol$ is here considered as stable even if the level of cholesterol can be also influenced by the patient. The rule \mathcal{R}_1 describes how the probability of having $Cholesterol(\geq 290)$ can be influenced by change of BMI for patients with height in interval $\langle 163, 175 \rangle$. Its message is that among patients satisfying $Height\langle 163, 175 \rangle \wedge BMI(> 30)$ are 20,4 percent of patients satisfying $Cholesterol(\geq 290)$ but among patients satisfying $Height\langle 163, 175 \rangle \wedge BMI(24; 27)$ only 8,5 percent of patients satisfy $Cholesterol(\geq 290)$.

5 Ac4ft-Miner

The *Ac4ft-Miner* is a GUHA procedure. It means that its input consists of a relatively simple definition of a large set Ω of relevant G-action rules of the form

$$\varphi_{St} \wedge \Phi_{Chg} \approx^* \psi_{St} \wedge \Psi_{Chg}$$

and of an analyzed data matrix \mathcal{M} . The *Ac4ft-Miner* procedure generates all G-action rules $\omega \in \Omega$ and verifies each of them in the analyzed data matrix \mathcal{M} . The output of the *Ac4ft-Miner* consists of all G-action rules ω true in \mathcal{M} .

The set Ω is given by definitions of the set $\mathcal{B}_{A, St}$ of relevant Boolean attributes considered as stable antecedents, the set \mathcal{C}_A of relevant changes of antecedent,

the set $\mathcal{B}_{S,St}$ of Boolean attributes considered as relevant stable succedents, the set \mathcal{C}_S of relevant changes of succedent, and the Ac4ft-quantifier \approx^* . The rule $\varphi_{St} \wedge \Phi_{Chg} \approx^* \psi_{St} \wedge \Psi_{Chg}$ belongs to the set Ω of relevant G-action rules if it is satisfied $\varphi_{St} \in \mathcal{B}_{A,St}$, $\Phi_{Chg} \in \mathcal{C}_A$, $\psi_{St} \in \mathcal{B}_{S,St}$, $\Psi_{Chg} \in \mathcal{C}_{St}$. An example of input of the *Ac4ft-Miner* procedure is in Fig. 4

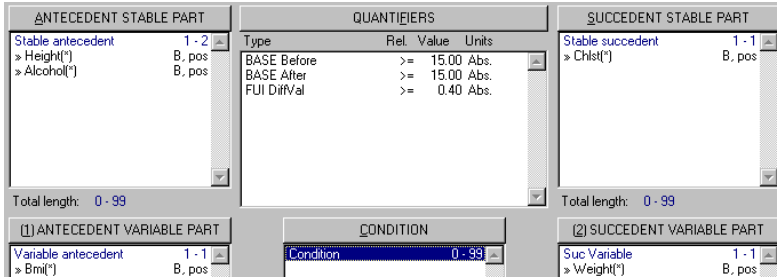


Fig. 4. Example of Input of the *Ac4ft-Miner* Procedure

The set $\mathcal{B}_{A,St}$ of relevant Boolean attributes considered as stable antecedents is defined in left upper part named ANTECEDENT STABLE PART of Fig. 4, details of the definition are in Fig. 5. The set $\mathcal{B}_{A,St}$ is defined as a conjunction of 1 - 2 of Boolean attributes, see third row in Fig. 5. The conjunction of 1 Boolean attribute is the attribute itself. The Boolean characteristics of the attributes *Height* and *Alcohol* are used. The sets $\mathcal{B}(Height)$ and $\mathcal{B}(Alcohol)$ of relevant Boolean characteristics of the attributes *Height* and *Alcohol* respectively are defined in Fig. 5. The set $\mathcal{B}_{A,St}$ consists of all Boolean attributes φ_1, φ_2 , and $\varphi_1 \wedge \varphi_2$ where $\varphi_1 \in \mathcal{B}(Height)$ and $\varphi_2 \in \mathcal{B}(Alcohol)$.

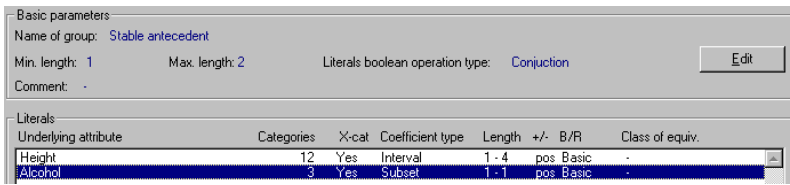


Fig. 5. Definition of the set $\mathcal{B}_{A,St}$

The attribute *Height* has 12 categories - intervals of height in cm: $\langle 148; 160 \rangle$, $\langle 160; 163 \rangle$, ..., $\langle 187; 190 \rangle$, $\langle 190; 202 \rangle$. The set $\mathcal{B}(Height)$ is defined as the set of all Boolean attributes $Height(\alpha)$ where α is an interval of 1-4 categories (i.e. 1-4 consecutive categories), see expressions "Interval 1-4" and "Coefficient type Length" in Fig. 5. This way 42 (i.e. $9+10+11+12$) particular Boolean attributes are defined. The attribute $Height(\langle 148; 160 \rangle, \langle 160; 163 \rangle)$ is an example of Boolean attribute with two consecutive categories - intervals $\langle 148; 160 \rangle$

and $\langle 160; 163 \rangle$, it is equivalent to $Height(148; 163)$. It is true for a patient if the height of this patient is in the interval $(148; 163)$. The attribute *Alcohol* concerns patient's consumption of alcohol. It has 3 categories - *no*, *occasionally*, and *regularly*. The set $\mathcal{B}(Alcohol)$ consists of three Boolean attributes $Alcohol(no)$, $Alcohol(occasionally)$, and $Alcohol(regularly)$. The definition in Fig. 5 means that the set $\mathcal{B}_{A,St}$ consists of 171 Boolean attributes - relevant stable antecedents.

The set \mathcal{C}_A of relevant changes of antecedent is given by a set $\mathcal{B}_{A,Fl}$ of relevant Boolean attributes created from flexible antecedent attributes. Remember the definition of the *change of Boolean attribute* given at the beginning of Sect. 4. The *initial state* $\mathcal{I}(A)$ of the change of the coefficient $A = [Z(\kappa) \rightarrow Z(\kappa')]$ is defined as $\mathcal{I}(A) = Z(\kappa)$ and the *final state* $\mathcal{F}(A)$ of A is defined as $\mathcal{F}(A) = Z(\kappa')$. The *change of Boolean attribute* is created from *changes of coefficient* and Boolean connectives in the same way as the Boolean attribute is created from the literals. If Φ is a change of Boolean attribute, then the *initial state* $\mathcal{I}(\Phi)$ of Φ is a Boolean attribute that we get by replacing all changes of literals A occurring in Φ by their initial states $\mathcal{I}(A)$. The *final state* $\mathcal{F}(\Phi)$ of Φ is a Boolean attribute that we get by replacing all changes of literals A occurring in Φ by their final states $\mathcal{F}(A)$. The set \mathcal{C}_A of relevant changes of antecedent consists of all changes Φ of Boolean attribute such that both $\mathcal{I}(\Phi) \in \mathcal{B}_{A,Fl}$ and $\mathcal{F}(\Phi) \in \mathcal{B}_{A,Fl}$.

Note that there are various fine tools how to define a set of relevant Boolean attributes in the procedure *Ac4ft-Miner*. They are the same as in the *4ft-Miner* procedure [10]. Their detailed description is out of the scope of this paper. A simple example is the definition of the set $\mathcal{B}_{A,St}$ above. Other simple examples are the definitions of the sets $\mathcal{B}_{A,Fl}$, $\mathcal{B}_{S,St}$, and $\mathcal{B}_{S,Fl}$ below.

The set $\mathcal{B}_{A,Fl}$ is defined in a left bottom part of Fig. 4 named ANTECEDENT VARIABLE PART. Details are not shown here due to limited space. The set $\mathcal{B}_{A,Fl}$ is defined as the set $\mathcal{B}(BMI)$ of Boolean characteristics of the attributes *BMI*, similarly to the definition of the sets $\mathcal{B}(Height)$ above. The attribute *BMI* has 13 categories - intervals $(16; 21)$, $(21; 22)$, \dots , $(31; 32)$, > 32 . The set $\mathcal{B}(BMI)$ is defined as the set of all Boolean attributes $BMI(\kappa)$ where κ is an interval of 1-3 categories. This way 36 particular Boolean attributes $BMI(\kappa)$ are defined. It means that there are 1010 relevant changes antecedent in the set \mathcal{C}_A , all of them have the form $[BMI(\kappa_1) \rightarrow BMI(\kappa_2)]$ where $\kappa_1 \cap \kappa_2 = \emptyset$.

The set $\mathcal{B}_{C,St}$ of relevant Boolean attributes considered as stable succedents is defined in right upper part named SUCCEDED STABLE PART of Fig. 4. The attribute *Chlst* (i.e. Cholesterol in mg%) is used as only one stable succedent attribute and thus the set $\mathcal{B}(Chlst)$ of relevant Boolean characteristics of the attribute *Chlst* corresponds to the set $\mathcal{B}_{C,St}$. The attribute *Chlst* has 19 categories - intervals ≤ 150 , $(150; 160)$, \dots , $(310; 320)$, > 320 . The set $\mathcal{B}(Chlst)$ is defined as the set of all Boolean attributes $Chlst(\kappa)$ where κ is an interval of 1-4 categories. This way 70 particular Boolean attributes $Chlst(\kappa)$ are defined.

The set \mathcal{C}_S of relevant changes of succedent is defined in right bottom part of Fig. 4 named SUCCEDED VARIABLE PART. The set $\mathcal{B}_{S,Fl}$ of relevant Boolean attributes is used to define \mathcal{C}_S in a same way the set $\mathcal{B}_{A,Fl}$ is used to define \mathcal{C}_A , see above. The attribute *Weight* is used as only one flexible succedent

attribute and thus the set $\mathcal{B}(Weight)$ of relevant Boolean characteristics of the attribute *Weight* corresponds to the set $\mathcal{B}_{S,Fl}$. The attribute *Weight* has 12 categories - intervals (50; 60), (60; 65), . . . , (105; 110), (110; 135). The set $\mathcal{B}(Weight)$ is defined as the set of all Boolean attributes $Weight(\kappa)$ where κ is an interval of 1-6 categories. This way 57 particular Boolean attributes $Weight(\kappa)$ are defined.

The Ac4ft-quantifier is defined in the middle upper part of Fig. 4 named QUANTIFIERS. There is written BASE Before ≥ 15.00 , BASE After ≥ 15.00 , and FUIDiffVal ≥ 0.40 , thus the Ac4ft-quantifier $\implies_{0.4,15,15}^{I>F}$ is used.

There is not known a precise formula to compute the number of relevant changes of antecedents and succedents defined this way, thus it is hard to estimate the number of all relevant G-action rules. The task defined in Fig. 4 was solved in 14 hours and 15 min at PC with 1.33 GHz and 1.99 GB RAM. More than $388 * 10^6$ of action rules $\varphi_{St} \wedge \Phi_{Chg} \implies_{0.4,15,15}^{I>F} \psi_{St} \wedge \Psi_{Chg}$ were generated and verified and 30 true rules were found. The strongest one is the rule \mathcal{R}_0 :

$$Height(175, 184) \wedge Alcohol(regularly) \wedge [BMI(21, 24) \rightarrow BMI(24, 27)] \implies_{0.4,15,15}^{I>F} \\ \implies_{0.4,15,15}^{I>F} Cholesterol(190, 230) \wedge [Weight(\leq 80) \rightarrow Weight(80, 85)] .$$

Its effect can be expressed by two association rules \mathcal{R}_{0I} and \mathcal{R}_{0F} with 4ft-quantifiers $\Rightarrow_{=,0.634,26}$ and $\Rightarrow_{=,0.202,12}$ and 4ft-tables $4ft(\varphi_{0I}, \psi_{0I}, STULONG)$ and $4ft(\varphi_{0F}, \psi_{0F}, STULONG)$ given in Fig. 6.

$$\mathcal{R}_{0I} : \quad \varphi_{0I} \Rightarrow_{=,0.634,26} \psi_{0I} \qquad \mathcal{R}_{0F} : \quad \varphi_{0F} \Rightarrow_{=,0.202,12} \psi_{0F}$$

Here $\varphi_{0I} = Height(175, 184) \wedge Alcohol(regularly) \wedge BMI(21, 24)$,

STULONG	ψ_{0I}	$\neg\psi_{0I}$	STULONG	ψ_{0I}	$\neg\psi_{0I}$
φ_{0I}	26	15	φ_{0I}	17	67
$\neg\varphi_{0I}$	242	1134	$\neg\varphi_{0I}$	69	1264
$4ft(\varphi_{0I}, \psi_{0I}, STULONG)$			$4ft(\varphi_{0F}, \psi_{0F}, STULONG)$		

Fig. 6. $4ft(\varphi_{0I}, \psi_{0I}, STULONG)$ and $4ft(\varphi_{0F}, \psi_{0F}, STULONG)$

$\psi_{0I} = Cholesterol(190, 230) \wedge Weight(\leq 80)$,
 $\varphi_{0F} = Height(175, 184) \wedge Alcohol(regularly) \wedge BMI(24, 27)$,
 and $\psi_{0F} = Cholesterol(190, 230) \wedge Weight(80, 85)$.

Note that the strongest rule is the rule with the highest difference of confidences of the rules \mathcal{R}_{0I} and \mathcal{R}_{0F} . The found 30 true rules can be grouped into four groups concerning patients with the same height. All results concern patients satisfying *Alcohol(regularly)*. More detailed interpretation of results requires deeper medical knowledge.

The above described application of the *Ac4ft-Miner* procedure can be understood as an attempt to answer the analytical question "How to decrease probability of having cholesterol level in a certain interval for patients with height in

a certain interval and with some type of alcohol consumption by changing BMI and with some induced change of weight? We can get some variants of answer by analyzing the output of the *Ac4ft-Miner*.

Note that the implementation of the *Ac4ft-Miner* procedure is based on the bit string representation of analyzed data, a-priori algorithm is not used, see Sect. 3. The used algorithm is similar to that of the *4ft-Miner* procedure, see 10. Its performance is sensitive to the parameters B_1 and B_2 of the Ac4ft-quantifier $\Rightarrow_{q, B_1, B_2}^{I>F}$. The higher are the B_1 and B_2 the more sure uninteresting G-action rules can be skipped. Let us give overview of results of four runs of the *Ac4ft-Miner* with the sets $\mathcal{B}_{A, St}$, \mathcal{C}_A , $\mathcal{B}_{S, St}$, and \mathcal{C}_S specified as above and with different Ac4ft-quantifiers: (1) $\Rightarrow_{0.4, 15, 15}^{I>F}$: 14 hours and 15 min, more than $388 * 10^6$ rules really verified and 30 true rules found (see above), (2) $\Rightarrow_{0.2, 50, 50}^{I>F}$: 2 hours and 9 min, $33.4 * 10^6$ rules really verified and 216 true rules found, (3) $\Rightarrow_{0.1, 100, 100}^{I>F}$: 16 min, 44 sec, $1, 81 * 10^6$ rules really verified and no true rule found, (4) $\Rightarrow_{0.05, 200, 200}^{I>F}$: 39 sec, 3888 rules really verified and no true rule found.

6 Conclusions

We can conclude that first experiments with the *Ac4ft-Miner* procedure and medical data show results that can be interesting from a medical point of view. The performance of the procedure is reasonable and it can be influenced by dealing with input parameters. The experiments show that the *Ac4ft-Miner* procedure deserves additional research. We suppose namely the following related research activities.

- Experiments with additional Ac4ft-quantifiers. There are various additional Ac4ft-quantifiers inspired by the 4ft-quantifiers used in the applications of the *4ft-Miner* procedure 12. An example of an interesting additional Ac4ft-quantifier is the Ac4ft-quantifier $\Rightarrow_{q, B_1, B_2}^{+, F>I}$ defined by the condition

$$\left(\frac{a_I}{a_I + b_I} - \frac{a_I + c_I}{a_I + b_I + c_I + d_I}\right) - \left(\frac{a_F}{a_F + b_F} - \frac{a_I + c_I}{a_I + b_I + c_I + d_I}\right) \geq q \wedge a_I \geq B_1 \wedge a_F \geq B_2$$

see 4ft-tables in Fig. 2. We assume $0 < q \leq 1$, $B_1 > 0$, and $B_2 > 0$. This Ac4ft-quantifier is inspired by the 4ft-quantifier $\sim_{p, Base}^+$ of *above average dependence* that is for $0 < p$ and $Base > 0$ defined by the condition $\frac{a}{a+b} \geq (1+p) \frac{a+c}{a+b+c+d} \wedge a \geq Base$, see Tab. 1. The rule $\varphi \sim_{p, Base}^+ \psi$ means that among objects satisfying φ is at least $100p$ percent more objects satisfying ψ than among all objects and that there are at least $Base$ objects satisfying both φ and ψ 9.

- Detailed study of relation of G-action rules and of the *Ac4ft-Miner* procedure to the approaches described in 5,6,7 that are only shortly mentioned in Sect. 2.
- Study of logic of G-action rules, namely study of correct deduction rules of the form $\frac{\varphi_{St} \wedge \Phi_{Chg} \approx^* \psi_{St} \wedge \Psi_{Chg}}{\varphi'_{St} \wedge \Phi'_{Chg} \approx^* \psi'_{St} \wedge \Psi'_{Chg}}$. We suppose to get results similar to results

on deduction rules of the form $\frac{\varphi \approx \psi}{\varphi' \approx \psi'}$ [9] concerning association rules $\varphi \approx \psi$ introduced in Sect. 3. Such deduction rules can be used e.g. to optimize the *Ac4ft-Miner* procedure in a similar way the rules $\frac{\varphi \approx \psi}{\varphi' \approx \psi'}$ [9] are used.

- Possibilities of application of formalized background knowledge stored in the LISP-Miner system. There are possibilities to use such knowledge e.g. to formulate reasonable analytical question and to arrange the output analytical reports [9].
- Research into parallelization of GUHA procedures and using PC-Grid to solve very large tasks.

References

1. Aggraval, R., et al.: Fast Discovery of Association Rules. In: Fayyad, U.M., et al. (eds.) *Advances in Knowledge Discovery and Data Mining*. AAAI Press, Menlo Park (1996)
2. Hájek, P. (guest editor): *International Journal of Man-Machine Studies*. special issue on GUHA 10 (1978)
3. Hájek, P., Havránek, T.: *Mechanizing Hypothesis Formation (Mathematical Foundations for a General Theory)*. Springer, Heidelberg (1978)
4. Ras, Z., Wieczorkowska, A.: Action-Rules: How to Increase Profit of a Company. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) *PKDD 2000. LNCS (LNAI)*, vol. 1910, pp. 587–592. Springer, Heidelberg (2000)
5. Ras, Z., Tsay, L.: Discovering the Concise Set of Actionable Patterns. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) *Foundations of Intelligent Systems. LNCS (LNAI)*, vol. 4994, pp. 169–178. Springer, Heidelberg (2008)
6. Seunghyun, I., Ras, Z.: Action Rule Extraction from a Decision Table: ARED. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) *Foundations of Intelligent Systems. LNCS (LNAI)*, vol. 4994, pp. 160–168. Springer, Heidelberg (2008)
7. E-Action Rules, System DEAR. In: Lin, T.Y., et al. (eds.) *Data Mining: Foundations and Practice. Studies in Computational Intelligence*, vol. 118, pp. 289–298. Springer, Heidelberg
8. Rauch, J.: Some Remarks on Computer Realisations of GUHA Procedures. *International Journal of Man-Machine Studies* 10, 23–28 (1978)
9. Rauch, J.: Logic of Association Rules. *Applied Intelligence* 22, 9–28 (2005)
10. Rauch, J.: An Alternative Approach to Mining Association Rules. In: Lin, T.Y., et al. (eds.) *Data Mining: Foundations, Methods, and Applications*, pp. 219–238. Springer, Heidelberg (2005)
11. Rauch, J., Šimunek, M.: GUHA Method and Granular Computing. In: Hu, X., et al. (eds.) *Proceedings of IEEE conference Granular Computing* (2005)
12. Rauch, J., Tomečková, M.: System of Analytical Questions and Reports on Mining in Health Data – a Case Study. In: Roth, J., et al. (eds.) *Proceedings of IADIS European Conference Data Mining 2007*, pp. 176–181. IADIS Press (2007)
13. Šimunek, M.: Academic KDD Project LISP-Miner. In: Abraham, A., et al. (eds.) *Advances in Soft Computing - Intelligent Systems Design and Applications*. Springer, Heidelberg (2003)

Semantic Analytical Reports: A Framework for Post-processing Data Mining Results

Tomáš Kliegr¹, Martin Ralbovský¹, Vojtěch Svátek¹, Milan Šimůnek¹,
Vojtěch Jirkovský², Jan Nemrava¹, and Jan Zemánek¹

¹ University of Economics, Prague, Faculty of Informatics and Statistics,
Nám. Winstona Churchilla 4, 130 67 Praha 3, Czech Republic
{tomas.kliegr,ralbovsm,svatek,simunek,nemrava,xzemj22}@vse.cz

² Czech Technical University, Dept. of Computer Science and Engineering,
Karlovo nám. 13, 121 35 Praha 2, Czech Republic
jirkov1@fel.cvut.cz

Abstract. Intelligent post-processing of data mining results can provide valuable knowledge. In this paper we present the first systematic solution to post-processing that is based on semantic web technologies. The framework input is constituted by PMML and description of background knowledge. Using the Topic Maps formalism, a generic Data Mining ontology and Association Rule Mining ontology were designed. Through combination of a content management system and a semantic knowledge base, the analyst can enter new pieces of information or interlink existing ones. The information is accessible either via semi-automatically authored textual analytical reports or via semantic querying. A prototype implementation of the framework for generalized association rules is demonstrated on the PKDD'99 Financial Data Set.

1 Introduction

Analytical report is a free-text document describing various elements of the data mining task: particularly the data, preprocessing steps, task setting and results. The analyst can also include additional information such as background knowledge, explanation of preprocessing steps and interpretation of the results. Creating analytical reports manually is time-consuming and the output document is not machine-readable, which hinders the possibilities for post-processing – e.g. querying, merging or filtering.

We present a novel framework for semi-automatic generation and processing of analytical reports that addresses these issues through the utilization of semantic web technologies. The framework is developed as part of the SEWEBAR (Semantic Web and Analytical Reports) initiative [1]. The framework is generic and should be suitable for most Data Mining (DM) algorithms. However, a specific implementation of the framework needs to take into account the knowledge representation used by the selected algorithm.

We also present a prototype implementation of the framework for association rules (ARs). As part of the prototype, a data mining ontology for generalized association rules is introduced. To demonstrate the feasibility of the approach, the prototype implementation is used to post-process the output of Ferda [6] association rule mining system on the PKDD'99 Financial Dataset [14].

The rest of the paper is organized as follows. In Section 2 we give an overview of the architecture of the framework and in Section 3 an overview of our prototype implementation. A case study introduced in Section 4 is used to demonstrate the benefits of the framework. The related work is placed towards the end of the paper into Section 5. Section 6 contains conclusions and a plan for future work.

2 Framework Outline

In this section we present an outline of a new framework for post-processing the results of data mining tasks. The framework is based on established standards and seamlessly integrates with existing data mining software as its input is constituted by PMML¹ (Predictive Model Markup Language), which is a widely adopted XML-based standard for definition and sharing of data mining and statistical models. The second part of the framework's input is optional and is constituted by the emerging Background Knowledge Exchange Format (BKEF) specification. While PMML is produced by the DM software, BKEF is created directly based on human input.

PMML and BKEF specifications are stored in a Content Management System (CMS), which allows to merge information contained in one or more specifications with human input by enabling the analyst to include visualizations of BKEF and PMML fragments into the analytical report.

Further in the work flow, PMML and BKEF specifications are transformed into a semantic representation conforming to the Data Mining Ontology and stored in a Knowledge Base (KB), which allows the analysts to append and interlink information in a structured way. KB can be searched using a semantic query language.

An overview of the framework is depicted in Figure 1.

2.1 Input Data Formats

The framework's input is constituted by the description of the data mining models and the description of background knowledge.

For model description the framework uses PMML 3.2, the latest version of the standard at the time of writing. PMML 3.2 has the following components:

- Data Dictionary: database fields that are input for the model
- Mining Schema: database fields used in the model

¹ <http://www.dmg.org/pmml-v3-2.html>

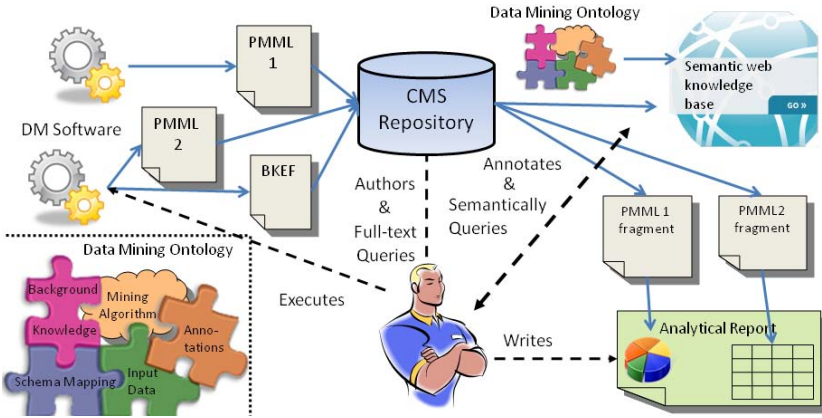


Fig. 1. Framework outline

- Transformation Dictionary: derived fields created through discretization or value mapping of database fields
- Model Description: settings and output specific to the mining algorithm

PMML 3.2 classifies DM algorithms according to knowledge representation into eleven types of Model Descriptions such as Association Rules, Cluster Models and Neural Networks. A framework implementation should support at least one Mining Description.

PMML allows two classes of applications – PMML producers and PMML consumers – to share a model. Export to PMML is supported by all major data mining software implementations acting as PMML producers. There are two major types of PMML consumers – scoring applications and archival (visualization, reference) applications [2]. The framework acts as a PMML consumer that performs the archival function but also reasons over the PMML, effectively establishing a new PMML consumer type.

To the best of our knowledge, there is no established standard analogical to PMML for background knowledge in data mining. Since the availability of such specification is vital, we are working on the BKEF specification. Its purpose is to foster exchange of information between background knowledge producers, such as dedicated user interfaces for acquisition of background knowledge e.g. [10][15], and background knowledge consumers like a CMS or a KB.

2.2 Analytical Report Authoring Support

The next element of the framework is a Content Management System (CMS), an application supporting storage and retrieval of electronic documents. The CMS is intended for storage and authoring of data mining analytical reports and related information. Analytical reports stored in a CMS are free-text documents whose purpose is to interpret the results of data mining tasks in the light of background information, other analytical reports and the expert’s knowledge.

From a large part analytical reports hence consist of visualizations of information contained in the PMML (such as data histograms, description of data preprocessing, or model parameters). This allows the analyst to reuse the visualization of the needed bits of PMML in the free-text report so that existing knowledge does not have to be reentered. Additionally, since this generated content preserves the link to its source fragment, the corresponding part of the free-text report can still be resolved to a machine-readable XML representation.

The CMS also handles background knowledge, an example of which are value ranges. Background knowledge can either be created by a dedicated software and exported to the CMS or preferably authored directly in the CMS. In either case, the background knowledge is stored in the BKEF machine-readable format and included into the free-text report in the same way as PMML.

The fact that statements in analytical reports are directly backed by the source data (PMML or BKEF fragments) not only allows to search the reports as structured data but also fosters the credibility of the reports.

2.3 ‘Semantization’ of Analytical Reports in Topic Maps

The CMS described in the previous subsection allows to query the content with full-text search (free-text reports) or XML query languages (PMML, BKEF). However, the structured content can be further ‘semantized’ by being stored into a KB according to some ontology. This is beneficial when merging heterogeneous data, such as PMML specifications of tasks executed on similar but not same datasets, or when working with background knowledge.

For the interchange and storage of semantic information, the framework relies on semantic web technologies in a broader sense. The resources for knowledge representation designed within the prototype framework implementation follow an ISO/IEC 13250 standard Topic Maps. Topic Maps are in principle interoperable with the semantic web formats RDF/OWL standardized by W3C. We have opted for Topic Maps, since they are simple and document orientated.

Topic Maps represent information through *topics*, *associations* and *occurrences*. A topic is any entity about which a human being can lead a discourse, an association represents a relationship between topics, and an occurrence represents a piece of information relevant to a topic. Types of topics, associations and occurrences used in a topic map constitute its ontology.

The framework defines: i) an ontology that allows to represent pieces of the structured XML content as instances of ontology types, ii) a transformation from structured content to instances.

After the analytical report has been semantized, it can be annotated and interlinked with reports already present in the repository. For example, in an association rule mining task, the analyst can annotate a discovered rule with the degree of its novelty, link it with an already existing rule coming either from a different report or from background knowledge. The prominent desired functionality is the search and reasoning over the resulting KB.

2.4 Data Mining Ontology

The Data Mining Ontology was derived from PMML 3.2 so that all PMML core features² can be automatically mapped into the ontology. The ontology consists of the following components: Input Data (including data transformations), Background Knowledge, Schema Mapping, Mining Algorithm and Annotations.

The *Input Data* component comes out of the corresponding components of the PMML standard: Data Dictionary, Mining Schema and Transformation Dictionary. Elements prescribed by the PMML Schema such as `DataField` were mapped to topic types; enumerations were represented as topic types with their instances representing the enumeration members. Knowledge represented as XML attributes in the PMML Schema, such as interval margins in a discretization, were represented as occurrence types.

The *Background Knowledge* component allows to relate various pieces of background knowledge to a specific data matrix through meta-attributes [9]. Meta-attribute is a generalization of an attribute³. The existence of meta-attributes stems from the fact that the same property can be coded in two datasets differently. For example, there can be an attribute `loan` with possible values A - F in one dataset, and an attribute `status` with possible values `bad`, `medium`, `good` in another; both attributes referring to loan quality. In the ontology, the meta-attribute provides a common name for the same property (here `loan quality`). A meta-attribute can have several *formats*. In our example, the formats can be named e.g. *Loan Quality [AF-Scale]*, *Loan Quality [Word Scale]*. An attribute in a dataset is a *realization* of a specific format of a meta-attribute.

The ontology supports the pieces of background knowledge introduced in [10]: i) basic value limits, ii) typical interval lengths for discretization, iii) groups of meta-attributes and iv) mutual influence among meta-attributes. These pieces of background knowledge are tied to meta-attributes through formats.

The *Schema Mapping* component allows to align an attribute or derived attribute used in one data mining task with its counterparts in other tasks. This is done through mapping the corresponding meta-attribute to its realizations. The current ontology version allows to express the mapping only in terms of *equivalence* of data values or categories.

The *Mining Algorithm* component is left for further work as it is out of the scope of this work to semantize the eleven Mining Descriptions defined in PMML 3.2. A reference Mining Algorithm component for association rules is, however, introduced as part of the framework's prototype implementation in Section 3.1.

The *Annotations* are the last component of the Data Mining Ontology. Annotations are pieces of knowledge that can be assigned by the analyst to some important concepts in the ontology. For example, an annotation can be assigned to an instance of `DiscretizeBin` to explain the reason behind the discretization.

Not considering the undefined Mining Algorithm component, basically all core PMML features are incorporated into the ontology. The current version of the ontology does not, however, support some PMML 3.2 features such as

² As core we consider features required by the PMML 3.2 XML Schema.

³ Also referred to as *data field* or *database column*.

model composition. We are not, however, aware of any issue that would prevent extending the ontology so that it encompasses the remaining features.

3 Framework Prototype

This section describes a reference implementation of the framework for association rules, which acts as a PMML consumer for two academic data mining programs Ferda and LISp-Miner⁴. Both use GUHA method to generate rules.

3.1 GUHA Method and GUHA-Based AR Ontology

GUHA method is one of the first methods of exploratory data analysis, developed in the mid-sixties in Prague. It is a general framework for retrieving interesting knowledge from data. The method has firm theoretical foundations based on observational logical calculi and statistics [5]. GUHA is realized by GUHA procedures, such as 4FT procedure for mining association rules. GUHA association rules extend mainstream association rules (as defined in [8]) in two ways:

- Boolean attributes are allowed for antecedent and consequent. *Boolean attributes* are recursive structures that enable conjunctions, disjunctions and negations of combinations of individual items. Details can be found in [7].
- A more general kind of dependency between antecedent and consequent than confidence and support or a specific interest measure is allowed. We call these dependencies *4ft-quantifiers*. The generalized association rule can be written in form $\varphi \approx \psi$, where φ and ψ are *Boolean attributes* and \approx is a *4ft-quantifier*.

It has been shown in [8] that GUHA association rules are a generalization of mainstream association rules. Hence an association rule mining ontology based on GUHA allows to express the setting and discovered rules not only for the GUHA algorithm but also for other AR mining algorithms such as those generated by the popular *apriori* algorithm.

Utilizing these facts we have proposed the Association Rule Mining Ontology based on GUHA (GUHA AR Ontology). GUHA AR Ontology is interoperable with the core features of the Model Description specification of ARs in PMML in addition to supporting the GUHA-specific extensions. This ontology is designed so that it can be used in place of the Mining Algorithm component of the Data Mining Ontology.

3.2 Framework Implementation

This section gives an overview of the steps necessary to implement the framework on the example of our GUHA-based prototype.

In order to meet the framework's input data requirement, we used the PMML Extension mechanism [2] to incorporate GUHA features into the PMML Association Model thus creating a GUHA AR PMML model. Both our DM tools had

⁴ <http://lispmminer.vse.cz>, <http://ferda.sourceforge.net>

to be made conformable with the adapted model. Based on our prior experience with background knowledge [9,10,15], the first version of BKEF was drafted.

For the analytical report authoring support we used the PHP-based open source Joomla! (<http://www.joomla.org>) CMS, one of the most popular open source CMS systems as of time of writing. Joomla's advantages include object-oriented architecture, thousands of available extensions and active developer community. We extended Joomla! with an XSLT transformation⁵ plug-in and defined transformations from PMML to HTML and from BKEF to HTML. This visualization is used by another new Joomla! extension, which allows the analyst to include visualized PMML fragments into the analytical report.

The semantization of analytical reports is based on the GUHA AR Ontology introduced in Subsection 3.1. Ontopia Knowledge Suite⁶ (OKS) was used as the topic map repository and knowledge base. OKS can be either interactively browsed through using the Omnigator application or queried using *tolog*, a query language based on Prolog and SQL.

4 Case Study: Financial Dataset

The goal of this case study is to evaluate the usability of the prototype implementation in operation and to demonstrate the potential of our approach. In the experiment we went through the process of designing a data mining task, executing it, generating a PMML model, uploading it to the Joomla! CMS, inputting background knowledge, authoring an analytical report, semantizing the information via the OKS knowledge base and, finally, executing sample tolog queries. We used the Financial Data Set introduced in PKDD'99 [14].

4.1 Data Mining Task

The Financial Dataset consists of 8 tables describing the operations of bank customers. Among the 6181 customers we aim to identify subgroups with high occurrence of bad loans. For the mining schema we used the columns `duration`, `status` and `district`; all columns come from the `Loans` table.

Since there is no background knowledge specified in the PKDD'09 task setting, the following was introduced for case study purposes:

- *Background Knowledge 1 (BK1)* The quality of loans in Central Bohemia, the richest region of the country, is generally good. In other regions, it is in average lower. If area is expressed in terms of cities rather than regions then the value `Central Bohemia` maps to the value `Prague`.
- *Background Knowledge 2 (BK2)* If loan quality is expressed in terms of values A - D then the value A maps to `good`, B to `medium` and C,D to `bad`.
- *Background Knowledge 3 (BK3)* If loan duration is expressed in months then three bins should be created `< 0; 12 >`, `< 13; 23 >` and `< 24; inf >`.

⁵ An XML technology for translating between different knowledge representations [1].

⁶ <http://www.ontopia.net>

All these pieces of background knowledge can be input in a structured way directly into the data mining systems: [10] introduces the mechanisms needed for BK1 and [15] for BK2 and BK3. For example, using the convention introduced in [9], BK1 can be input into LISp-Miner’s KB module:

$$Region(CentralBohemia) \rightarrow^+ LoanQuality(good). \quad (1)$$

Using BK2 and BK3, the data were preprocessed in the following way: the `duration` column was discretized into `1 year`, `13-23 months` and `two years+` categories. A `statusAggregated` derived field was created from the `status` column by mapping the status values A to `Good`, B to `Medium` and C, D to the category `Bad`. The derived field `district` was created by 1 : 1 mapping from the `district` column, which has a granularity of municipality.

In Ferda, we formulated the following task:

$$duration(SS[1 - 1])\&district(Praha) \Rightarrow statusAggregated(I[2 - 2])$$

Here, $duration(SS[1 - 1])$ means that all subsets of the attribute of minimal and maximal length equal to 1 are created. Derived boolean attribute setting $statusAggregated(I[2 - 2])$ means that intervals of `status` (viewed as cardinal domain) of maximal and minimal length 2 are created. The *4ft-quantifier* used was *above average dependence* with parameters $p = 0.1$ and $minSup = 0.1$. This quantifier can be verbally interpreted as *Among object satisfying antecedent, there are at least 100p per cent more objects satisfying consequent than among all observed objects and there are at least minSup percent observed objects satisfying both antecedent and consequent.*

AR 1: $duration(13 - 23months) \Rightarrow statusAggregated(medium, bad)$

AR 2: $duration(1year) \Rightarrow statusAggregated(good, medium)$

AR 3: $duration(2y+)\&district(Praque) \Rightarrow statusAggregated(good, medium)$

The three rules listed above are the strongest among the 7 found.

4.2 Handling Data Mining Knowledge in the Prototype Framework

The knowledge relating to the DM task was then input to the CMS system. The mining model was input automatically via PMML while the background knowledge was input manually, since automated import for BKEF is not yet available. Using tools offered by the CMS, the analyst authored the report; the analyst’s productivity increased through the possibility to reuse the HTML visualization of the structured content that was generated by the XSLT transformation.

In the report the analyst stated that AR 1 is not interesting – despite its strength concerning quantifier values. Rule AR 2 was, in turn, found surprising and useful. The analyst did not comment on Rule AR 3. The result of the analyst’s work is a human-readable report, which is interlinked with its source data – primarily the PMML. However, since the report is written in free text, the information added by the analyst, such as information pertaining to the novelty of the association rules, is not machine-readable.

This is solved by the next step in the framework. The structured content is ‘semantized’ through the conversion to the Data Mining Ontology. The semantization can be done e.g. with an XSLT transformation, however, in our experiment this was done by manually reentering the data to OKS.

The GUHA AR Ontology introduced in Subsection 3.1 was used in place of the Mining Algorithm component of the ontology. The attributes referred to by background knowledge were expressed in terms of meta attributes *Area*, *Loan Quality* and *Loan Duration* and their formats *Area [Region]*, *Loan Quality [Alphabetical]* and *Loan Duration [Months]*. Formats were mapped to realizations using the schema mapping component of the ontology.

The resulting topic map was stored in a knowledge base created in OKS. The knowledge base allows the analyst to input new pieces of knowledge: machine-readable annotations. In this way, AR1 is annotated as ‘Not Interesting’ while AR2 is annotated as ‘Surprising and Useful’.

4.3 The Added Value of Semantization

The knowledge base allows for sophisticated search with the tolog language. The analyst can choose which ‘vocabulary’ to use in the query. The analyst can e.g. decide between querying in terms of background knowledge and/or in terms of a specific dataset. Two example queries are listed below.

In the first query, the analyst wants to find all discovered rules that are annotated as surprising and have a good loan quality in the consequent. The primary ontology elements exploited by the corresponding tolog query are schema mapping and meta-attributes. The rule found is (as expected)

AR 2: *duration(1year) ⇒ statusAggregated(good, medium)*.

The analyst’s second query uses the tolog’s inference engine to find discovered rules whose antecedent subsumes the background knowledge rule BK 1: *Region(CentralBohemia) →⁺ LoanQuality(good)*. The result of the query is AR3. Although the analyst had initially failed to notice that AR 3 corresponds to BK 1, the system was able to infer this automatically.

5 Related Work

The current work follows up on initial attempts for data mining analytic reporting published in [11]. However, this early work did not explicitly consider ontologies or even mark-up languages such as PMML. We are not aware of any other initiative for sharing data mining results, from any domain, over the semantic web. The impact of semantic web technology on data mining has typically been perceived as shift to ‘knowledge-intensive’ mining, in which ontologies serve as prior knowledge [3,4,16] or as a means to (locally) interpret the results [12]. We also cover this aspect to some degree, although we put more stress on the data integration and inferencing (querying) aspects.

Ontologies have already been suggested as formal models of the data mining field. The applications were in data mining work flows [3,16], grid-based data

mining [4] or parameterizing a specific mining tool [13]. Most importantly, these DM ontologies were only applied in the phases of the DM process before or during the actual mining disregarding the problem of post-processing.

PMML has been used by various subjects in industry and research. However, search and aggregation applications over PMML documents have not been significantly reported. Use of this powerful mark-up language has typically been restricted to model exchange among mining tools. Combination of PMML with truly semantic resources has not been mentioned so far.

The setup of the prototype framework was complicated by the lack of open source semantic-aware CMS systems that would be in production-ready stage of development. This fact caused the separation of the framework into two independent systems – the Joomla! CMS and the OKS knowledge base.

To the best of our knowledge, there are no such systems based-upon the Topic Map standard existing or under active development. However, what concerns the W3C RDF/OWL standards, there is some support emerging. For example, the community around Drupal CMS is working on RDF and SPARQL support [7]. For this reason, we are considering a reimplementaion of our framework to technologies conforming to the W3C standards.

6 Conclusions and Future Work

This paper introduces, to the best of our knowledge, the first systematic solution to post-processing data mining results that exploits semantic web technologies. The framework is built upon proven standards and technologies such as XML technologies and content management systems. The proposed data mining ontology is designed with respect to the industry standard PMML format, which should foster adoption of the framework among data mining practitioners.

Using the PKDD'99 Financial Dataset, we have shown how the framework can ease routine tasks such as authoring an analytical report. Its main strength lies, however, in the possibility to benefit from the querying and data integration capabilities given by the use of semantic web technologies. Using a topic-map-driven knowledge base, we have executed two example queries that used background knowledge, semantic annotation and schema mapping.

Since the input of the framework is constituted by PMML, the prototype implementation can be easily adapted to consume results from other DM tools such as Weka or SPSS [8]. The ontology, Joomla! extensions, BKEF specification, and other resources are available online [9].

The future work should focus on the completion of the Data Mining Ontology and of the BKEF specification. There is a work-in-progress on a new Joomla! extension for elicitation of background knowledge from the experts. We are also

⁷ <http://drupal.org/project/sparql>

⁸ XSLT transformations need to be customized to fit the required PMML Mining Model and the possible DM tool's extensions to PMML.

⁹ <http://keg.vse.cz/sewobar/>

considering reimplementing of the knowledge base part of our framework to technologies conforming to the W3C RDF/OWL standards.

Acknowledgement. The work described here has been supported by Grant No. 201/08/0802 of the Czech Science Foundation, by Grant No. ME913 of Ministry of Education, Youth and Sports, of the Czech Republic, and by Grant No. IGA 21/08 of the University of Economics, Prague.

References

1. W3C: XSL Transformation (1999), <http://www.w3.org/TR/xslt>
2. DMG: PMML 3.2 Specification, <http://www.dmg.org/pmml-v3-2.html>
3. Bernstein, A., Provost, F., Hill, S.: Toward intelligent assistance for a data mining process: An ontology-based approach for cost-sensitive classification. *IEEE Trans. on Knowl. and Data Eng.* 17(4), 503–518 (2005)
4. Cannataro, M., Comito, C.: A data mining ontology for grid programming. In: *Proceedings of the 1st Int. Workshop on Semantics in Peer-to-Peer and Grid Computing (SemPGrid 2003)*, pp. 113–134 (2003)
5. Hájek, P., Havránek, T.: *Mechanizing Hypothesis Formation (Mathematical Foundations for a General Theory)*. Springer, Heidelberg (1978)
6. Kováč, M., Kuchař, T., Kuzmin, A., Ralbovský, M.: Ferda, New Visual Environment for Data Mining. In: *Znalosti 2006, Czech Rep.*, pp. 118–129 (2006) (in Czech)
7. Ralbovský, M., Kuchař, T.: Using Disjunctions in Association Mining. In: Perner, P. (ed.) *ICDM 2007. LNCS (LNAI)*, vol. 4597, pp. 339–351. Springer, Heidelberg (2007)
8. Rauch, J.: Logic of Association Rules. *Applied Intelligence* 22, 9–28 (2005)
9. Rauch, J., Šimůnek, M.: Dealing with Background Knowledge in the SEWEBAR Project. In: *ECML/PKDD Workshop: Prior Conceptual Knowledge in Machine Learning and Data Mining, Warsaw 2007*, pp. 97–108 (2007)
10. Rauch, J., Šimůnek, M.: LAREDAM Considerations on System of Local Analytical Reports from Data Mining. Toronto 20.05.2008 – 23.05.2008. In: *Foundations of Intelligent Systems*, pp. 143–149. Springer, Berlin (2008)
11. Rauch, J., Šimůnek, M.: Semantic Web Presentation of Analytical Reports from Data Mining – Preliminary Considerations. In: *Web Intelligence 2007*, pp. 3–7. IEEE Computer Society, Los Alamitos (2007)
12. Svátek, V., Rauch, J., Ralbovský, M.: Ontology-Enhanced Association Mining. In: Ackermann, M., Berendt, B., Grobelnik, M., Hotho, A., Mladenič, D., Semeraro, G., Spiliopoulou, M., Stumme, G., Svátek, V., van Someren, M., et al. (eds.) *EWMF 2005 and KDO 2005. LNCS (LNAI)*, vol. 4289, pp. 163–179. Springer, Heidelberg (2006)
13. Suyama, A., Yamaguchi, T.: Specifying and Learning Inductive Learning Systems using Ontologies. In: *AAAI 1998 Workshop on the Methodology of Applying Machine Learning*, pp. 29–36 (1998)
14. *Workshop Notes on Discovery Challenge (workshop at PKDD 1999)*, Prague, Czech Rep. (August 1999)
15. Zeman, M., Ralbovský, M., Svátek, V., Rauch, J.: Ontology-Driven Data Preparation for Association Mining. In: *Znalosti 2009, Czech Republic*, pp. 270–283 (2009)
16. Žáková, M., Křemen, P., Železný, F., Lavrač, N.: Planning to Learn with a Knowledge Discovery Ontology. In: *Planning to Learn Workshop at ICML 2008 [CD-ROM]*, pp. 29–34. Omnipress, Madison (2008)

Medical Decision Making through Fuzzy Computational Intelligent Approaches

Elpiniki I. Papageorgiou

Department of Informatics and Computer Technology, Technological Educational Institute (TEI) of Lamia, 3rd km Old National Road Lamia-Athens, 35100 Lamia, Greece
epapageorgiou@teilam.gr

Abstract. A new approach for the construction of Fuzzy Cognitive Maps augmented by knowledge through fuzzy rule-extraction methods for medical decision making is investigated. This new approach develops an augmented Fuzzy Cognitive Mapping based Decision Support System combining knowledge from experts and knowledge from data in the form of fuzzy rules generated from rule-based knowledge discovery methods. Fuzzy Cognitive Mapping (FCM) is a fuzzy modeling methodology based on exploiting knowledge and experience from experts. The FCM accompanied with knowledge extraction and computational intelligent techniques, contribute to the development of a decision support system in medical informatics. The proposed approach is implemented in a well-known medical problem for assessment of treatment planning decision process in radiotherapy.

1 Introduction

This paper investigates a fuzzy computational intelligent framework to handle different data types for decision support tasks in medical informatics. More specifically, it reports a methodology to construct an advanced framework implementing a fuzzy knowledge extraction method for the design of Fuzzy Cognitive Mapping decision support system in medicine.

Fuzzy Cognitive Map (FCM) is a soft computing technique used for causal knowledge acquisition and supporting causal knowledge reasoning process. FCM permits the necessary cycles for knowledge expression within their feedback framework of systems. FCMs are useful methods for exploring and evaluating the impact of inputs on dynamical systems that involve a set of objects such as processes, policies, events and values as well as the causal relationships between those objects.

Generally, a decision-making procedure is a complex process that has to take under consideration a variety of interrelated functions. In Medical Decision Support Systems (MDSS) we are not only interested on the accuracy and prediction of the results (as in classification and data mining techniques) but for the transparency and interpretability of the results from the medical practitioner who uses the MDSS in his daily clinical practice [1].

The *a priori* knowledge about a problem to be solved is frequently given in a symbolic, rule-based form. Extraction of knowledge from data, combining it with available symbolic knowledge, and refining the resulting knowledge-based expert systems is a great challenge for computational intelligence. Reasoning with logical rules is

more acceptable to human users than the recommendations given by black box systems [2], because such reasoning is comprehensible, provides explanations, and may be validated by human inspection. It also increases confidence in the system, and may help to discover important relationships and combination of features, if the expressive power of rules is sufficient for that.

In this study, FCM is developed combining knowledge from experts and from data, using rule extraction methods that generate meaningful fuzzy rules. The performance of FCMs is known to be sensitive to the initial weight setting and architecture. This shortcoming can be alleviated and the FCM model can be enhanced if a fuzzy rule base (IF-THEN rules) is available. A number of knowledge extraction techniques such as, fuzzy systems, neurofuzzy, machine learning and other computational intelligence techniques used for the generation of fuzzy rule base [3,4]. These methods extract the available knowledge from data in the form of fuzzy rules and insert them into the FCM based decision support system.

Few frameworks based on fuzzy cognitive maps for the task of reasoning and learning in medical decision systems have been proposed [5-8]. This paper investigates a new approach to address the complex problem of medical decision making that applied to the radiation therapy treatment planning problem for making decisions on ionizing radiation.

2 Fuzzy Cognitive Map Theory

Fuzzy cognitive maps (FCMs) are simple, yet powerful tool for modeling and simulation of dynamic systems. They were originally introduced by Kosko [9] as an extension of cognitive maps. The main advantage of FCMs lies in their straightforward graph representation, which consists of nodes connected by edges. Nodes correspond to concepts or variables within given domain of application, whereas directed edges reflect mutual relationship between concepts. Each edge is associated with a weight value from the range $[-1, 1]$ that expresses both the type and strength of given relationship. Negative value indicates prohibitory effect that source concept exerts on the destination one, Positive value indicates a promoting effect. The zero value denotes no causal relationship between two concepts. The graph representation can be equivalently denoted by a square matrix, called *connection matrix*. It accumulates all weight values for edges between corresponding concepts. Figure 1 shows an example of FCM model that concerns public city health issues [10].

During simulation, FCM iteratively calculates its state that is represented by a *state* vector \mathbf{A} , which consists of all nodes values (A_i) at given iteration. Value of each node is determined based on values of nodes that exert influence on the given node, i.e. nodes that are connected to this node. These values are multiplied by corresponding connection matrix E_{ij} and the sum of these products is taken as the input to a transformation function f . So, the value A_i of each concept C_i is calculated by the following rule:

$$A_i^{(k+1)} = f(A_i^{(k)} + \sum_{\substack{j \neq i \\ j=1}}^N A_j^{(k)} \cdot e_{ji}) \quad (1)$$

The purpose of using the function f is to normalize the node value, usually to the range $[0,1]$. As a result, each node can be defined as active (value of 1), inactive (value of 0), or active to a certain degree (value between 0 and 1) [11].

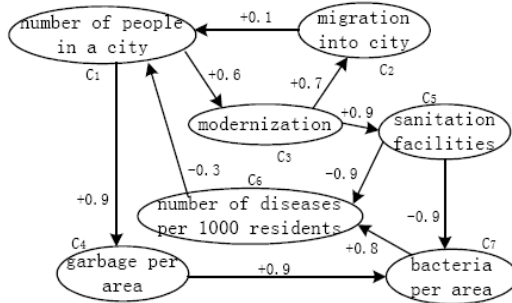


Fig. 1. An example of FCM model for public city health

Compared with other schemes for developing knowledge bases, such as the rule base in an expert system, the process of constructing an FCM might seem relatively simple. FCMs can be produced by expert manually or generate by other source of information computationally. Experts develop a mental model (graph) manually based on their knowledge in related area. At first, they identify key domain issues or concepts. Secondly, they identify the causal relationships among these concepts and thirdly, they estimate causal relationships strengths. This achieved graph shows not only the components and their relations but also the strengths. In fuzzy diagrams, the influence of a concept on the others is considered as “negative”, “positive” or “neutral”. All relations are expressed in fuzzy terms, e.g. very weak, weak, medium, strong and very strong. Then, the proposed linguistic variables suggested by experts, are aggregated using the SUM method and an overall linguistic weight is produced, which with the defuzzification method of Center of Gravity, is transformed to numerical weight e_{ji} . A detailed description on the FCM development is given in [8].

Although developing an FCM manually might seem simple, it is in fact difficult because the knowledge from experts is not enough and there is difficulty to handle knowledge from different sources of information. Therefore, a systematic way should be found in order to bridge this gap. For example, designing a new method using data mining and knowledge extraction approaches from data could eliminate the existing weakness and enhance the FCM structure.

3 Generation of Fuzzy Rules Using Knowledge Based Extraction Methods

The huge amount of medical data and the different sources of medical information make the task of decision making difficult and complex. Data mining and knowledge processing systems are intelligent systems that used in medicine for the tasks of diagnosis, prognosis, treatment planning and decision support [12,13].

In the medical field, it is preferable not to use black box approaches. The user should be able to understand the modeler and to evaluate its results. Among the wide range of possible approaches, the fuzzy decision tree based rule generation computing method was selected to extract the knowledge and construct a compact and useful fuzzy rule base.

3.1 Extraction Method Using Fuzzy Decision Trees

Fuzzy decision trees exploit the popularity of decision tree algorithms for practical knowledge acquisition and the representative power of the fuzzy technology. They are extensions of Quinlan ID3 trees, with the tree-building routine modified to utilize fuzzy instead of strict domains, and with new inferences combining fuzzy defuzzification with the inductive methodology. Fuzzy decision trees represent the discovered rules most natural for human (for example thanks to the linguistic variables). As ID3 trees, they require that real-valued and multi-valued domains be partitioned prior to tree construction.

Till recently years, many fuzzy decision tree induction algorithms have been introduced [14,15]. The work in [15] takes a detailed introduction about the non fuzzy rules and the different kind of fuzzy rules.

In this point it is essential to refer that the data (real values) are partitioned into fuzzy sets by two different ways: (a) define linguistic values based on experts' knowledge into a range or (b) based on variable behavior data where it is possible to determine the number and the shape of sets. This approach consists on the following steps:

Step 1: A fuzzy clustering algorithm is used for input domain partition. The supervised method takes into account the class labels during the clustering. Therefore the fuzzy membership functions (fuzzy sets) represent not only the distribution of data, but the distribution of the classes too.

Step 2: During a pre-pruning method the resulted partitions could analyze and combine the unduly overlapped fuzzy sets.

Step 3: The results of the pre-pruning step are input parameters (beside data) for the tree induction algorithm. The applied tree induction method is the FID (Fuzzy Induction on Decision Tree) algorithm by C. Z. Janikow.

Step 4: The fuzzy ID3 is used to extract rules which are then used for generating fuzzy rule base.

Step 5: While the FID algorithm could generate larger and complex decision tree as it is necessary, therefore a post pruning method is applied. The rule which yields the maximal fulfillment degree in the least number of cases is deleted.

This method provides compact fuzzy rule base that can be used for building FCM-DSS.

4 Presentation of the Proposed Approach and Application Example

As it has already been stated, the central idea of the proposed technique is to combine different data driven methods to extract the available knowledge from data and to generate fuzzy If-Then rules. The resulted fuzzy rule base is applied to construct an

augmented FCM based clinical treatment simulation tool (CTST-FCM) used for decisions in radiation treatment planning. Then, a simple discriminant method is used for the characterization of output concepts of the resulting FCM-DSS. According to the desired values of output concepts, the augmented FCM-DSS reaches a decision about the acceptance of treatment planning technique.

In our approach, the FDTs algorithm is used because of the type of data for the problem of radiation therapy. The fuzzy decision tree algorithm proposed by Janikow [15] is an efficient one, providing fuzzy rule base that can be used to build advance FCM-DSS systems.

4.1 Application Problem

Radiotherapy is the application of ionizing radiation to cure patients suffering from cancer (and/or other diseases) and to eliminate infected cells, alone or combined with other modalities. The aim of radiation therapy is to design and perform a treatment plan how to deliver a precisely measured dose of radiation to the defined tumor volume with as minimal damage as possible to the surrounding healthy tissue.

In a previous work, a decision system based on human knowledge and experience had been proposed and developed by Papageorgiou et al., 2003. A two-level hierarchical structure with a FCM in each level had been created producing an Advanced Decision-Making System. The lower-level FCM modeled the treatment planning, taking into consideration all the factors and treatment variables and their influence. The upper-level FCM modeled the procedure of the treatment execution and calculated the optimal final dose for radiation treatment. The upper level FCM supervised and evaluated the whole radiation therapy process. The proposed two-level integrated structure for supervising the procedure before treatment execution seems a rather realistic approach to the complex decision making process in radiation therapy.

At this point, according to the AAPM protocols [16] and opinions of radiotherapists-doctors for the most important factors that should be taken into consideration (in order to achieve a good distribution of the radiation on the tumor, as well as to protect the healthy tissues, five factor concepts and eight selector-concepts were selected with discrete and fuzzy values for the determination of the output concepts. Now, a new FCM model that represents the radiotherapy treatment planning procedure according to the test packages, protocols and radiotherapists' opinions is designed and the new CTST-FCM is given in Figure 2.

The number of concepts has been reduced to 16 concepts thus to avoid the complexity of the previously developed CTST-FCM model and to be more clear the proposed technique to no specialist readers. Concepts F-C1 to F-C5 are the Factor-concepts, that represent the depth of tumor, the size of tumor, the shape of tumor, the type of the irradiation and the amount of patient thickness irradiated. Concepts S-C1 to S-C8 are the Selector-concepts, representing size of radiation field, multiple field arrangements, beam directions, dose distribution from each field, stationery vs. rotation-isocentric beam therapy, field modification, patient immobilizing and use of 2D or 3D conformal technique, respectively. The concepts OUT-C1 to OUT-C3 are the three Output-concepts. The value of the OUT-C1 represents the amount of dose applied to mean Clinical Target Volume (CTV), which have to be larger than the 90% of the amount of prescribed dose to the tumor. The value of concept OUT-C2 represents the amount of

the surrounding healthy tissues' volume received a dose, which have to be as less as possible, less than the 5% of volume received the prescribed dose and the value of concept OUT-C3 represents the amount of organs at risk volume received a dose, which have to be less than the 10% of volume received the prescribed dose [16].

After the description of CTST-FCM concepts, the design of FCM model continues with the determination of fuzzy sets for each one concept variable. For example, the FC2 has three fuzzy sets. Next, each expert was asked to define the degree of influence among the concepts using an if-then rule, as presented in [8].

Then the fuzzy decision tree algorithm was implemented to the initial clinical data and measurements and a set of fuzzy rules were produced. Some of the fuzzy rules that considered important to the decision making approach were selected from the fuzzy decision tree-based rule extraction technique according to the test packages and experimental data. Some of these rules are presented at follows:

If F-C1 is medium Then S-C1 is high
 If F-C1 is medium Then S-C2 is very high
 If F-C2 is high Then S-C1 is high
 If F-C2 is small and F-C3 is small Then S-C1 is very high
 If S-C4 is 1 and S-C6 is medium Then F-C5 is very high
 If F-C1 is small and F-C2 is small Then S-C3 is small

In this point, due to the large number of fuzzy rules produced by the fuzzy decision tree algorithm, we selected only those which differ from the initially suggested by experts and used for the reconstruction of the augmented CTST-FCM in radiation treatment planning. These rules accompanied by rules suggested by experts produce the new augmented CTST-FCM simulation tool for radiation therapy, which has new relationships among concepts and assigns new decisions and treatment planning suggestions.

5 Results and Discussion of Augmented FCM-DSS in Radiotherapy

Two case studies for the problem of prostate cancer therapy will be considered using the new CTST-FCM model, which consists of 16 concepts and 64 interconnections among concepts, in order to test the validity of the model. In the first case the 3-D conformal technique consisting of six-field arrangement is suggested and in the second one the conventional four-field box technique. Radiotherapy physicians and medical physicists choose and specified, in our previous study the fuzzy membership functions for the weights for each case study as well as the fuzzy rules according to their knowledge for each treatment planning procedure. The numerical weights between factor and selector concepts for the new CTST-FCM are given in Table 1 after the defuzzification process.

For the first case study, the conformal radiotherapy was selected with the following characteristics: the S-C2 takes the value of six-field number; S-C1 has the value of "small-size" for radiation field that means that the influence of S-C1 and S-C2 toward OUT-Cs is great. In the same way the S-C3 and S-C4 have great influence at OUT-Cs because different beam directions and weights of radiation beams are used. The S-C5

Table 1. Numerical weights among F-Cs, S-Cs and OUT-Cs of new CTST-FCM for the first case, as they derived from combined knowledge from experts and data

Concepts	S-C3	S-C4	S-C5	S-C6	S-C7	S-C8	S-C9	S-C10	OUT-C1	OUT-C2	OUT-C3
F-C1	0.7	0.75	0.4	0.4	0.65	0.6	0	0	0	0	0
F-C2	0.75	0.6	0	0.6	0.55	0.5	0.6	0.5	0	0	0
F-C3	0.6	0.7	0.45	0.2	0.4	0	0	0.75	0	0	0
F-C4	0.25	0.6	0.5	0.55	0.4	0.5	0	0.4	0	0	0
F-C5	0.5	0.6	0.6	0.5	0.2	0.5	0.6	0	0	0	0
S-C1	0	0	0	0	0	0	0	0	0.4	-0.4	-0.4
S-C2	0	0	0	0.5	0	0	0	0	0.3	-0.5	-0.4
S-C3	0	0	0	0	0	0	0	0	0.4	-0.3	-0.3
S-C4	0	0	0	0	0	0	0	0	0.4	-0.4	-0.4
S-C5	0	0	0	0	0	0.7	0	0	0.3	-0.3	-0.3
S-C6	0	0	0	0	0.6	0	0	0	0.4	-0.4	-0.4
S-C7	0	0	0	0	0	0	0	0	0.5	-0.5	-0.5
S-C8	0	0	0	0	0	0	0	0	0.6	-0.5	-0.5
OUT-C1	0	0	0	0	0	0	0	0	0	-0.6	-0.5
OUT-C2	0	0	0	0	0	0	0	0	-0.7	0	0
OUT-C3	0	0	0	0	0	0	0	0	-0.6	0	0

takes the discrete value of isocentric beam therapy. Concept S-C6 takes values for the selected blocks and/or wedges, influencing the OUT-Cs. The S-C7 takes a value for accurate patient positioning and the S-C8 takes the discrete value of 3-D radiotherapy.

The following initial vector is formed for this particular treatment technique:

$$A_1=[0.6 \ 0.55 \ 0.55 \ 0.6 \ 0.6 \ 0.4 \ 0.65 \ 0.7 \ 0.45 \ 0.6 \ 0.6 \ 0.5 \ 0.6 \ 0.5 \ 0.5 \ 0.45].$$

Using the eq. (1), the resulting CTST-FCM starts to interact and simulates the radiation procedure. New values of concepts were calculated after 8 simulation steps. The following vector gives the calculated values of concepts in the equilibrium region.

$$A1_new=[\ 0.6590 \ 0.6590 \ 0.6590 \ 0.6590 \ 0.6590 \ 0.9420 \ 0.9568 \ 0.8988 \ 0.9412 \ 0.9515 \ 0.9585 \ 0.8357 \ 0.8770 \ 0.9813 \ 0.0203 \ 0.0336].$$

In the steady state, the following values of OUT-Cs are: for OUT-C1 is 0.9813, for OUT-C2 is 0.0201 and for OUT-C3 is 0.0336. Based on the referred protocols [18,19], the calculated values of output concepts are accepted. The calculated value of OUT-C1 is 0.981, which means that the CTV receives the 98% of the amount of the prescribed dose, which is accepted. The value of OUT-C2 that represents the amount of the surrounding healthy tissues' volume received a dose was found equal to 0.0201, so the 2.01% of the volume of healthy tissues receives the prescribed dose, and the OUT-C3 was found equal to 3.36% of the dose received from organs at risk.

In the second case study, the conventional four-field box technique is implemented for the prostate cancer treatment. This technique is consisted of a four-field box arrangement with gantry angles 0, 90, 180, and 270. For this case, the new CTST-FCM was reconstructed which means that the cause-effect relationships and weights have been reassigned not only from radiotherapists' suggestions but also from data knowledge using the proposed rule extraction technique. For this case, the Selector-concept S-C2 has the value of four-field number; S-C1 has the value of "large-size" of radiation field, which means that the influence of S-C1 and S-C2 toward OUT-Cs is very low. In the same way, the S-C3 and S-C4 have lower influence on OUT-Cs because different beam directions and weights of radiation beams are used. The S-C5 takes the

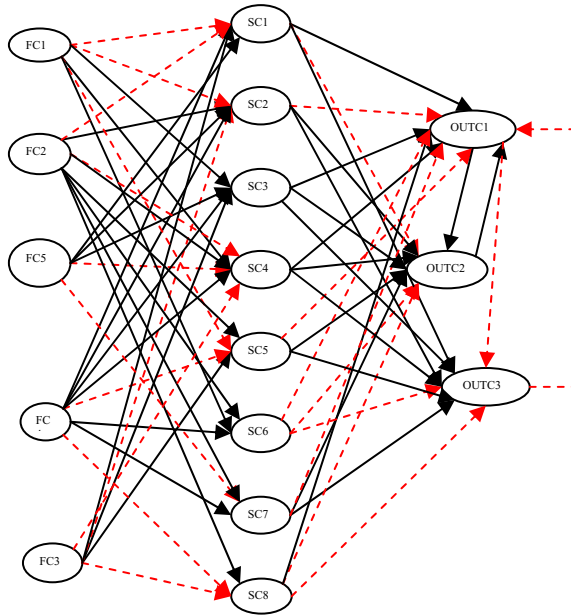


Fig. 2. The new CTST-FCM tool for decision making in radiotherapy after combining knowledge from experts and data (the broken lines are the new or changed weight values)

discrete value of isocentric beam therapy and has the same influence on OUT-Cs as the above conformal treatment case. S-C6 has zero influence on OUT-Cs because no blocks (and/or no wedges and any filters) are selected for this treatment case. The S-C7 takes a low value for no accurate patient positioning and the S-C8 takes the discrete value of 2-D radiotherapy. The numerical weights among F-Cs, S-Cs and OUT-Cs, of new CTST-FCM for the second case study, are given in Table 2.

Using this new CTST-FCM model, with the new modified weight matrix, the simulation of the radiotherapy procedure for this case starts with the following initial values of concepts:

$$A_2 = [0.5 \ 0.48 \ 0.4 \ 0.6 \ 0.5 \ 0.7 \ 0.45 \ 0.4 \ 0.6 \ 0.6 \ 0.3 \ 0.2 \ 0.4 \ 0.4 \ 0.2 \ 0.2]$$

The final values of OUT-Cs are as follows: for OUT-C1, 0.9533; for OUT-C2, 0.0830; and for OUT-C3, 0.1133 (illustrated in the following vector A_{2_new}).

$A_{2_new} = [0.6590 \ 0.6590 \ 0.6590 \ 0.6590 \ 0.6590 \ 0.9420 \ 0.9568 \ 0.8988 \ 0.9412 \ 0.9515 \ 0.9585 \ 0.8357 \ 0.8770 \ 0.9533 \ 0.0830 \ 0.1133]$. These values for OUT-C2 and OUT-C3 are not accepted according to related protocols [16].

The new augmented CTST-FCM model, with less number of concepts and weights and especially with weights not only determined by radiotherapists-experts' suggestions but also by knowledge extracted from fuzzy decision tree-based rule extraction technique, is a dynamic and less complex model which works efficiently. This radiation therapy decision making tool can adapt its knowledge from available data and not only from experts opinions. Thus, through the proposed approach, an acceptable decision is succeeded and the new CTST-FCM tool is less time consuming and easy for use from no specialists.

Table 2. Numerical weights among F-Cs, S-Cs and OUT-Cs of new CTST-FCM for the second case

Concepts	S-C3	S-C4	S-C5	S-C6	S-C7	S-C8	S-C9	S-C10	OUT-C1	OUT-C2	OUT-C3
F-C1	0.7	0.75	0.4	0.4	0.6	0.6	0	0	0	0	0
F-C2	0.75	0.6	0	0.6	0.55	0.5	0.6	0.5	0	0	0
F-C3	0.6	0.7	0.45	0.2	0.4	0	0	0.75	0	0	0
F-C4	0.25	0.6	0.5	0.5	0.4	0.5	0	0.4	0	0	0
F-C5	0.5	0.6	0.6	0.5	0.2	0.5	0.6	0	0	0	0
S-C1	0	0	0	0	0	0	0	0	0.3	-0.4	-0.3
S-C2	0	0	0	0.5	0	0	0	0	0.25	-0.5	-0.4
S-C3	0	0	0	0	0	0	0	0	0.3	-0.3	-0.3
S-C4	0	0	0	0	0	0	0	0	0.25	-0.2	-0.2
S-C5	0	0	0	0	0	0.7	0	0	0.3	-0.3	-0.3
S-C6	0	0	0	0	0.6	0	0	0	0.2	0	0
S-C7	0	0	0	0	0	0	0	0	0.4	-0.3	-0.3
S-C8	0	0	0	0	0	0	0	0	0.4	-0.4	-0.4
OUT-C1	0	0	0	0	0	0	0	0	0	-0.4	-0.4
OUT-C2	0	0	0	0	0	0	0	0	-0.7	0	0
OUT-C3	0	0	0	0	0	0	0	0	-0.6	0	0

6 Conclusion

In this investigation, a different dynamic approach for construction of FCM-based decision support tools presented. The main goal of the proposed methodology was not to achieve better accuracies or to present a better classifier, but to investigate an efficient enhancement of FCM-DSS accompanied by fuzzy rule base that has constructed by sufficient extraction of knowledge methods. The new decision support tool seems simple, no time consuming and less complex to be accepted for medical applications. The distinguishing feature of such augmented FCM-DSS is its situations with large amount of data, not enough knowledge from experts and difficulty to handle the available knowledge from many different sources of information. In our opinion using this fuzzy rule based decision support system in the physicians’ education process provides a more useful environment for the students than huge, hard-covered materials.

Acknowledgment

The research was supported in part by the European Commission’s Seventh Framework Information Society Technologies (IST) Programme, Unit ICT for Health, project DEBUGIT (no. 217139).

References

- [1] Dhar, V., Stein, R.: Intelligent Decision Support Methods: The Science of Knowledge Work. Prentice-Hall, Upper Saddle River (1997)
- [2] Zurada, J.M., Duch, W., Setiono, R.: Computational intelligence methods for rule-based data understanding. In: Proc. of the IEEE International Conference on Neural Networks, vol. 92(5), pp. 771–805 (2004)

- [3] Mitra, S., Hayashi, Y.: Neuro-Fuzzy rule generation: Survey in soft computing. *IEEE Trans. Neural Networks* 11(3), 748–760 (2000)
- [4] Nauck, U.: Design and implementation of a neuro-fuzzy data analysis tool in Java, Master's thesis, Technical, University of Braunschweig, Braunschweig (1999)
- [5] Stylios, C.D., Georgopoulos, V.C., Malandraki, G.A., Chouliara, S.: Fuzzy cognitive map architectures for medical decision support systems. *Appl. Soft Comput.* 8(3), 1243–1251 (2008)
- [6] Papageorgiou, E.I., Spyridonos, P., Ravazoula, P., Stylios, C.D., Groumpos, P.P., Niki-foridis, G.: Advanced Soft Computing Diagnosis Method for Tumor Grading. *Artif. Intell. Med.* 36, 59–70 (2006a)
- [7] Papageorgiou, E.I., Stylios, C.D., Groumpos, P.P.: A Combined Fuzzy Cognitive Map and Decision Trees Model for Medical Decision Making. In: *Proceedings of the 28th IEEE EMBS Annual Intern. Conference in Medicine and Biology Society*, New York, August 30– September 3, pp. 6117–6120 (2006b)
- [8] Papageorgiou, E.I., Stylios, C.D., Groumpos, P.: An Integrated Two-Level Hierarchical Decision Making System based on Fuzzy Cognitive Maps (FCMs). *IEEE Trans. Biomed. Engin.* 50(12), 1326–1339 (2003)
- [9] Kosko, B.: Fuzzy Cognitive Maps. *International Journal of Man-Machine Studies* 24, 65–75 (1986)
- [10] Lee, K.C., Lee, W.J., Kwon, O.B., Han, J.H., Yu, P.I.: Strategic Planning Simulation Based on Fuzzy Cognitive Map Knowledge and Differential Game. *Simulation* 75(5), 316–327 (1998)
- [11] Bueno, S., Salmeron, J.L.: Benchmarking main activation functions in fuzzy cognitive maps. *Expert Systems with Applications* 36(3), 5221–5229 (2009)
- [12] Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.): *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, Menlo Park (1996)
- [13] Kurgan, L.A., Musilek, P.: A Survey on Knowledge Discovery and Data mining processes. *The Knowledge Engineering Review* 21(1), 1–24 (2006)
- [14] Janikow, C.Z.: Fuzzy Decision Trees Manual, free version for Fuzzy Decision Trees (1998), <http://www.cs.umsl.edu/Faculty/janikow/janikow.html>
- [15] Janikow, C.Z.: Fuzzy decision trees: issues and methods. *IEEE Trans. Systems Man Cybernet. Part B (Cybernetics)* 28(1), 1–14 (1998)
- [16] AAPM Report No. 55, American Association of Physicists in Medicine, Report of Task Group 23 of the Radiation Therapy Committee, Radiation Treatment planning dosimetry verification. American Institution of Physics, Woodbury (1995)

Fuzzy Cognitive Map Based Approach for Assessing Pulmonary Infections

Elpiniki I. Papageorgiou^{1,*}, Nikolaos Papandrianos², Georgia Karagianni³,
G. Kyriazopoulos³, and D. Sfyras³

¹ Department of Informatics & Computer Technology, Technological Educational Institute of Lamia, 3rd Km Old National Road Lamia-Athens, 35100 Lamia, Greece
epapageorgiou@teilam.gr

² Department of Nuclear Medicine, University General Hospital of Patras, 26500 Patras, Greece
nikpapan@upatras.gr

³ Department of Intensive Care Unit, Lamia General Hospital, 35100 Lamia, Greece
gikaragianni@yahoo.gr, dimitriosfyras@yahoo.gr

Abstract. The decision making problem of predicting infectious diseases is a complex process, because of the numerous elements/parameters (such as symptoms, signs, physical examination, laboratory tests, cultures, chest x-rays, e.t.c.) involved in its operation, and a permanent attention is demanded. The knowledge of physicians according to the physical examination and clinical measurements is the main point to succeed a diagnosis and monitor patient status. In this paper, the Fuzzy Cognitive Mapping approach is investigating to handle with the problem of pulmonary infections during the patient admission into the hospital or in Intensive Care Unit (ICU). This is the first step in the development of a decision support system for the process of infectious diseases prediction.

1 Introduction

During the last years, an enormous number of decision support systems (DSS) for diverse medical problems have been developed. The traditional medical expert systems [1], were equipped with a rule knowledge base supplied by the domain experts (physicians). On the basis of rules inserted in the expert system, it is possible to classify new instances of medical observations by matching symptoms to the conditional part of a rule and then to perform forward and backward reasoning to achieve the diagnosis or construct a therapy plan. In our opinion, one of the main disadvantages for the application of the classic rule-based knowledge representation in medical DSS is its limitation of representing some of the more complex associations that may be experienced within the medical data. For example, in a rule-based DSS, the representation of the complex phenomenon of causality [2] is, in fact, left to the interpretation and expertise of the doctor.

There are a vast number of knowledge-representation methods that can be considered, in general, as exemplification of the conceptual modeling approach. The

* Corresponding author.

best-known of them are ontologies and semantic networks that are able to express concepts and relationships among them. Maybe less known in computer science are fuzzy cognitive maps (FCMs).

FCM is a soft computing technique capable of dealing with situations including uncertain descriptions using similar procedure such as human reasoning does [3,4]. FCMs are originated from cognitive maps and are used to model knowledge and experience for describing particular domains using nodes-concepts (representing i.e. variables, states, inputs, outputs) and the relationships between them in order to outline a decision-making process.

In this work, the process of making medical diagnoses is our primary attention. The FCM approach is used as a first step, to model a physician-expert's behavior in the decision making [5]. The behavior to be modeled is centered in the decision making process, whose reasoning implies to reach a predefined goal, coming from one or more initial states. Therefore, the reasoning system will be more efficient when a least number of transitions to reach the final goal are achieved. They have been used in many different scientific fields for modeling and decision making and a special attention given in medical diagnosis and medical decision support through the recently works [6-8].

FCM was chosen because of the nature of the application problem. The prediction of infectious diseases in pneumonia is a complex process with sufficient interacting parameters and FCMs have been proved suitable for this kind of problems. To the best of our knowledge, no any related work has been done till today on implementing FCMs to handle with the specific problem of defining factors as well as their complex cause-effect relationships that affecting infectious diseases and/or adverse events in Intensive Care Unit. Therefore, this is the first step in the development of an expert system tool that will help in decision making process in medicine, through the design of the knowledge representation and the design of reasoning with FCM to automate the decision.

2 Main Aspects of Fuzzy Cognitive Maps

A FCM is a representation of a belief system in a given domain. It comprises of concepts (C) representing key drivers of the system, joined by directional edges of connections (w) representing causal relationships between concepts. Each connection is assigned a weight w_{ij} which quantizes the strength of the causal relationship between concepts C_i and C_j [3]. A positive weight indicates an excitatory relationship, i.e. as C_i increases C_j increases while a negative weight indicates an inhibitory relationship, i.e. as C_i increases C_j decreases. In its graphical form, A FCM provides domain knowledge as a collection of "circles" and "arrows" that is relatively easy to visualize and manipulate. Key to the tool is its potential to allow feedback among its nodes, enabling its application in domains that evolve over time. It is particularly suited for use in soft-knowledge domains with a qualitative rather than a quantitative, emphasis. The tool is said to be semi-quantitative, because of the quantification of drivers and links can be interpreted in relative terms only [4].

Fig. 1 shows a fuzzy cognitive map consisting of a number of concepts, some of them are input concepts and the rest are decision (output) concepts, as well as their

fuzzy interactions. The main objective of building a fuzzy cognitive map around a problem is to be able to predict the outcome by letting the relevant issues interact with one another.

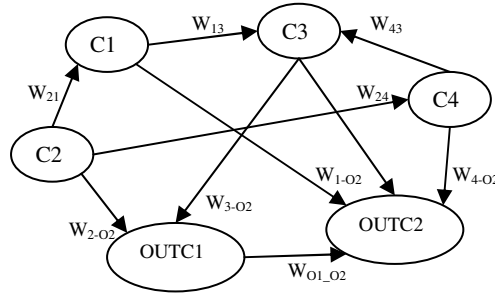


Fig. 1. A generic FCM model for decision making

The concepts C_1, C_2, \dots, C_n , (where n is the number of concepts in the problem domain) represent the drivers and constrains that are considered of importance to the issue under consideration. The link strength between two nodes C_i and C_j as denoted by W_{ij} , takes values within $[-1,1]$. If the value of this link takes on discrete values in the set $\{-1, 0, 1\}$, it is called a simple or crisp FCM. The concept values of nodes C_1, C_2, \dots, C_n together represent the state vector V . The state vector takes values usually between 0 and 1. the dynamics of the state vector is the principal output of applying a FCM. To let the system evolve, the state vector V is passed repeatedly through the FCM connection matrix W . This involves multiplying V by W , and then transforming the result as follows:

$$V = f(V + V \cdot W) \tag{1}$$

or

$$V_i(t+1) = f(V_i(t) + \sum_{\substack{j \neq i \\ j=1}}^N V_j(t) \cdot W_{ji}) \tag{2}$$

where $V_i(t)$ is the value of concept C_i at step t , $V_j(t)$ is the value of concept C_j at step t , W_{ji} is the weight of the interconnection from concept C_j to concept C_i and f is the threshold function that squashes the result of the multiplication in the interval $[0, 1]$, $[9]$. We use the function $f(x)$:

$$f(x)=1/(1+\exp(-mx)) \tag{3}$$

where m is a real positive number ($m=1$) and x is the value $V_i(t)$ on the equilibrium point.

2.1 Construction of Fuzzy Cognitive Maps

Usually, a group of experts, who operate, monitor, supervise and know the system behaviour, are used to construct the FCM model. The experts, based on their experience, assign the main factors that describe the behaviour of the system; each of these

factors is represented by one concept of the FCM. They know which elements of the systems influence other elements, thus they determine the negative or positive effect of one concept on the others, with a fuzzy degree of causation for the corresponding concepts. The development methodology extracts the available knowledge from the experts by a form of fuzzy “if-then” rules. The following form of rules is assumed, where **A**, **B** and **C** are linguistic variables:

*IF value of concept C_i is **A** THEN value of concept C_j is **B** and thus the linguistic weight e_{ij} is **C** (from the set $T(\text{influence})$)*

Each interconnection associates the relationship between the two concepts and determines the grade of causality between the two concepts. The causal interrelationships among concepts are usually declared using the variable *Influence* which is interpreted as a linguistic variable taking values in the universe $U=[-1,1]$. Its term set $T(\text{influence})$ is suggested to comprise twelve variables. Using twelve linguistic variables, an expert can describe in detail the influence of one concept on another and can discern between different degrees of influence. The twelve variables used here are: $T(\text{influence}) = \{\text{negatively very strong, negatively strong, negatively medium, negatively weak, negatively very weak, zero, positively very weak, positively weak, positively medium, positively strong, positively very strong, positively very very strong}\}$.

Then, the linguistic variables **C** proposed by the experts for each interconnection are aggregated using the SUM method and so an overall linguistic weight is produced which is defuzzified with the Centre of Gravity method and finally a numerical weight for W_{ij} is calculated. Using this method, all the weights of the FCM model are inferred.

3 Fuzzy Cognitive Map Approach to Assess Pulmonary Infections

The FCM is suitable technique to cope with complex decision making tasks such as the prediction of infection, the severity of infectious disease and the therapy plan acceptance. It is simple, no time consuming and exploits experience and accumulated knowledge from experts.

A large number of parameters, factors, constraints and different conditions exist in the complex problem of pulmonary infections [10,11]. For the problem of pneumonia, a number of typical symptoms are associated including fever (80%) often accompanied by chills or hypothermia in a small group of patients, altered general well-being and respiratory symptoms such as cough (90%), expectoration (66%), dyspnea-shortness of breath (66%), pleuritic pain-a sharp or stabbing pain, experienced during deep breaths or coughs (50%), and hemoptysis-expectoration of blood (15%). The initial presentation is frequently acute, with an intense and unique chill. Productive cough is present and the expectoration is purulent or bloody. Pleuritic pain may be present.

Physical examination reveals typical findings of pulmonary consolidation- bronchial breath sounds, bronchophony, crackles, increased fremitus, dullness during percussion, tachypnea-increased respiratory rate, tachycardia-high heart rate (pulse should increase by 10 beats per minute per degree Celsius of temperature elevation) or a low oxygen saturation, which is the amount of oxygen in the blood as indicated by either pulse oximetry or blood gas analysis. In elderly and immunocompromised patients, the signs and symptoms of pulmonary infection may be muted and overshadowed by nonspecific complaints. If pneumonia is suspected on the basis of a patient's symptoms and

findings from physical examination, further investigations are needed to confirm the diagnosis. From the lab tests only the WBC have been considered as the most important one to increase mainly the risk of infection. These data provide a logical basis for evaluation the risk of infection and the need for intensive care.

Table 1. Factor concepts coding pulmonary infection

<i>Concepts</i>	<i>Type of values</i>
C1: Dyspnea	Four fuzzy values (no dyspnea, less serious, moderate serious, serious dyspnea state)
C2: Cough	Three fuzzy values (no cough, non-productive and productive)
C3: Rigor/chills	Two discrete values (exist or no)
C4: Fever	Six Fuzzy values (hypothermia (34-36 ⁰), no fever (36-38,4 ⁰), low grade (38.5-38.9 ⁰), moderate, high grade (39.5-40.9 ⁰), hyperpyrexia (>41 ⁰))
C5: Loss of appetite	Two discrete values (0,1)
C6: Debility	Four fuzzy values (no, small, moderate, large)
C7: Pleuritic pain	Two discrete values (0, 1)
C8: Heamoptysis	Two discrete values (0, 1)
C9:Oxygen requirement	Four fuzzy values (no need of oxygen, low, medium and high)
C10: Tachypnea	Four fuzzy values (normal (12-24), moderate (25-38), severe (35-49) and very severe (>50))
C11:Acoustic characteristics	Three fuzzy values (no rales, localized and generalized)
C12:GCS	Three fuzzy values: (Severe altered mental status, GCS ≤ 8 , Moderate, GCS 9 - 12 , Minor altered mental status, GCS ≥ 13)
C13: Systolic Blood Pressure	Seven fuzzy values (Hypotension <90, Optimal <120 , Normal <130, High-normal 130-139, Grade 1 hypertension 140 – 159 Grade 2 hypertension 160-179 Grade 3 hypertension ≥180 (<i>British hypertension society</i>)
C14: Diastolic blood pressure	Seven fuzzy values (Hypotension <60, Optimal <80, Norma l<85, High-normal 85-89, Grade 1 hypertension 90-99 Grade 2 hypertension 100-109, Grade 3 hypertension ≥110 (<i>British hypertension society</i>)
C15: Tachycardia	Four fuzzy values (low (less than 80 beats/min), normal (90-110), moderate sevre (110-140), severe (>140))
C16:Radiologic evidence of pneumonia	Two discrete (exist or no)
C17: Radiologic evidence of complicated pneumonia	Two fuzzy values (presence, absence)
C18: pH	Three fuzzy values (Acidosis <7.35, Normal 7,35 – 7,45 , Alkalosis >7.45)
C19:pO2	Two fuzzy value (<i>normal</i> 70-100mmHg, <i>hypoxia</i> is every value under normal)
C20: pCO2	Three fuzzy values: (normal 35-45mmHg , hypocapnia <35 mmHg , hypercapnia >45mmHg)
C21: sO2%	Two fuzzy values: (normal >95%, hypoxia <95%)
C22: WBC	Three fuzzy values: (Normal 4,3 - 10x10 ³ /µl leukocytosis>10x10 ³ /µl, leukopenia<1000/µl)
C23: Immunocompromise	Two fuzzy values: (presence, absence)
C24: Comorbidities	Two discrete values: (presence=1, absence=0)
C25: Age	Three fuzzy: (Young, middle age, older)
OUT-C: Risk of pulmonary infection	Five fuzzy values: (very low, low, moderate, high, very high)

Three physicians-experts were pooled to define the number and type of parameters-factors affecting the problem of pulmonary infection. Two of the physicians were from the General Hospital of Lamia, and one from the University General Hospital of Patras, Greece. The factors are represented in Table 1 and are well documented in bibliography. These factors assign the main variables that play an important role in the final diagnostic decision about the risk of pulmonary infection and are the concepts of the FCM. The concept values can take two, three, four or five possible discrete or fuzzy values, as shown in Table 1.

These 26 concepts are the factor concepts representing the main variables that physicians in ICU usually take into consideration in assigning the existent and the grade of the infection. The output (decision) concept (OUT-C) represents the risk of pulmonary infection in percentage and takes *five fuzzy values (very low, low, moderate, high, very high)*.

After the description of FCM concepts, each expert was asked to define the degree of influence among the concepts and to describe their interrelationship using an IF—THEN rule, assuming the following statement where C_i and C_j are all the ordered pair of concepts:

IF a {no, very small, small, medium, large, very large} change occurs in the value of concept C_i **THEN** a {no, very small, small, medium, large, very large} change in value of concept C_j is caused. **THUS** the influence of concept C_i on concept C_j is *T(influence)*.

Then, experts inferred a linguistic weight to describe the cause and effect relationship between every pair of concepts. To illustrate how numerical values of weights are produced, the three experts' suggestions on how to indicate the interconnection between concept C22 (number of white blood cells) and concept OUT-C (risk of pulmonary infection) are shown below:

1st expert:

IF a small change occurs in the value of concept C22, THEN a medium change in value of concept OUT-C is caused.

Infer: The influence from C22 to OUT-C is positive medium.

2nd expert:

IF a small change occurs in the value of concept C22, THEN a large change in value of concept OUT-C is caused.

Infer: The influence from C22 to OUT-C is positive high.

3rd expert:

IF a very small change occurs in the value of concept C22, THEN a large change in value of concept OUT-C is caused.

Infer: The influence from C22 to OUT-C is positive very high.

These linguistic variables (medium, positive strong and positive very strong) are summed and an overall linguistic weight is produced, which with the defuzzification method of CoG is transformed into the numerical value of $W_{22-OUTC}=0.617$.

The 26 identified concepts (Table 1) keep relations with each other, in order to characterize the process of predicting the risk of pulmonary infectious diseases and to provide a first front-end decision tool about the prediction of pulmonary infection. All the corresponding relations between concepts are given in the following Table 2.

Table 2. Linguistic weights assigned by each one of three experts and the produced numerical weight

	<i>OUT-C</i>	<i>OUT-C</i>	<i>OUT-C</i>	<i>Numerical weight</i>
	<i>First Expert</i>	<i>Second Expert</i>	<i>Third expert</i>	
C1-Dyspnea	Very weak	medium	weak	0.311
C2-Cough	v. weak	med	Very weak	0.2
C3-Rigor	weak	med	weak	0.345
C4-Fever	med	weak	strong	0.448
C5-appetite	v. weak	weak	v. weak	0.20
C6-debility	Med	Med	med	0.50
C7-chest pain	v. weak	weak	v. weak	0.15
C8- hemoptysis	strong	Very strong	strong	0.691
C9-ox.req.	weak	med	weak	0.345
C10-tachypnea	v.weak	v.weak	weak	0.20
C11- rales	weak	weak	v.weak	0.260
C12- GCS	Neg.med	Neg.weak	Neg.med	-0.455
C13-systolic	med	strong	strong	0.584
C14-diastolic	weak	strong	strong	0.50
C15-heart rate	weak	med	med	0.40
C16-infiltrate	med	strong	strong	0.584
C17-rad. complicated evidence	strong	v. strong	v. strong	0.740
C18-pH	weak	v.weak	v.weak	0.20
C19-PO2	v.weak	v.weak	weak	0.20
C20-pCO2	v.weak	v.weak	weak	0.20
C21-sO2	weak	med	weak	0.345
C22-WBC	strong	v.strong	med	0.617
C23-immunoc	weak	weak	med	0.345
C24-comorbid	weak	strong	strong	0.50
C25-Age	med	med	weak	0.40

Some more linguistic relationships among concepts exist which are not included in the above Table 2. There is one influence from concept C16-(describe the radiology evidence) towards concept C4 representing “fever”. This influence from C16 to C4 is medium. Also there are influences from C25 (Age) to C12 (GCS), from C4 to C22 (WBC), from C4 to C5 (loss of appetite), from C4 to C15, from C21 to C1, from C21 to C10, from C21 to C9, from C21 to C19, from C19 to C1, from C19 to C10, from C19 to C9, from C20 to C1, from C20 to C18. The influence from C25 (Age) to C12 (GCS) is negatively weak (-0.4). The influence from C4 to C22 is strong (0.7). The influence from C4 to C5 is low and the produced numerical weight is equal to 0.3.

The same approach was used to determine all the weights of the FCM. Figure 3 illustrates the FCM model for predicting the risk of pulmonary infection with the assigned numerical values of weights.

The proposed method based on FCMs for predicting pulmonary infection provides a framework within which physicians evaluate a series of traditional diagnostic concepts (symptoms, signs, laboratory tests, chest x-rays, risk and other factors). The way the FCM prediction model is designed increases the objectivity of the diagnostic process by taking into account the different physicians’ opinions regarding the interplay of factor and selector variables in the diagnostic output/decision. Using these variables, the FCM model predicts the risk percentage of the pulmonary infection.

In the next section, two different case studies have been considered for the model simulation and the operation of the FCM tool for the specific approach is evaluated and summarized.

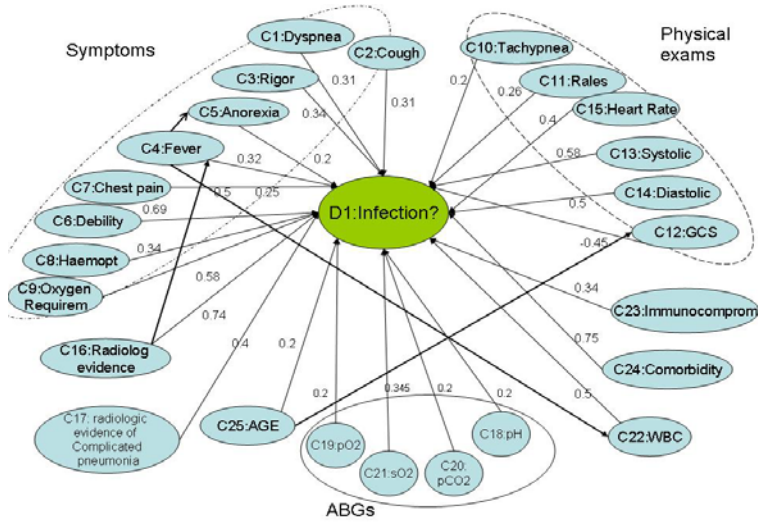


Fig. 3. The FCM model for assessing the risk of pulmonary infection

4 Simulations for Two Case Studies

In each of the test scenarios we have an initial vector V_i , representing the presented events at a given time of the process, and a final vector V_f , representing the last state that can be arrived at.

For the interpretation of the results, an average only for the output value of the decision concept OUT-C is computed according to the following criteria:

$$R(x) = \begin{cases} 0, & x \leq 0.5 \\ \frac{x-0.5}{0.5} \times 100\%, & 0.5 < x \leq 1 \end{cases}$$

where 0 represents the characteristic of the represented process by the concept is null, and 1 represents, the characteristic of the process represented by the concept is present 100%. For the specific approach, the function $R(x)$ gives the risk of pulmonary infection in percentage. When $R(\text{value of OUT-C})=1$, then the risk is 100%. The final value of decision concept D1 applying this criterion is denoted by OUT-C_f .

Thus $D1_f = R(V_f(26))$. This criterion can be modified according with the expert judgment. The final vector V_f is the last vector produced in convergence region and the 26th value of this vector is the OUT-C_f , the final value of decision concept.

The algorithm used to obtain the final vector \mathbf{V}_f is the following:

- (1) Definition of the initial vector \mathbf{V} that corresponds to the concepts identified in Table 1.
- (2) Multiply the initial vector \mathbf{V} and the matrix \mathbf{W} defined by experts by the eq.(2).
- (3) The resultant vector is updating using Eqs. (1)–(3).
- (4) This new vector is considered as an initial vector in the next iteration.
- (5) Steps 2–4 are repeated until $\mathbf{V}^t - \mathbf{V}^{t-1} \leq e = 0.001$.

The FCM performance is illustrated by means of simulation of three case scenarios in case of medium risk, high risk and very high risk of pulmonary infection.

First Scenario: For this scenario, an immunocompromised patient ($C_{23}=1$) has been considered, with high Fever ($C_4=0.7$), loss of appetite ($C_5=1$), high systolic blood pressure ($C_{13}=0.7$), with radiologic evidence present in chest x-rays ($C_{16}=1$) and small number of WBCs ($C_{22}=0.4$). Is infection existent and which is the probability risk of infection?

The initial concept vector is: $\mathbf{V1}=[0 \ 0 \ 0 \ 0.7 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0.7 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0.4 \ 1 \ 0 \ 0 \ 0]$. After the FCM simulation process described in previous five steps the system converges in a steady state with the final concept vector to be: $\mathbf{V1}_f=[0 \ 0 \ 0 \ 0.7000 \ 0.7163 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0.7000 \ 0 \ 0 \ 1.0000 \ 0 \ 0 \ 0 \ 0.7660 \ 0 \ 0.4000 \ 1.000 \ 0 \ 0 \ 0.9710]$.

The final value of decision concept $V1_f(26)=0.9710$, which following the above criterion ($OUT-C_f=R(0.9710)$) correspond to the 91,34% of risk, thus means that the risk of infection is very high according to the related fuzzy regions, initially prescribed.

Second Scenario: For this scenario, an old patient ($C_{25}=0.8$) has been considered, with altered mental status ($C_{12}=0.4$), with high oxygen requirements ($C_9=0.8$), and normal number of leukocytes-WBC ($C_{22}=0$). Is the infection existent and which is the probability risk of infection?

The initial concept vector is: $\mathbf{V2}=[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0.8 \ 0 \ 0 \ 0.4 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0.8 \ 0]$. After the FCM simulation process described in previous five steps the system converges in a steady state with the final concept vector to be: $\mathbf{V2}_f=[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0.8 \ 0 \ 0 \ 0.559 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0.8 \ 0.748]$.

The final value of decision concept is $V2_f(26)=0.7483$, and following the above criterion, corresponds to the 49.66% of risk, thus means that the risk of pulmonary infection is medium according to the related fuzzy regions in concept description.

We have tested our system using data taken from real medical cases. Unfortunately, the presentation of the set of all contributed parameters and the entire FCM decision support module fall out beyond the scope of this paper. This is the first attempt to construct and present the FCM tool that will help on the prediction of risk in pulmonary infections.

5 Conclusions

The knowledge-based approach used in this work focuses on the soft computing technique of fuzzy cognitive maps to address the issue of pulmonary risk prediction. The modeling methodology using the FCM tool was applied as a part of decision support

with a view to help medical and nursing personnel to assess patient status, assist in making a diagnosis, and facilitate the selection of a course of antibiotic therapy. It was demonstrated that FCMs can be a useful tool for capturing the physicians' understanding of the system and their perceptions on the medical requirements of the infectious diseases management. The main advantage of the proposed FCM tool in medical support is the sufficient simplicity and interpretability for physicians in decision process, which make it a convenient consulting tool in predicting the risk of infectious diseases. Our future work will be directed towards the insertion of other knowledge schemes into this approach, thus to enhance the performance of the suggested tool.

Acknowledgment

The research was supported in part by the European Commission's Seventh Framework Information Society Technologies (IST) Programme, Unit ICT for Health, project DEBUGIT (no. 217139).

References

- [1] Hudson, D.L.: Medical Expert Systems, Encyclopedia of Biomedical Engineering. John Wiley and Sons, Chichester (2006)
- [2] Pearl, J.: Causality, Models Reasoning and Inference. Cambridge University Press, Cambridge (2000)
- [3] Kosko, B.: Fuzzy Cognitive Maps. International Journal of Man-Machine Studies 24(1), 65–67 (1986)
- [4] Miao, Y., Liu, Z.Q.: On causal inference in fuzzy cognitive maps. IEEE Transactions on Fuzzy Systems 8, 107–119 (2000)
- [5] Papageorgiou, E.I., Stylios, C.D., Groumpos, P.: An Integrated Two-Level Hierarchical Decision Making System based on Fuzzy Cognitive Maps (FCMs). IEEE Trans. Biomed. Engin. 50(12), 1326–1339 (2003)
- [6] Papageorgiou, E.I., Stylios, C.D., Groumpos, P.P.: Novel architecture for supporting medical decision making of different data types based on Fuzzy Cognitive Map Framework. In: Proceedings of 28th IEEE EMBS 2007, Lyon, France, August 21–23 (2007)
- [7] Papageorgiou, E.I., Spyridonos, P., Glotsos, D., Stylios, C.D., Ravazoula, P., Nikiforidis, G., Groumpos, P.P.: Brain tumour characterization using the soft computing technique of fuzzy cognitive maps. Applied Soft Computing 8, 820–828 (2008)
- [8] Stylios, C.D., Georgopoulos, V.C.: Fuzzy Cognitive Maps Structure for Medical Decision Support Systems. Studies in Fuzziness and Soft Computing 218, 151–174 (2008)
- [9] Bueno, S., Salmeron, J.L.: Benchmarking main activation functions in fuzzy cognitive maps. Expert Systems with Applications 36(3), 5221–5229 (2009)
- [10] Hoare, Z., Lim, W.S.: Pneumonia: update on diagnosis and management. BMJ 33, 1077–1079 (2006), <http://www.bmj.com/cgi/content/full/332/7549/1077>
- [11] Gennis, P., Gallagher, J., Falvo, C., Baker, S., Than, W.: Clinical criteria for the detection of pneumonia in adults: guidelines for ordering chest roentgenograms in the emergency department. The Journal of emergency medicine 7(3), 263–268 (1989)

A Knowledge-Based Framework for Information Extraction from Clinical Practice Guidelines

Corrado Loglisci, Michelangelo Ceci, and Donato Malerba

Dipartimento di Informatica, Università degli Studi di Bari
via Orabona, 4 - 70126 Bari, Italy
{loglisci,ceci,malerba}@di.uniba.it

Abstract. Clinical Practice Guidelines guide decision making in decision problems such as the diagnosis, prevention, etc. for specific clinical circumstances. They are usually available in the form of textual documents written in natural language whose interpretation, however, can make difficult their implementation. Additionally, the high number of available documents and the presence of information for different decision problems in the same document can further hinder their use. In this paper, we propose a framework to extract practices and indications considered to be important in a particular clinical circumstance for a specific decision problem from textual clinical guidelines. The framework operates in two consecutive phases: the first one aims at extracting pieces of information relevant for each decision problem from the documents, while the second one exploits pieces of information in order to generate a structured representation of the clinical practice guidelines for each decision problem. The application to the context of Metabolic Syndrome proves the effectiveness of the proposed framework.

Keywords: Information Extraction, Clinical Practices Guidelines, Medical Decision Making.

1 Introduction

Clinical practice guidelines (CPGs) are systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances [5]. They can support medical personnel in deciding the activities to follow for the diagnosis, treatment, prevention or management (decision problems, DP) of the specific health condition of a patient.

However, CPGs are usually available in the form of unstructured textual documents written in natural language whose interpretation often hinders their implementation at point of care [2]. Indeed, the typical lacking of structure in textual guidelines and the usual blending of information of several DP in a CPG can make the role of guidelines cumbersome and the work of practitioners unbearable.

One of the successful strategies for this issue is the integration of information technologies and research frameworks in health care environments. Works

reported in the literature follow two main research lines: 1) *model-centric*, which aims to model CPGs with directly computable formalisms (e.g., ontologies [10]), 2) *document-centric*, which aims to transform CPGs into a human interpretable structured format. In the latter large attention has been paid to approaches for manually marking-up the unstructured textual CPGs and transferring the literal content into a pre-defined representation format [9]. However, both research lines are likely to be resource-consuming because demand knowledge on both medical domains and modeling languages and require a lot of human intervention especially for the activity of marking-up the CPGs. A different approach is that of (semi)-automatically processing the content of unstructured textual CPGs and generate a structured representation of it based on pieces of information in natural language present in the text: CPGs are thus represented in a format more rigorous than that unstructured, which practitioners can more easily follow. The problem is often faced by adapting domain-independent frameworks of Information Extraction (IE) to the case of CPGs.

However, adapting IE techniques can be expensive and can suffer for low results accuracy because of the complexity of the CPGs. Instead, ad-hoc approaches to process CPGs are often designed for particular medical domains, specific sections of documents or specific DP (e.g., treatment [7]).

In this paper we present a two-phase framework which aims to extract information on practices and indications of any DP from all sections which compose the textual CPGs. It transforms CPGs in a human-interpretable and structured format, require little human intervention and is designed ad-hoc for CPGs. The framework operates in two consecutive phases. The first one localizes and recognizes relevant pieces of information from the sections composing CPGs through Text Processing techniques. The second one builds a structured representation of the CPG of a specific DP by filling a template structure with the previously extracted pieces of information: practices and indications will be represented as a composition of pieces of information. The paper is so organized. In the next section, after a brief overview of similar techniques, we point out the peculiarities of this work and introduce the proposed two-phase framework. A detailed description of the two phases is provided in the sections 3 and 4 respectively. In Section 5 we explore the application to the domain of Metabolic Syndrome and a quantitative evaluation is reported too. Finally conclusions close this work.

2 Related Works and Contribution

As stated before, extracting relevant information from textual CPGs in order to support their application is an approach investigated in the literature either adapting domain-independent IE techniques or building ad-hoc IE systems. In both cases the extraction is guided by a rule set previously acquired (typically called *extraction patterns*). The mode of obtaining the rule set divides these techniques in two threads: *Automatic Learning*, when the rules are induced from example documents through a learning process [3], and *Knowledge Engineering*

(KE), when the rules are defined by exploiting knowledge on the documents [7,12]. Due to the difficulty to prepare a large training set and the high variability and complexity of the sections of textual CPGs, KE is often preferred to Automatic Learning [1]. A representative paper of the KE approach which is similar to ours has been proposed by Kaiser et al. [7]. In their work the goal was to extract information on clinical treatment processes and use it to support the manual modeling of the guidelines concerning that specific DP. The method outputs representations of the CPGs at different level of refinement by exploiting hand-made heuristics (extraction patterns) defined at phrase-level and discourse-level. However, although this approach allows to structure guidelines even at levels of single activities, the usage of extraction patterns tailored for the clinical treatment make it adequate only for that specific DP (in this case treatment).

Differently, our framework does not limit its application to a specific DP and extracts information from the several sections composing the CPGs. This allows us to take into account an important characteristic of the textual guidelines: a CPG contains practices and indications of any DP disseminated in the several sections of the document. For instance, the practices of any DP are summarized and reported together in a section while they are detailed and extended in another section.

The framework processes original CPGs with Text-Processing techniques [11] in order to extract relevant pieces of information in the several sections (first phase). Next, for each DP of a specific health condition, it builds a structured representation by arranging pieces of information in a template structure with a KE approach which exploits a production system (second phase). In this work a piece of information (afterward, pINF) has to be intended as a textual unit composed of (at least) a sentence. This textual unit expresses a well-defined activity to follow for a specific DP. A practice or an indication of a specific DP is then defined as a set of activities.

3 Extraction of Relevant Pieces of Information

The first phase is performed through natural language processing functionalities which analyze the textual CPGs at both section-level and sentence-level and output pINF and annotations. These annotations are represented in the form of attribute-value pairs and express information concerning the sections and pINF. Analysis proceeds in two steps: the first one (TS1) merely segments CPGs into sections and subsections, and obtains relational properties on them (i.e., ordering among (sub)sections, membership of a subsection to a section), while the second one (TS2) performs a linguistic analysis on the obtained (sub)sections aiming at capturing the semantics of pINF and representing it in the form of annotations.

More precisely, TS1 starts by transforming initial CPGs into semi-structured documents with tagged sections since guidelines are often released as marked-up and irregularly formatted documents (e.g., HTML documents). This is done

integrating the Xerces Parser tool¹. Next, a segmentation on tagged sections is performed to split them into subsections and then to identify sets of pINF. Trivially, a section contains several subsections which, in their turn, contain sets of pINF. TS1 generates also instances of two kinds of annotations: *GROUP* and *VALUE* (see Table 1). *GROUP* annotations represent sections and subsections, and express relational properties on them. *VALUE* annotations rather represent information on sections, subsections and the contained pINF.

For instance, given the section *Scope* containing the subsection *Interventions and Practices considered* of the CPG concerning *Essential Hypertension* below reported obtained after the transformation in a semistructured document, TS1 generates the *GROUP* and *VALUE* annotations illustrated in Table 2.

```
<doc_id> <section_lbl>Scope</section_lbl>
  <field_lbl>Interventions and Practicesconsidered</field_lbl>...
  <strong> Treatment/Management </strong> <ol start="1" type="1">
  <li> Drug therapy <ul type="disc">
    <li> Diuretics (thiazide or loop) </li>
    <li> Beta blockers </li>
    <li> Angiotensin converting enzyme (ACE) inhibitors </li> ... </doc_id>
```

A further segmentation on the sets of pINF (namely, *CONTENT* of the *VALUE*) is performed w.r.t. delimiters of text in order to recognize single pINF and their relational properties. Here three annotations are produced: *LIST*, *ITEM*, *PARAGRAPH* (see Table 1). *LIST* describes the placement of a collection of pINF affiliated to a subsection, while *ITEM* and *PARAGRAPH* annotate pINF respectively contained in a *LIST* and not contained but included in the *CONTENT* attribute of the *VALUE*. In addition *LIST*, *ITEM*, *PARAGRAPH* express relational properties on pieces of information present in each subsection. By following the example above, from the attribute *CONTENT* of *VALUE* the *LIST* and *ITEM* annotations illustrated in Table 2 can be generated.

Once the pINF have been localized and annotated, TS2 can be performed. It resorts to linguistic analysis techniques [11] first to split pINF containing several sentences into single sentences, then to capture their semantics. Linguistic analysis exploits hand-coded controlled dictionaries and grammars and includes tokenization, sentence splitting, stemming, part-of-speech tagging and named-entity recognition to be executed in sequence. Three further annotations are produced: *SENTENCE*, *WORD*, *DOMAIN* which describe respectively i) placement of the sentence contained in the pINF, ii) lexical and morphological features of the words contained in the sentences and iii) domain-specific information on the *SENTENCE* annotations (see Table 1). For instance, from the *ITEM* annotations of the previous example the *SENTENCE* and *WORD* annotations in Table 2 can be generated. Annotations so produced will be exploited by Template Filling method to arrange pINF in a template form, namely the final structured representation of the CPGs.

¹ <http://xerces.apache.org/xerces-j/>

Table 1. Representation of the annotations generated by the first phase

<i>GROUP: [ID, TYPE, CONTENT, PARENT, POSITION]</i>
where ID is a unique identifier of the annotation, TYPE is the tag of the corresponding (sub)section, CONTENT is the tag value, PARENT is the identifier of father GROUP for the subsections, POSITION is a progressive index which univocally represents the section in the CPG (in the case of sections) or the subsection in the section (in the case of subsections).
<i>VALUE: [ID, CONTENT, GROUP]</i>
where CONTENT is the tag value, GROUP is the identifier of the group of affiliation of the VALUE.
<i>LIST: [ID, TYPE, GROUP, POSITION]</i>
where TYPE denotes whether the list is ordered or not, GROUP indicates of the group of affiliation of LIST, POSITION is a progressive index representing the position of the LIST in its GROUP.
<i>ITEM: [ID, CONTENT, LIST, POSITION]</i>
where CONTENT is the textual content of the represented element, LIST indicates the LIST which contains the ITEM, POSITION is a progressive index representing the ITEM in its LIST.
<i>PARAGRAPH: [ID, CONTENT, GROUP, POSITION]</i>
where CONTENT is the textual content of the represented element, GROUP indicates the GROUP which contains the PARAGRAPH, POSITION is the position of PARAGRAPH in its GROUP.
<i>SENTENCE: [ID, CONTENT, PARAGRAPH, ITEM, GROUP, POSITION]</i>
where CONTENT is the textual content, PARAGRAPH, ITEM, GROUP denote the object of the current annotation: only one of them can be instantiated. PARAGRAPH is instantiated when the sentence refers to a PARAGRAPH, ITEM is instantiated when the sentence refers to a ITEM and GROUP is instantiated when the sentence refers to a GROUP and none of the previous. POSITION is a progressive index of the SENTENCE.
<i>WORD: [ID, CONTENT, SENTENCE, CATEGORY, POS, PROPERTIES, POSITION]</i>
where CONTENT is the textual content, SENTENCE denotes the SENTENCE of the current annotation, CATEGORY represents a generalization of the concept expressed by CONTENT (e.g. organ is the category of liver) according to controlled vocabularies, POS denotes the part-of-speech tag of CONTENT, PROPERTIES expresses linguistic/orthographic properties of CONTENT (e.g. lowercase), POSITION is a progressive index of WORD in the corresponding SENTENCE.
<i>DOMAIN: [ID, CONTENT, SENTENCE, CATEGORY]</i>
where CONTENT is the textual content of the corresponding SENTENCE, SENTENCE denotes the SENTENCE of the current annotation, CATEGORY represents a generalization of the concept expressed by CONTENT (e.g. intended users is the category of Allied Health Personnel) according to controlled vocabularies.

Table 2. Annotation examples generated by the first phase

<i>GROUP: [ID1, Section, Scope, null, 1]</i>
<i>GROUP: [ID2, Field, Interventions and Practices Considered, ID1, 1]</i>
<i>VALUE: [ID3, Treatment/Management <ol start="1" type="1"> Drug therapy <ul type="disc"> Diuretics (thiazide or loop) Beta blockers Angiotensin converting enzyme (ACE) inhibitors , ID2]</i>
<i>LIST: [ID4, ordered, ID2, 1]</i>
<i>ITEM: [ID5, Drug therapy, ID4, 1]</i>
<i>LIST: [ID6, unordered, ID2, 1]</i>
<i>ITEM: [ID7, Angiotensin converting enzyme (ACE) inhibitors, ID6, 3]</i>
<i>SENTENCE: [ID13, Drug therapy, null, ID5, null, 1]</i>
<i>SENTENCE: [ID9, Angiotensin converting enzyme (ACE) inhibitors, null, ID7, null, 1]</i>
<i>WORD: [ID8, Drug therapy, ID13, therapeutic procedure, complex POS, null, 1]</i>
<i>WORD: [ID10, Angiotensin converting enzyme, ID9, pharmacological substance, complex POS, null, 1]</i>
<i>WORD: [ID11, ACE, ID9, pharmacological substance acronym, nn, upperCase, 2]</i>
<i>WORD: [ID12, inhibitor, ID9, word, nns, lowerCase, 3]</i>

4 Template Filling for Structuring CPGs

A final CPG has a pre-defined template structure composed of five main slots named as *Disease/ Health Conditions*, *Target Population*, *Benefits*, *Harms*, *Practices*. Each of them, in its turn, can contain nested slots: for instance, the *Practices* slot is composed of as many inner slots as the practices of the corresponding CPG are, while each inner slot contains, in its turn, as many basic slots as the activities of the corresponding practice are. It is thus expected that two different CPGs have different numbers of inner and basic slots albeit they present the same number of main slots.

The filling method constructs the final CPG through a bottom-up strategy which fills basic slots and merges them to return inner slots up to obtain the main slots: initially it acquires information on the pINF (annotations) for a particular DP of a specific health condition, then it infers the content of the slots *Disease/ Health Conditions*, *Target Population*, *Benefits*, *Harms* (which express information required by standard conventions or expected by practitioners). Subsequently, it derives the content of the *Practices* slot by first identifying the sections of the original CPG which potentially contain the single practices and, then, by inferring, for each practice, the content of its slots, namely the activities described in summarized and detailed form.

The filling procedure is based on a production system with forward chaining inferential mechanism [6], which is composed of an inferential engine, facts-list and knowledge base. Facts-list contains two kind of assertions: the previously generated annotations represented in a suitable language and facts deduced during the inferential process which determine the pINF which fill the slots. Deducted facts, in their turn, are divided into *mapping* and *matching* facts. Mapping facts identify the sections which can contain pINF for the slots, while matching ones indicate pINF which can fill the slots. Each mapping or matching fact presents a numerical score based on the annotations of the associated pINF. During the inferential process different matching facts can compete for the same slot and the final assignment is decided on the basis of the higher score value. Knowledge base consists of *if-then* rules whose conditional part is described in terms of facts, while the consequential part specifies particular operations to do on the facts-list. Rules are divided into standard conventions rules (SCr), control rules (Cr), filling rules (Fr). SCr define characteristics that final CPG must have and information expected by practitioners. For instance, one of the SCr rules states that in the final CPGs the summarized description of each practice has to be followed by the detailed description. Cr regulate the activation of Fr by retracting facts which can fire Fr when other facts with higher score have been previously inferred. Finally, Fr can infer matching facts (namely, facts associated to the pINF which can fill the slots) and mapping facts (namely, facts on the sections of the original CPG which can contain potential pINF for the slots).

A concrete example is reported in Table 3 and illustrates the inference of facts for the original CPG reported in Figure 1. The subsections *Interventions and Practices considered* (IPc) and *Major Recommendations* (MR) contain respectively the summarized and detailed description of the practices for

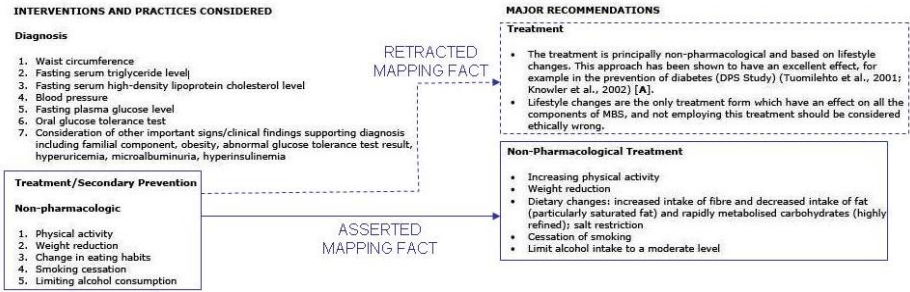


Fig. 1. Example of activation of rules in the knowledge base: the rule retracts a mapping fact previously inferred and asserts another one having higher numerical score

any DP. Suppose that *IPc* and *MR* have been annotated with *GROUP1* and *GROUP2* respectively. The rule in Table 3 derives that the practices for the non-pharmacological treatment DP have to be sought in the subsections *Non-pharmacologic* of *IPc* (annotated with *GROUP1_1*) and *Non-pharmacological Treatment* of *MR* (*GROUP2_2*): more precisely, it retracts the mapping fact (dotted line in Figure 1) on *Non-pharmacologic* of *IPc* and *Treatment* of *MR* (*GROUP2_1*) and infers the new mapping fact (solid line in Figure 1) on the basis of the occurrence of the same DP in the *SENTENCE* annotations (i.e., *Treatment DP*) and the occurrence of the shared terms in *WORD* annotations (e.g., *Non-pharmacologic*).

5 Application to the Metabolic Syndrome CPGs

In order to prove the viability of the proposed framework we applied it to the context of CPGs concerning Metabolic Syndrome. The phase of extraction of relevant pINF exploits the facilities of GATE (General Architecture for Text Engineering) system [4], while the production system used for template filling is developed as an expert system in CLIPS language. Moreover, the vocabularies³ used for linguistic analysis are built considering the dictionaries available

Table 3. Example of Cr used for regulating the activation of Fr on a mapping fact

<p>IF $\text{word_set}(\text{SENTENCE}(\text{GROUP1_1}) \cap \text{word_set}(\text{SENTENCE}(\text{GROUP2_2})) \neq \emptyset$ and $\text{DP}(\text{GROUP1_1}) = \text{DP}(\text{GROUP2_1})$ and $\text{affiliationToSummary}(\text{GROUP1_1})$ and $\text{affiliationToDetail}(\text{GROUP2_2})$ and $\text{any_mapping_exists}(\text{GROUP1_1})$</p> <p>THEN $\text{retract}(\text{any_mapping}(\text{GROUP1_1}))$ and $\text{assert}(\text{mapping}(\text{GROUP1_1}, \text{GROUP2_2}))$</p> <p>FACTS-LIST: $\text{mapping_fact}(\text{GROUP1_1}, \text{GROUP2_1})$, $[\text{SENTENCE1}, \text{Non-pharmacologic}, \text{null}, \text{null}, \text{GROUP1_1}, \text{GROUP1}, 1]$, $[\text{SENTENCE3}, \text{Non-pharmacological Treatment}, \text{null}, \text{null}, \text{GROUP2_2}, \text{GROUP2}, 1]$, $[\text{SENTENCE2}, \text{Treatment}, \text{null}, \text{null}, \text{GROUP2_1}, \text{GROUP2}, 1]$, $[\text{WORD1}, \text{Non-pharmacologic}, \text{SENTENCE1}, \text{GROUP1_1}, \text{word}, \text{adjective}, \text{upperInitial}, 1]$, $[\text{WORD2}, \text{Non-pharmacologic}, \text{SENTENCE3}, \text{word}, \text{adjective}, \text{upperInitial}, 1]$, $[\text{WORD3}, \text{Treatment}, \text{SENTENCE3}, \text{Treatment DP}, \text{noun}, \text{upperInitial}, 2]$, $[\text{WORD4}, \text{Treatment}, \text{SENTENCE2}, \text{Treatment DP}, \text{noun}, \text{upperInitial}, 1]$</p>

Table 4. Experimental results: percentage values of precision and recall

Guideline Title	Decision Problem	#avs	#tfs	#cfs	recall	precision
Osteoporosis in gastrointestinal disease	diagnosis	17	18	17	94	100
	treatment	27	29	22	81	75.8
	management	27	29	22	81	75.8

in Unified Medical Language System (UMLS) specific for Metabolic Syndrome provided by domain experts. The set of initial unstructured guidelines was retrieved by submitting the query “Metabolic Syndrome” to National Guideline Clearinghouse (NGC)² search engine: from the returned CPGs (more than 100) we selected 39 documents which did not present text in tabular structures and which concern at least one of the following decision problems: Diagnosis, Treatment, Management, Prevention, Risk Assessment, Evaluation. A subset of 23 documents was thus used to develop the knowledge base of the production system whose rules were hand-coded by exploiting knowledge of the practitioners on which sections are present within a CPG, which sections are useful for them and standard conventions of CPGs. Knowledge base comprises a total set of 40 rules³ so partitioned: SCr (5), Cr (17), Fr (18). The similar organization of these documents permits us to analyze a small set of them and apply the derived rule set also to others. The remaining 16 documents were processed by the framework which returned 43 templates so distributed: Diagnosis (15), Management (11), Treatment (11), Prevention (3), Screening (1), Evaluation(2). Final CPGs (namely, filled templates) were evaluated according to *Precision*, *Recall*, *macroaveraged Precision* (π^M) and *macroaveraged Recall* (ρ^M) [8]. Recall estimates the number of filled slots w.r.t. the total number of available slots (#avs), while Precision estimates the number of correctly filled slots (#cfs) against all filled slots (#tfs). The values of π^M and ρ^M amount respectively to 79.39% and 80.79% w.r.t the total number of the final CPGs. However, for limitations of space, we report only the most significant CPG (see Table 4).

By analyzing the results of Table 4 it emerges that templates of different DP can present different recall and precision values even when extracted from the same initial guideline (e.g., diagnosis and treatment of *Osteoporosis in gastrointestinal disease*). This can be attributed to the irregular organization of the detailed description of the practices, namely MR section (see Figure 1). For instance, in the case of *Osteoporosis in gastrointestinal disease* (see Figure 2), the detailed description of the Treatment practices is blended in the MR section with the practices of other DPs. In this case, after that the annotations for the sections IPc and MR have been instantiated, the procedure of template filling first searches for the summarized practices of the Treatment DP in IPc, then infers that the Treatment practices have in the MR section a dedicated and subsection within which they are detailed, while that subsection actually does not exist. Hence, it derives that the detailed practices have to be sought in the entire

² <http://www.guidelines.gov/>

³ Downloadable at

<http://www.di.uniba.it/~malerba/software/FELIPE/CPGresources/>

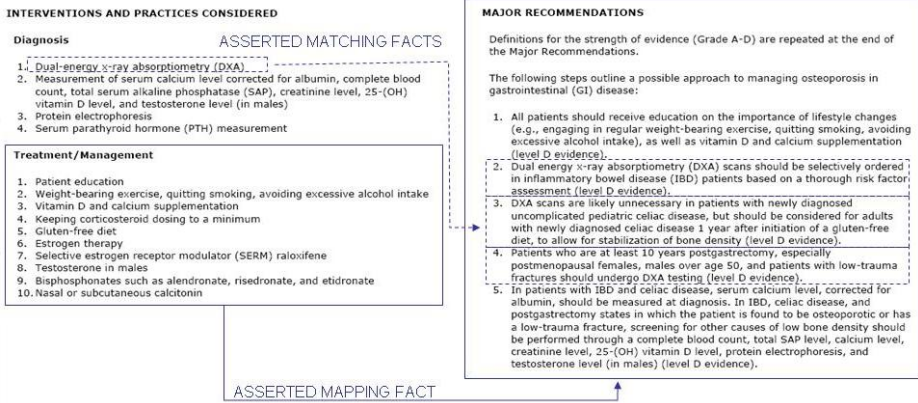


Fig. 2. Summarized and detailed descriptions of Treatment and Diagnosis practices for *Osteoporosis in gastrointestinal disease*

section MR and infers the mapping fact between the GROUP of Treatment of IPc and the GROUP of MR (solid line in Figure 2) which anyway contains also the practices of other DP. These may represent “noisy” practices for the generation process of the Treatment template and may negatively influence the final precision and recall (75.8%, 81% w.r.t. the averages 79.39% and 80.79%).

Another factor which can affect Precision and Recall lies in the domain-specific vocabularies used for the linguistic analysis: exploiting well curated vocabularies can facilitate the instantiation of annotations which better capture the semantics and domain knowledge of pINF (e.g., WORD, DOMAIN annotations in Table 1). For instance, in the case of the Diagnosis template in Figure 2, integrating domain acronyms (i.e., DXA) and hierarchies among background concepts (i.e., x-ray absorptiometry is a Diagnostic procedure) into vocabularies allows to generate WORD and DOMAIN annotations, for the subsection Diagnosis in IPc and for the section MR, which i) better express information of the Diagnosis practices, ii) mitigate the effect of the irregular organization of the MR section and iii) improve the final accuracy of Diagnosis template. A concrete illustration of that is represented by the inference of the matching fact (dotted line) of the practice 1 of the Diagnosis in Figure 2: indeed, its summarized description is extracted by the list item 1 in IPc while that detailed is localized in the items 2,3,4 of the MR section.

6 Conclusions

In this paper we have proposed a computational solution to support the usage and interpretation of CPGs. It based on the extraction of information deemed to be important and the representation of it in a structured mode which practitioners can more easily follow. The approach investigates two particular issues of CPGs: the presence of information (i.e., practices and indications) concerning

different decision problems and the typical lacking of structure of the textual guidelines. The purposeful aspect of this work is the automatic identification of practices relevant for a given problem in the several sections of CPGs and the presentation of these in the structured form where each practice is both synthetically and fully described. The application to the Metabolic Syndrome scenario shows the adaptability of the approach also to real contexts. As future work, we intend to integrate an Automatic Learning technique into the filling procedure to generate more accurate structured representation of the CPGs.

Acknowledgments

This work is partial fulfillment of objective of ATENEO-2009 project “Modelli e metodi computazionali per la scoperta di conoscenza in dati biomedici”.

References

1. Appelt, D.E.: Introduction to information extraction. *AI Communications* 12, 161–172 (1999)
2. Cabana, M.D., Rand, C.S., Powe, N.R., Wu, A.W., Wilson, M.H., Abboud, P.A., Rubin, H.R.: Why don't physicians follow clinical practice guidelines? A framework for improvement. *JAMA* 282(15), 1458–1465 (1999)
3. Califf, M.E., Mooney, R.J.: Relational Learning of Pattern-Match Rules for Information Extraction. In: *Proc. of AAAI/IAAI*, pp. 328–334 (1999)
4. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: Gate: A framework and graphical development environment for robust nlp tools and application. In: *40th Anniversary Meeting of the Association for Computational Linguistics* (2002)
5. Field, M.J., Lohr, K.H. (eds.): *Clinical Practice Guidelines: Directions for a New Program*. National Academy Press, Institute of Medicine (1990)
6. Ignizio, J.P.: *Introduction to Expert Systems: The Development and Implementation of Rule-Based Expert Systems*. McGraw-Hill, Inc., New York (1991)
7. Kaiser, K., Miksch, S.: Modeling Treatment Processes Using Information Extraction. *Advanced Computational Intelligence Paradigms in Healthcare* (1), 189–224 (2007)
8. Sebastiani, F.: Machine learning in automated text categorization. *ACM Comput. Surv.* 34(1), 1–47 (2002)
9. Svatek, V., Ruzicka, M.: Step-by-step formalisation of medical guideline content. *Int. Journal of Medical Informatics* 70(2-3), 329–335 (2003)
10. Wang, D., Peleg, M., Tu, S.W., Boxwala, A.A., Ogunyemi, O., Zeng, Q.T., Greenes, R.A., Patel, V.L., Shortliffe, E.H.: Design and implementation of the GLIF3 guideline execution engine. *Journal of Biomedical Informatics* 37(5), 305–318 (2004)
11. Weiss, S., Indurkha, N., Zhang, T., Damerou, F.: *Text Mining: Predictive Methods for Analyzing Unstructured Information*. Springer, Heidelberg (2004)
12. Yangarber, R., Grishman, R.: NYU: description of the Proteus/ PET system as used for MUC-7 ST. In: *Proc. of the 7th MUC*. Morgan Kaufmann, San Francisco (1998)

RaJoLink: A Method for Finding Seeds of Future Discoveries in Nowadays Literature

Tanja Urbančič^{1,2}, Ingrid Petrič¹, and Bojan Cestnik^{2,3}

¹ University of Nova Gorica, Vipavska 13, 5000 Nova Gorica, Slovenia

² Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

³ Temida, d.o.o., Dunajska 51, 1000 Ljubljana, Slovenia

tanja.urbancic@p-ng.si, ingrid.petric@p-ng.si,
bojan.cestnik@temida.si

Abstract. In this article we present a study which demonstrates the ability of the method RaJoLink to uncover candidate hypotheses for future discoveries from rare terms in existing literature. The method is inspired by Swanson's ABC model approach to finding hidden relations from a set of articles in a given domain. The main novelty is in a semi-automated way of suggesting which relations might have more potential for new discoveries and are therefore good candidates for further investigations. In our previous articles we reported on a successful application of the method RaJoLink in the autism domain. To support the evaluation of the method with a well-known example from the literature, we applied it to the migraine domain, aiming at reproducing Swanson's finding of magnesium deficiency as a possible cause of migraine. Only literature which was available at the time of the Swanson's experiment was used in our test. As described in this study, in addition to actually uncovering magnesium as a candidate for formulating the hypothesis, RaJoLink pointed also to *interferon*, *interleukin* and *tumor necrosis factor* as candidates for potential discoveries connecting them with migraine. These connections were not published in the titles contemporary to the ones used in the experiment, but have been recently reported in several scientific articles. This confirms the ability of the RaJoLink method to uncover seeds of future discoveries in existing literature by using rare terms as a beacon.

Keywords: Literature mining, knowledge discovery, rare terms, migraine.

1 Introduction

As in other fields, also in research we face the problem of an overwhelming flow of information. This includes textual information presented in numerous professional articles, many of them available on-line through largely increasing bibliographic databases which are of immense help, but have, on the other hand, made literature-based knowledge discovery a very time-consuming and laborious task. A good example is MEDLINE (the United States National Library of Medicine's bibliographical database) which contains more than 16 million citations from 1949 to the present and increases for more than 2.000 complete references daily [1]. In these circumstances, software tools providing support to researchers in their literature-based discovery are highly desirable.

Another feature that sometimes makes discovery processes difficult is a prevailing overspecialization of scientists and other professionals. This can be traced even in publications and cross-references that tend to be closed in specific professional communities. Many problems that this society of today has to solve are extremely complex and solutions can't be found without cooperation across the boundaries of different disciplines. Again, software tools can support researchers in crossing these boundaries, helping them in putting dispersed pieces of knowledge together into a bigger, coherent picture.

Scientific discovery usually encompasses also different types of cognitive processes, such as finding analogies or selecting and combining ideas that originate in different contexts or disciplines. When these usually separated ideas or fragments of knowledge come together in a creative way, they may lead to original, sometimes surprising findings. If there is enough evidence for such findings to be interesting and useful, they can be viewed as hypotheses which can, if proved with methods approved by scientific community, represent new discoveries. These kinds of context-crossing »associations« are called bisociations [2] and are often needed for creative, innovative discoveries. This is very important also in complex interdisciplinary settings. However, the information that is related across different contexts is difficult to identify with the conventional associative approach.

Already in 1986 it was shown by Swanson that bibliographic databases such as MEDLINE could serve as a rich source of hidden relations between concepts [3]. He proposed a simple, but extremely powerful method for finding hypotheses which span over previously disjoint sets of literature. To find out whether phenomenon a is associated with c although there is no direct evidence for this in the literature, he suggests finding intermediate concepts b which are connected with a in some articles and with c in some others. Putting these connections together and looking at their meaning sometimes reveals interesting connections between a and c , worth further investigations.

In one of his studies, Swanson investigated if magnesium deficiency can cause migraine headaches [4]. In this case, migraine played the role of c and magnesium played the role of a . He found more than 60 pairs of articles (consisting of one article from the literature about migraine and one from the literature about magnesium) connecting a with c via terms b . Closer inspection showed that 11 pairs of documents were, when put together, suggestive and supportive for a hypothesis that magnesium deficiency may cause migraine headaches [5].

Several researchers have followed this line of research, among them Weeber et al. [6], Srinivasan et al. [7], Hristovski et al. [8], Yetisgen-Yildiz and Pratt [9]. Our research was also inspired by the Swanson's idea. In particular, the question we wanted to answer was: Being interested in phenomenon c , how do we find an agent a as a promising candidate for generating hypotheses about c being connected with a ? In other words, being interested in migraine, why did Swanson focus on magnesium and not on something else? Swanson was not very specific in this respect and does not provide any systematic guidance for this particular part of discovery. To support this process, we developed a method RaJoLink which suggests candidates in a semi-automated way. As the main novelty, it uses rare terms from the literature about c for guidance. The RaJoLink method was introduced and illustrated with its application to the autism domain in Urbančič et al. [10]. It was based on ideas and earlier work published in Petrič et al. [11]. Later it was generalized and described in more detail in Petrič et al. [12] where

also a software tool implementing the RaJoLink method was presented. RaJoLink was tested in the autism domain, a complex field of investigations which are carried out in different medical disciplines, but there is an evident lack of a more coherent understanding connecting their findings [13], [14]. With the aid of RaJoLink, we discovered a relation between autism and calcineurin which hasn't been published before and which has been evaluated by a medical expert as relevant for better understanding of autism. Similarly, we found NF-kappaB, a transcriptional factor that according to the analysis of a medical expert represents one possible point of convergence between "oxidative stress" and "immunological disorder" paradigm in autism. A more detailed report on autism-related issues of the study can be found in [10].

To get a thorough medical confirmation of the found hypotheses in the autism domain, additional expert investigations will be needed. Since this may take some time, we wanted to check the capabilities of RaJoLink also in another way, more suitable for an immediate evaluation. To answer today if RaJoLink has the potential to reveal future discoveries (without waiting these discoveries to actually be confirmed), we decided to make an experiment as it was done in the past: Having all evidence that was available at a certain time in the past, can RaJoLink point to discoveries that were not known at that chosen moment, but were confirmed some years later? To answer to this question we carried out an experiment in the migraine domain, simulating the Swanson's situation. We investigated two issues: having information about papers that were available in 1988, (1) does RaJoLink find magnesium, and (2), does RaJoLink maybe find something else that has later proved to be connected with migraine.

In the next section we give a brief overview of Swanson's model of knowledge discovery and related work. In section 3 we briefly present the method RaJoLink. Section 4 reports on the application of RaJoLink to the migraine domain. Results are commented in more detail in Section 5. In Section 6, we summarize the most important findings and suggest some further work.

2 Swanson's Model of Literature-Based Knowledge Discovery

Swanson regards scientific articles as clusters of somewhat independent sets or literatures, where common matters are considered within each set [5]. Such distinct unrelated literatures could be linked to each other by arguments that they treat. Consequently, if two literatures can be logically related by arguments that each of them addresses the unobserved connections between them represent potential sources of new knowledge. For instance, if a literature *A* (i.e. a set of all available records about *a* in the database serving as a source of data) reports about a term *a* being in association with a term *b*, and another literature *C* associates a term *c* with a term *b*, we can thus assume the literature *B* as an unintended implicit potential connection between the literatures *A* and *C*.

Swanson investigated the process of finding implicit connections between disjoint literatures using the titles of articles and their MeSH descriptors from MEDLINE records. MeSH stands for the Medical Subject Headings thesaurus which is produced by the U.S. National Library of Medicine [15].

For literature-based discovery, Smalheiser and Swanson [16] designed the ARROWSMITH system based on MEDLINE search. Their major focus was on the hypothesis testing approach [17], which Weeber and colleagues [6] defined as a

closed discovery process, where both a and c have to be specified at the start of the process. On the other hand, an *open* discovery process is characterized by the absence of requirement for advance specification of target concepts. If we are investigating a subject denoted with the term c , the open discovery starts with having only the term c and the corresponding set of articles in which term c appears (called also literature C), without knowing the target term a , which is discovered later as a result of this process. By considering unconnected sets of articles Swanson managed to make several surprising discoveries. In one of them, he discovered that patients with Raynaud's syndrome might benefit from consuming dietary fish oil [3].

Similarly, while studying two separate literatures, the literature on migraine headache and the articles on magnesium, Swanson found implicit connections that were unnoticed at the outset of his research [5]. Swanson noticed the possible relationship between the disjoint literatures on migraines and on magnesium by the intermediate literature. In fact, some linking terms, such as *calcium channel blockers* and *spreading cortical depression* appeared frequently in the titles of both the migraine literature and the magnesium literature. However, prior to the Swanson's discovery, a few researchers (e.g., [18]) had given attention to a direct magnesium-migraine connection, but laboratory and clinical investigations started numerous only after the publication of the Swanson's convincing evidence. "Migraine and magnesium" example has become a golden standard in the literature mining field and has been used as a benchmark in several studies, including [6], [19], [20] and [21].

3 RaJoLink Method

Swanson stated that in open discovery processes success depends entirely on the knowledge and ingenuity of the searcher [5]. The aim of the RaJoLink method is to reduce the search space, thus making the task easier for the searcher. The method consists of the following steps: (1) Identification of n interesting rare terms r_1, r_2, \dots, r_n in literature about c . (2) Search for a joint term a in an intersection of available literatures about these rare terms. (3) Search for m linking terms b_1, b_2, \dots, b_m such that for each b_i there exists a pair of articles, one from literature A and one from literature C , both mentioning b_i . (4) Expert evaluation in which statements about a and b_i and statements about b_i and c from the obtained articles are put together to see whether they, being put together, indicate new hypotheses connecting a and c . The method is described in more detail in [12]. It is named by its most distinctive elements: *Rare* terms, *Joint* terms and *Linking* terms. Accordingly, we call the first three steps *Ra*, *Jo* and *Link*.

Note that rare terms don't have the same property as linking terms (b), although rare terms individually co-occur with the literature A and the literature C , but not with A and C jointly. In fact, when testing hypotheses by searching for meaningful linking terms b between the literatures A and C , the focus should be on the most frequent terms, while the rarest ones are interesting in the hypotheses generation phase. By focusing on rare terms the system identifies the candidates that are most likely to lead towards meaningful still unpublished relations. This way, the system automatically produces intermediate results. Search space is further reduced by human choices of rare terms and joint terms a . In addition, human involvement in these steps assures that search process concentrates on those parts of the search space that are interesting

and meaningful for a subject expert. One or more selected joint terms a are then considered in the last step (i.e. hypothesis testing) for detection of implicit links with the problem domain denoted by term c . Based on these strategies, RaJoLink is designed to make the expert's involvement easy and efficient.

The search for the linking terms b in the closed discovery process is combinatorially complex [5], [22]. The process is equivalent to Swanson's hypothesis testing in the closed discovery approach [5], where the search for linking terms consists of looking for terms b that can be found in separate sets of records, namely in the literature A as well as in the literature C . Nevertheless, our closed discovery approach contains an unique aspect in comparison to the literature-based discovery investigated by others. It is the focusing on neighbouring documents in the documents' similarity graphs, which are defined over the combined datasets consisting of literatures A and C . Such visual analysis can show direction to the previously unseen relations, which provide new knowledge. Here, RaJoLink gives priority to the terms b from two records, one from the set of records A and one from the set of records C , so that records in which b appears are close with respect to the similarity measure. One record in the database corresponds to one title or one abstract or one full text of an article. A relationship between two records, one from the set of records A and one from the set of records C was graphically illustrated by similarity graphs in [10] and [12]. The final choice of linking terms is, again, provided by a domain expert. Expert's explicit involvement in the process enables more focused and faster obtainment of results which are meaningful and interesting for further investigation. In the key steps of the process, RaJoLink supports the expert by listing term candidates sorted by frequencies which turned out to be a useful estimate for their potential for knowledge discovery.

4 Applying RaJoLink to the Migraine Domain

To evaluate the performance of the RaJoLink system we replicated the early Swanson's migraine-magnesium experiment that represents a benchmark problem in the literature-based discovery. Moreover, besides the magnesium, we estimated also the rest of the potential discoveries that the RaJoLink system generated in the open discovery process in the migraine domain. This way we performed a more complete evaluation of the RaJoLink method, which gives additional information about the method's capability of detecting interesting terms in large text collections.

Like Swanson in his original study of the migraine literature [4] we used titles as input for our literature-based discovery. However, we excluded from the analysis the few articles that mentioned both migraine and magnesium in their title although they were published before the Swanson's discovery of the migraine-magnesium connection. This way we proved the originality of the magnesium discovery in our migraine experiment. With the automatic support of the RaJoLink system, we performed the experiment on the entire set of the MEDLINE titles of articles that were published before 1988 and that we retrieved with the search of the phrase: *migraine NOT magnesium*. As a result we got 6127 titles of the MEDLINE articles that we analysed according to the RaJoLink method to identify interesting discoveries in connection with migraine. With the RaJoLink analysis of the 6127 titles of the MEDLINE articles about migraine that were published before 1988, we identified 4663 different terms, among which 2442 terms with their document frequency equal to 1. In our

experiments we considered only the rarest terms (with their document frequency equal to 1) from the *A*, *C* and *D* category of the 2008 MeSH tree structure. Thus we chose meaningful rare terms belonging to categories *Anatomy*, *Diseases*, and *Chemicals and Drugs*, as in our experimental case the words:

- *hypocalcemia* and *toxemia* from the category *C – Diseases* and *phosphorus* from the category *D – Chemicals and Drugs* that led to *magnesium* in the step Jo.
- *chromatid* from the category *A – Anatomy*, *lymphoma* from the *C – Diseases* and *tuberculin* from the category *D – Chemicals and Drugs* that led to *interferon*.

The evaluation procedure used in this experiment differs from the canonical RaJoLink method in that a human expert was not involved. However, our systematic investigation showed that quite a number of triples of different rare terms resulted in finding magnesium in the intersection of their literatures.

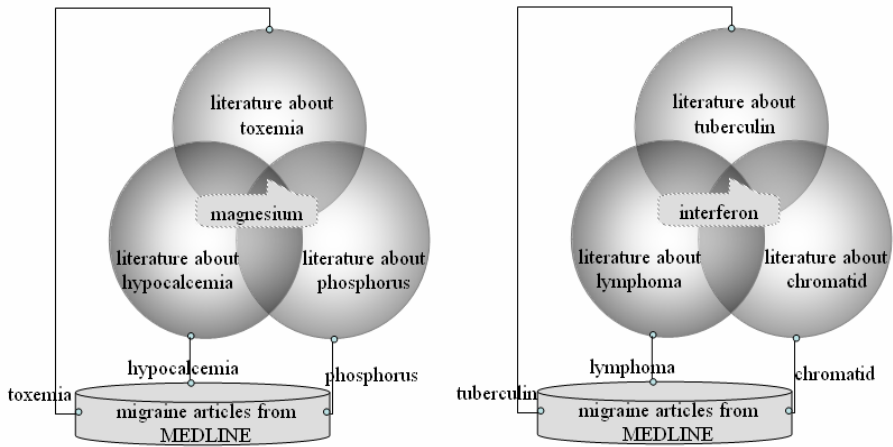


Fig. 1. Magnesium (left) and interferon (right) were found with RaJoLink as joint terms in the intersection of literatures about terms found as rare in the context of migraine

In the open discovery process of the migraine experiment we obtained another three important discoveries related to migraine, besides magnesium (Figure 1):

- *interferon* that denotes proteins produced by the cells of the immune system,
- *interleukin* that is a type of signalling molecules called cytokines,
- *tnf* that stands for tumor necrosis factor.

At the time of Swanason’s experiment, *interferon*, *interleukin* and *tnf* were not connected with migraine in MEDLINE at all. The connections appeared later. *Interleukin* and *tnf* were first associated with migraine in articles accessible through MEDLINE in 1991 and 1990, respectively [23], [24], [25], [26]. The connection between *interferon* and migraine was according to articles on MEDLINE established even later, starting in 1995 [27], [28], [29]. The discoveries of the interferon’s, interleukin’s and *tnf*’s role and their underlying mechanisms involved in the migraine headaches have

been recently reported in 16 scientific articles about interferon, in 39 studies of interleukin and in 29 scientific studies of tumour necrosis factor (Source: MEDLINE search, December 2008).

5 Category-Based Filtering

To support the domain expert during the choices of the potentially relevant rare terms in the open discovery process, we performed the analysis of the migraine-magnesium experimental results by observing which MeSH categories are significant for new discoveries. In the RaJoLink method we use MeSH to reduce the set of possible candidates in both, *rare* and *joint* phase. By estimating the number of rare terms within a particular MeSH category that led to the discovery of magnesium as a joint term, we obtained the density distribution of MeSH categories. Based on the number of rare terms that resulted as relevant for the hypotheses generation, we can observe the influence of a particular MeSH category to the achievement of results.

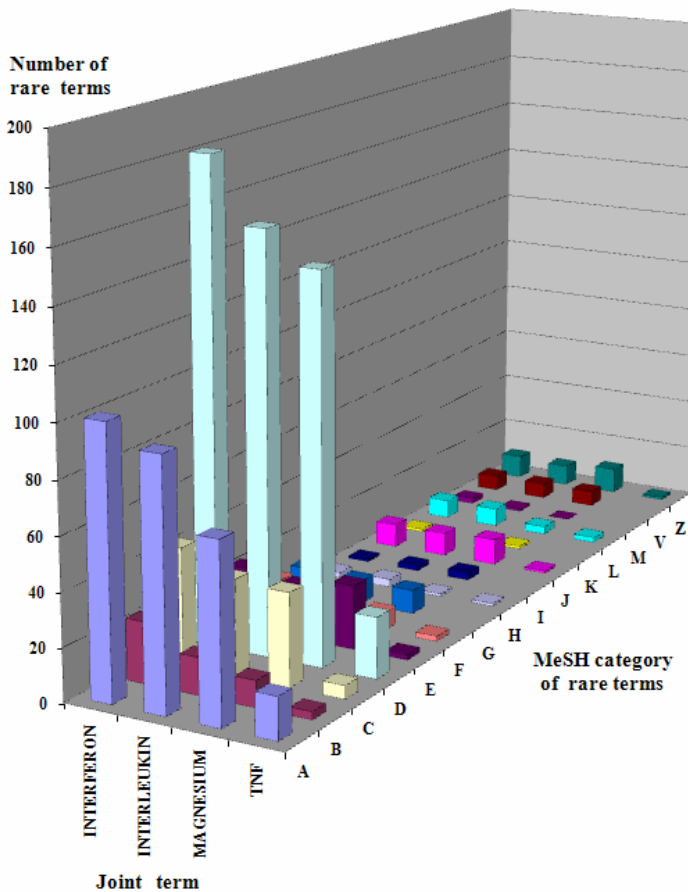


Fig. 2. The MeSH category rankings for the results in the migraine domain

Figure 2 presents the results for the four joint terms, namely the term *interferon*, *interleukin*, *magnesium*, and *tnf* that we examined in more detail during the open discovery process of the migraine experiment. For the sake of this experiment we automatically retrieved from the MEDLINE database 1000 titles of articles for each of the selected rare terms identified in the literature about migraine. Again, the PubMed search and text analysis was performed on the articles published before 1988 with the support of the RaJoLink system. We calculated the document frequency statistics for all potentially relevant rare terms. The MeSH category rankings were computed separately for each of the selected joint terms: *interferon*, *interleukin*, *magnesium*, and *tnf*. As shown in Figure 2, the rare terms from the MeSH category D – Chemicals and Drugs were the most prominent in the migraine literature because the largest number of terms that led to the discovery of the four joint terms was from the category of Chemicals and Drugs. The second category of terms that frequently led to the four discoveries was the MeSH category A – Anatomy. The third significant MeSH category for each of the analyzed joint terms was the category C – Diseases.

These results show that the migraine–magnesium hypothesis is very plausible if we take more rare terms from the category D. Besides, this MeSH category was found as the most prominent in our migraine experiment also for the discoveries of interferon, interleukin and *tnf*.

6 Conclusions

RaJoLink is a literature mining method for uncovering hidden relations from a set of articles in a given biomedical domain. When the newly discovered relations are interesting from a medical point of view and can be verified by medical experts, they can contribute to better understanding of diseases. In this article we demonstrated the ability of the RaJoLink method to rediscover migraine–magnesium relation in a setting that simulated Swanson’s well-known experiment as a benchmark in the literature mining. Moreover, RaJoLink was not only able to repeat Swanson’s findings, but also pointed out additional connections between migraine and interleukin, tumour necrosis factor and interferon which were not known at the time of Swanson’s experiment, but were published in the subsequent research papers. This demonstrates additional strength of open discovery support provided by RaJoLink and adds a more objective validation to the subjective validation of medical experts that find it a useful tool in their knowledge discovery process.

The article also contributes with an investigation about the role of MeSH categorisation in identification of target terms. The text mining process normally identifies a large number of rare terms. This leads to the question of how to determine a subset of terms, such that they would form a valid statistical criterion for a selection relevant to the knowledge domain. This will also be one of the crucial issues investigated based on presented findings in our further work.

Acknowledgments. This work was partially supported by the project Knowledge Technologies (grant P2-0103) funded by the Slovene National Research Agency, and the EU project FP7-211898 BISON.

References

- [1] PubMed. Overview, <http://www.ncbi.nlm.nih.gov/> (accessed, September 2008)
- [2] Koestler, A.: *The act of creation*. MacMillan Company, New York (1964)
- [3] Swanson, D.R.: Undiscovered public knowledge. *Libr Q* 56(2), 103–118 (1986)
- [4] Swanson, D.R.: Migraine and magnesium: Eleven neglected connections. *Perspectives in Biology and Medicine* 31, 526–557 (1988)
- [5] Swanson, D.R.: Medical literature as a potential source of new knowledge. *Bull. Med. Libr. Assoc.* 78(1), 29–37 (1990)
- [6] Weeber, M., Vos, R., Klein, H., de Jong-van den Berg, L.T.W.: Using concepts in literature-based discovery: Simulating Swanson's Raynaud–fish oil and migraine–magnesium discoveries. *J. Am. Soc. Inf. Sci. Tech.* 52(7), 548–557 (2001)
- [7] Srinivasan, P., Libbus, B.: Mining MEDLINE for implicit links between dietary substances and diseases. *Bioinformatics* 20 (suppl.1), i290–i296 (2004)
- [8] Hristovski, D., Peterlin, B., Mitchell, J.A., Humphrey, S.M.: Using literature-based discovery to identify disease candidate genes. *Int. J. Med. Inform.* 74(2-4), 289–298 (2005)
- [9] Yetisgen-Yildiz, M., Pratt, W.: Using statistical and knowledge-based approaches for literature-based discovery. *J. Biomed. Inform.* 39(6), 600–611 (2006)
- [10] Urbančič, T., Petrič, I., Cestnik, B., Macedoni-Lukšič, M.: Literature mining: towards better understanding of autism. In: Bellazzi, R., Abu-Hanna, A., Hunter, J. (eds.) *AIME 2007. LNCS (LNAI)*, vol. 4594, pp. 217–226. Springer, Heidelberg (2007)
- [11] Petrič, I., Urbančič, T., Cestnik, B.: Discovering hidden knowledge from biomedical literature. *Informatica* 31(1), 15–20 (2007)
- [12] Petrič, I., Urbančič, T., Cestnik, B., Macedoni-Lukšič, M.: Literature mining method RaJoLink for uncovering relations between biomedical concepts. *Journal of Biomedical Informatics* 42, 219–227 (2009)
- [13] Belmonte, M.K., Allen, G., Beckel-Mitchener, A., Boulanger, L.M., Carper, R.A., Webb, S.J.: Autism and abnormal development of brain connectivity. *J. Neurosci.* 24(42), 9228–9231 (2004)
- [14] Persico, A.M., Bourgeron, T.: Searching for ways out of autism maze: genetic, epigenetic and environmental clues. *Trends in neurosciences* 29(7) (July 2006)
- [15] Nelson, S.J., Johnston, D., Humphreys, B.L.: Relationships in Medical Subject Headings. In: Bean, C.A., Green, R. (eds.) *Relationships in the organization of knowledge*, pp. 171–184. Kluwer Academic Publishers, New York (2001)
- [16] Smalheiser, N.R., Swanson, D.R.: Using ARROWSMITH: a computer-assisted approach to formulating and assessing scientific hypotheses. *Comput. Methods Programs Biomed.* 57(3), 149–153 (1998)
- [17] Swanson, D.R., Smalheiser, N.R., Torvik, V.I.: Ranking indirect connections in literature-based discovery: The role of Medical Subject Headings (MeSH). *J. Am. Soc. Inf. Sci. Tec.* 57(11), 1427–1439 (2006)
- [18] Altura, B.M.: Calcium antagonist properties of magnesium: implications for antimigraine actions. *Magnesium* 4(4), 169–175 (1985)
- [19] Blake, C., Pratt, W.: Automatically Identifying Candidate Treatments from Existing Medical Literature. In: *Proceedings of AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*, Stanford, CA (2002)
- [20] Lindsay, R.K., Gordon, M.D.: Literature-based discovery by lexical statistics. *Journal of the American Society for Information Science and Technology* 50(7), 574–587 (1999)
- [21] Srinivasan, P.: Text Mining: Generating Hypotheses from MEDLINE. *Journal of the American Society for Information Science and Technology* 55(5), 396–413 (2004)

- [22] Hearst, M.A.: Untangling text data mining. In: Dale, R. (ed.) *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 3–10. Morgan Kaufmann Publishers, San Francisco (1999)
- [23] Covelli, V., Munno, I., Pellegrino, N.M., Di Venere, A., Jirillo, E., Buscaino, G.A.: Exaggerated spontaneous release of tumor necrosis factor-alpha/cachectin in patients with migraine without aura. *Acta Neurol. (Napoli)* 12(4), 257–263 (1990)
- [24] Covelli, V., Munno, I., Pellegrino, N.M., Altamura, M., Decandia, P., Marcuccio, C., Di Venere, A., Jirillo, E.: Are TNF-alpha and IL-1 beta relevant in the pathogenesis of migraine without aura? *Acta Neurol. (Napoli)* 13(2), 205–211 (1991)
- [25] Shimomura, T., Araga, S., Esumi, E., Takahashi, K.: Decreased serum interleukin-2 level in patients with chronic headache. *Headache* 31(5), 310–313 (1991)
- [26] van Hilten, J.J., Ferrari, M.D., Van der Meer, J.W., Gijsman, H.J., Looij Jr., B.J.: Plasma interleukin-1, tumour necrosis factor and hypothalamic-pituitary-adrenal axis responses during migraine attacks. *Cephalalgia* 11(2), 65–67 (1991)
- [27] Detry-Morel, M., Boschi, A., Gehenot, M., Geubel, A.: Bilateral transient visual obscurations with headaches during alpha-II interferon therapy: a case report. *Eur. J. Ophthalmol.* 5(4), 271–274 (1995)
- [28] Gunderson, C.H.: The impact of new pharmaceutical agents on the cost of neurologic care. *Neurology* 45(3 Pt. 1), 569–572 (1995)
- [29] Gidal, B.E., Wagner, M.L., Privitera, M.D., Dalmady-Israel, C., Crismon, M.L., Fagan, S.C., Graves, N.M.: Current developments in neurology, Part I: Advances in the pharmacotherapy of headache, epilepsy, and multiple sclerosis. *Ann Pharmacother* 30(11), 1272–1276 (1996)

Automatic Generation of P2P Mappings between Sources Schemas

Karima Toumani¹, H elene Jaudoin², and Michel Schneider¹

¹ LIMOS, Blaise Pascal University
{ktoumani,michel.schneider}@isima.fr
² IRISA-ENSSAT
jaudoin@enssat.fr

Abstract. This paper deals with the problem of automatic generation of mappings between data sources schemas, by exploiting existing centralized mappings between sources schemas and a global ontology. We formalize this problem in the settings of description logics and we show that it can be reduced to a problem of rewriting queries using views. We identify two subproblems: the first one is equivalent to a well known problem of computing maximally contained rewritings while the second problem constitutes a new instance of the query rewriting problem in which the goal is to compute minimal rewritings that contain a given query. We distinguish two cases to solve this latter problem: (i) for languages closed under negation, the problem is reduced to the classic problem of rewriting queries using views, and (ii) for languages with the property of structural subsumption, a technique based on hypergraphs is proposed to solve it.

Keywords: Schema mappings, description logics, rewriting queries, hypergraphs.

1 Introduction and Motivation

The growing number of data sources distributed over networks and the relevant exploitation of such data, often heterogeneous, pose the problem of information integration. The use of this information has become a major concern in many areas in the industry, agriculture and commerce [1]. The underlying problems of sharing and integrating information has interested the research community in databases over the two past decades and continue today to be the subject of active investigations [11,10,11,12]. Work in this domain led to the development of techniques and tools to allow an efficient and a transparent access to multiple heterogeneous, distributed and autonomous sources. Two major classes of integration systems were defined : Mediation systems based on the paradigm mediator/wrapper [10] and peer-to-peer systems (P2P) [9]. In the first type of architecture, a central component, called *mediator*, acts as an interface between users and data sources. The mediator is composed of an *global schema* which provides a unified view of data sources. The queries are formulated over a mediation

schema (global schema) and are rewritten using the terms of sources schemas. The complexity of mediation systems increases with respect to the number and the capacities of data sources. Although these systems are effective for applications with few data sources, they are poorly adapted to the new context of integration raised by the web, because they are based on a single global schema. The peer-to-peer systems are composed of a set of autonomous data sources (peers) such that each peer is associated with a schema that represents its domain of interest. Each peer's schema constitutes an entry point into the P2P system. In other words, queries are expressed on the local schema of the peer. The resolution of queries in mediation or P2P systems requires the existence of semantic mappings between global schema and data sources schemas in the mediation approach and between peers schemas in the peer to peer approach. From these mappings, the mediator, respectively the peer, analyzes queries and reformulates them in sub-queries executable by sources or peers.

In this article, we address the problem of automatic generation of semantic mappings between schemas of independent data sources. We call them *P2P* mappings compared to centralized mappings stored in the mediator. The main idea is to exploit existing centralized mappings for inferring P2P mappings. Indeed, various efforts have helped develop centralized mappings for example: (i) between data sources schemas and a global schema in the context of mediation systems or (ii) between web applications and a global ontology of the domain (i.e, ontology that defines the concepts of reference in a particular area). Regarding the latter case, the current development of the web, and particularly the Semantic Web, led to the development of more and more applications that are based on domain global ontologies. For example, the standard **ISCO88** gives a classification of professions while the consortium **RosettaNet** provides standards for trade. Our aim is at exploiting such centralized mappings in order to convert them into P2P mappings between data sources. For example, P2P mappings can be useful to process the distributed queries in P2P systems like SomeWhere [2] and PIAZZA [8,9].

In this paper, we investigate the problem of discovering P2P mappings in the setting of description logics [3]. These logics constitute an important knowledge representation and reasoning formalism. They have been used in different application areas and constitute an important language for the semantic web (e.g.; OWL)¹ of the W3C(World Wide Web Consortium)². In the nutshell, the main contributions of our paper are the following. We formalise the problem of finding P2P mappings from a centralized set of mappings using description logics. Then we propose a technique for automatically generating such mappings. For this purpose, we show that the problem of discovering mappings is reduced to the problem of rewriting queries using views. Specifically, we identify two sub-problems: (i) the first one is equivalent to a well known problem of rewriting queries using views, where we search the maximally contained rewritings of queries [7,4,11], and (ii) the second one is to find minimal rewritings

¹ <http://www.w3.org/2004/OWL/>

² <http://www.w3.org/>

which contain a query. We show that the second problem constitutes a new way of rewriting queries problem and we distinguish two cases to solve it. For languages closed under negation, the problem is reduced to the classic problem of rewriting queries using views and for languages with the property of structural subsumption, a technique based on hypergraphs is proposed to solve it.

The paper is organized as follows. Section 2 introduces the basic concepts of description logics. Section 3 shows how the mapping generation problem can be reduced to the problem of rewriting queries using views. Two kind of rewriting problems, namely M_{max} and M_{min} are defined. Solving the problem M_{min} is detailed in the section 4. Finally, section 5 concludes the paper.

2 Preliminaries

In this section, first we give the basic definitions in the description logics, then we describe the notion of rewriting queries using views which is the core operation in our framework.

2.1 Description Logics: An Overview

Description logics (DLs) [3] are a family of knowledge representation formalisms designed for representing and reasoning about terminological knowledge. In DLs, an application domain is represented in terms of *concepts* (unary predicates) denoting sets of individuals, and *roles* (binary predicates) denoting binary relations between individuals. For example, using the conjunction constructor \sqcap and the qualified existential quantification $\exists R.C$, where R is a role and C is a concept, we can describe the concept *Parent* as follows : $Human \sqcap \exists hasChild.Human$. This concept denotes the set of individuals in the domain who are human and have at least one child.

There are different description logics defined by the set of the constructors they allow. For example, figure 1 give the constructors of two description logics used in our framework : \mathcal{FL}_0 and \mathcal{ALN} . \mathcal{FL}_0 is a simple logic that contains the universal concept (noted \top), the conjunction of concepts (\sqcap) and ($\forall R.C$). The language \mathcal{ALN} is more expressive because it contains in addition to the constructors of \mathcal{FL}_0 , the inconsistent concept (noted \perp), the negation of atomic concepts (i.e., descriptions formed only by a concept name) and cardinalities constraints on the roles ($\geq n R$ and $\leq n R$, where n is a positive integer and R is a role name).

The semantics of a concept description is defined by the notion of interpretation. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty set $\Delta^{\mathcal{I}}$, the domain of the interpretation, and an interpretation function $\cdot^{\mathcal{I}}$ that maps each concept name A to a subset of $\Delta^{\mathcal{I}}$ and each role name R to a binary relation $R^{\mathcal{I}}$, subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. For example, any interpretation of a concept \mathcal{FL}_0 (resp. \mathcal{ALN}) must respect the semantics of this language constructors as given in figure 1.

The notions of satisfiability, subsumption and equivalence between two concepts are defined as follows:

Semantic	\mathcal{FL}_0	\mathcal{ALN}
$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$	X	X
$\perp^{\mathcal{I}} = \emptyset$		X
$(\neg A)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$		X
$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$	X	X
$(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y : (x, y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$	X	X
$(\leq nR)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid (x, y) \in R^{\mathcal{I}}\} \leq n\}$		X
$(\geq nR)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid (x, y) \in R^{\mathcal{I}}\} \geq n\}$		X

Fig. 1. Syntax and Semantic of the concepts constructors for description logics \mathcal{FL}_0 and \mathcal{ALN} , where n denotes a positive integer, A an atomic concept, C and D concepts, R a role and the symbol $\#$ denotes the cardinality of a set

- A concept C is satisfiable iff there is an interpretation I such as $C^I \neq \emptyset$. We say then that I is a valid interpretation or a model for C . The concept C is said inconsistent, noted $C \equiv \perp$ iff C does not admit a model.
- The concept C is subsumed by the concept D , noted $C \sqsubseteq D$, iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}} \forall I$.
- The concept C is equivalent to the concept D , noted $C \equiv D$, iff $C \sqsubseteq D$ and $D \sqsubseteq C$.

The description logics allow to describe an application domain at intentional level using a terminology or Tbox (i.e., a schema or an ontology). Let C be a name of concept and D a concept description. $C \equiv D$ (resp., $C \sqsubseteq D$) is a terminological axiom called *definition* (resp., *primitive specification*). A concept C occurring in the left-hand side of a definition (resp, of a primitive specification) is called *defined concept* (resp, *primitive concepts*). A terminology \mathcal{T} is a finite set of terminological axioms such that no concept name appears more than once in the left-hand side of an terminological axiom.

In this paper, we consider that the terminologies are acyclic; it means that no concept name appears directly or indirectly in its own definition (resp., in its primitive specification). The semantics of a terminology is obtained by extending the notion of interpretation as follows. An interpretation \mathcal{I} satisfied a terminological axiom $C \equiv D$ (resp., $C \sqsubseteq D$) iff $C^{\mathcal{I}} = D^{\mathcal{I}}$ (resp., $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$). An interpretation \mathcal{I} satisfied (or is a model for) a terminology \mathcal{T} iff \mathcal{I} satisfied every terminological axiom in \mathcal{T} .

Let C and D be two concepts of a terminology \mathcal{T} , the notions of subsumption and equivalence can be extended to the terminology as described below.

- C is subsumed by the concept D according to a terminology \mathcal{T} noted ($C \sqsubseteq_{\mathcal{T}} D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model \mathcal{I} of \mathcal{T} .
- C and D are equivalents according to a terminology \mathcal{T} noted ($C \equiv_{\mathcal{T}} D$) iff $C^{\mathcal{I}} = D^{\mathcal{I}}$ for every model \mathcal{I} of \mathcal{T} .

2.2 Rewriting Queries Using Views

The problem of rewriting queries using views is to find a query expression that uses only a set of views and is equivalent to (or maximally contained in) a given

query Q . The problem of rewriting queries using views is intensively investigated in the database area [4]. This problem is important for query optimization and for applications such as information integration and data warehousing.

We address here the problem of rewriting queries using views as studied in the context of integration information systems [11]. This problem can be expressed in description logics, since, views definitions can be regarded as a terminology \mathcal{T} and queries as concepts. More precisely, let Q be a query expression described using a subset of defined concepts in the terminology \mathcal{T} . A rewriting of Q using \mathcal{T} is a query expression Q' which verifies the two following conditions: (i) Q' is described using only the defined concepts that appear in \mathcal{T} and do not appear in Q , and (ii) $Q' \sqsubseteq_{\mathcal{T}} Q$.

Besides, Q' is called a *maximally-contained rewriting* of Q using \mathcal{T} if there is no rewriting Q'' of Q such that $Q' \sqsubseteq_{\emptyset} Q''$ and $Q'' \sqsubseteq_{\mathcal{T}} Q$. Note that here the subsumption test between Q' and Q'' is achieved according to an empty terminology. This allows to consider, during the test, the concepts that appear in the descriptions of Q' and Q'' as atomic concepts. This allows, for example, to capture the Open World assumption (incomplete views), which is generally admitted in integration systems [11].

3 Generation of P2P Mappings

In this section, we present the problem of P2P mappings generation. This problem consists in the conversion of existing centralized mappings into P2P ones. More precisely, starting from mappings between sources schemas and a global ontology, our aim is at inferring logical relations (e.g., subsumption or equivalence) between concepts of source schemas. We first illustrate below the mapping generation problem on a running example and then we provide a formalization in the DL framework.

Figure 2 describes the inputs of the mapping generation problem: (i) an ontology \mathcal{O} , (ii) a set of sources together with their corresponding schemas $\mathcal{S} = \{S_1, S_2, S_3, S_4\}$, and (iii) a set \mathcal{M} of centralized mappings between sources schemas and the ontology \mathcal{O} . A centralized mapping is a terminological axiom that establishes a semantic links between the concepts that appear in source schemas and the concepts of a global ontology. For example, the mapping $S_3.USAFlightWithStop \equiv USADeparture \sqcap StopFlight$ in Figure 2 specifies that the concept *USAFlightWithStop* of the source S_3 has exactly the same meaning (i.e., is equivalent to) as the flights from USA with stops (i.e., the description $USADeparture \sqcap StopFlight$) in the global ontology. We aim at exploiting such centralized mappings to infer mappings between concepts of sources schemas. These latter mappings, called hereafter P2P mappings, are illustrated in Figure 3. The goal of P2P mappings is to establish semantic links between concepts of different source schemas. For example, the P2P mapping $S_1.FlightToEU \sqcap S_3.USAFlightWithStop \equiv S_2.FlightFromUSA$, specifies that the concept *FlightFromUSA* of the source S_2 has the same meaning as the conjunction of concepts *FlightToEU* of S_1 and *USAFlightWithStop* of S_3 .

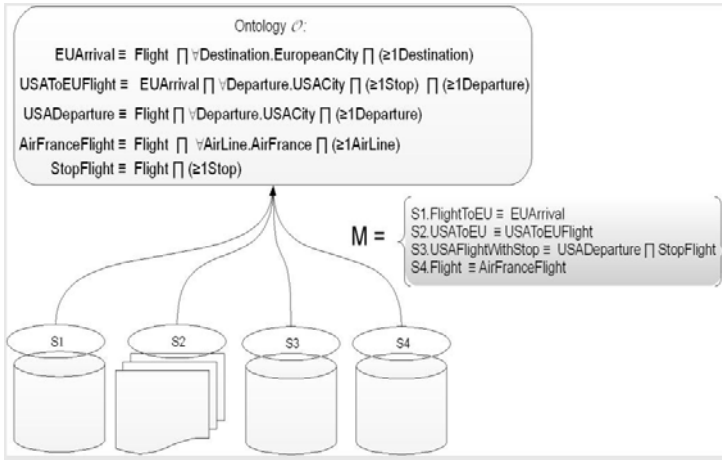


Fig. 2. Example of centralized mappings

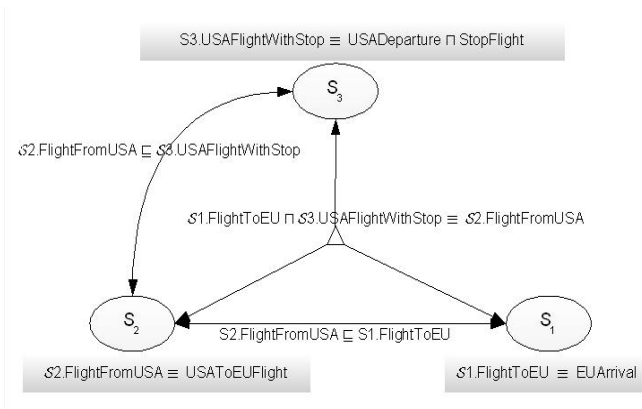


Fig. 3. Example of P2P mappings

The main goal of this paper is to investigate the technical problems underlying the conversion of centralized mappings into P2P ones.

3.1 Formalization in the DL Setting

We consider the following inputs of our problem:

- a global ontology \mathcal{O} , described by mean of a terminology in a given description logic. We denote by D the set of defined concepts in \mathcal{O} ,
- a set $\mathcal{S} = \{S_1, \dots, S_n\}$, where $S_i, i \in \{1, \dots, n\}$, is a terminology that describes a source schema. We note C_{si} the set of defined concepts in S_i , and

- a set of centralized mappings $\mathcal{M} = \{M_1, \dots, M_l\}$ between source schemas and the global ontology. A centralized mapping is given by a terminological axiom $S_i.C_j \equiv D_j$ where $C_j \in C_{si}$, $j \in \{1, n\}$, and D_j is a conjunction of defined concepts from D .

In order to formally define the mapping generation problem, we first introduce the notions of P2P and non redundant (P2P) mappings.

Definition 1 (P2P mapping). *Let \mathcal{O} , \mathcal{M} defined as previously. Let Q_1 and Q_2 be two descriptions defined using concepts from $\bigcup_{i \in \{1, \dots, n\}} C_{si}$ (i.e., using defined concepts that appear in source schemas). We assume that the set of defined concepts used in Q_1 is disjoint from the one used in Q_2 .*

Then, any assertion M' of the form :

- $M' : Q_1 \equiv_{\mathcal{M} \cup \mathcal{O}} Q_2$, or
- $M' : Q_1 \sqsubseteq_{\mathcal{M} \cup \mathcal{O}} Q_2$

is a P2P mapping between source schemas.

As an example, the mapping $S_1.FlightToEU \sqcap S_3.USAFFlightWithStop \equiv S_2.FlightFromUSA$ is a P2P mapping that establishes a semantic link between the schema S_2 and the schemas S_1 and S_3 . Note that, from the previous definition, a P2P mapping establish a link between two disjoint sets of sources schemas (i.e., concepts of a given source schema cannot appear both at right-hand side and left-hand side of a P2P mapping).

It is worth noting that, not all the P2P mappings are interesting to discover. For example, in the Figure 3, the mapping $M'_1 : S_2.FlightFromUSA \sqcap S_3.USAFFlightWithStop \sqsubseteq S_1.FlightToEU$ is redundant because of the presence of the mapping $M'_2 : S_2.FlightFromUSA \sqsubseteq S_1.FlightToEU$. Indeed, the mapping M'_1 do not convey any additional information w.r.t. to the one already provided by the mapping M'_2 . We define below more precisely the notion of non redundant P2P mappings.

Definition 2 (Non redundant P2P mapping). *Let \mathcal{O} , \mathcal{M} and S defined as previously. Let Q_1 , Q_2 and Q' , be descriptions defined using concepts from $\bigcup_{i \in \{1, \dots, n\}} C_{si}$ (i.e., using defined concepts that appear in source schemas).*

A P2P mapping $M' : Q_1 \sqsubseteq_{\mathcal{M} \cup \mathcal{O}} Q_2$, is non redundant w.r.t. $(\mathcal{O}, \mathcal{M}, S)$ if and only if :

- $\not\exists Q' \mid Q_1 \sqsubseteq_{\emptyset} Q'$ and $Q' \sqsubseteq_{\mathcal{M} \cup \mathcal{O}} Q_2$ and
- $\not\exists Q' \mid Q' \sqsubseteq_{\emptyset} Q_2$ and $Q_1 \sqsubseteq_{\mathcal{M} \cup \mathcal{O}} Q'$.

For practical purposes (e.g., query processing in a P2P system), only non redundant mappings are relevant to consider. Continuing with the running example, the presence of the mapping M'_2 makes the mapping M'_1 useless for processing queries. For instance, consider a query $Q \equiv S_1.FlightToEU$, posed on the source S_1 , and that looks for flights to Europe. Using the mapping M'_1 , the query Q can be rewritten into a query $Q' \equiv S_2.FlightFromUSA \sqcap S_3.USAFFlightWithStop$

that computes answers from the sources S_2 and S_3 . In the same way, using the mapping M'_2 , the query Q can be rewritten into a query $Q'' \equiv S_2.FlightFromUSA$ that compute the answers from the sources S_2 . However, we can observe that the set of answers returned by Q' is a subset of Q'' answers. Therefore, in this example, the mapping M'_2 is sufficient to answer the query Q and hence, in the presence of M'_2 , the redundant mapping M'_1 is useless for computing Q answers.

In the remaining of this paper, we focus our attention on the problem of computing non redundant P2P mappings from a set of existing centralized mappings. We show in the next section how instances of such a problem can be reduced to different instances of the problem of query rewriting using views.

3.2 From Mappings Discovery to Query Rewriting

We consider in this section the problem of computing non redundant P2P mappings. This problem, noted *ConvertMapping* is defined precisely below.

Problem 1 (*ConvertMapping*($\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input}$)). *Let \mathcal{O} be a global ontology, \mathcal{M} a set of centralized mappings and $\mathcal{S} = \{S_1, \dots, S_n\}$ a set of sources schemas. The input description Q_{input} is a concept described using the defined concepts from $\bigcup_{i \in \{1, n\}} C_{si}$ (i.e., the description of Q_{input} uses only defined concepts that appear in source schemas). The problem is then to compute all the non redundant mappings w.r.t. $(\mathcal{O}, \mathcal{M}, \mathcal{S})$ of the following forms: $Q_{input} \sqsubseteq_{\mathcal{M} \cup \mathcal{O}} Q$, $Q \sqsubseteq_{\mathcal{M} \cup \mathcal{O}} Q_{input}$ and $Q_{input} \equiv_{\mathcal{M} \cup \mathcal{O}} Q$.*

Hence, the problem is to compute all the non redundant mappings that involve the input description Q_{input} either in their left-hand side or in their right-hand side. It is worth noting that the definition of the *ConvertMapping* problem assumes that the input description Q_{input} is given *a priori* (e.g., provided by a user or selected from the defined concepts of the sources that appear in centralized mappings). Automatically computing relevant *input descriptions* w.r.t. some QoS criteria is an interesting question that is left open in this paper.

Note that, to solve *ConvertMapping*($\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input}$) it is sufficient to focus only on subsumption based mappings (i.e., $Q_{input} \sqsubseteq_{\mathcal{M} \cup \mathcal{O}} Q$ and $Q \sqsubseteq_{\mathcal{M} \cup \mathcal{O}} Q_{input}$) since by definition, their discovery enables to find equivalence mappings when they exist. In the sequel, we decompose a *ConvertMapping* problem into two subproblems: the first one, noted $M_{max}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$, concerns the discovery of P2P mappings of the form $Q \sqsubseteq_{\mathcal{M} \cup \mathcal{O}} Q_{input}$, and the second one, noted $M_{min}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$, concerns the discovery of P2P mappings of the form $Q_{input} \sqsubseteq_{\mathcal{M} \cup \mathcal{O}} Q$.

As stated by the following lemma, the problem $M_{max}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$ can be straightforwardly reduced to the classical problem of query rewriting using views.

Lemma 1. *Let \mathcal{O} , \mathcal{S} , \mathcal{M} and Q_{input} defined as previously. Then:*

$Q \sqsubseteq_{\mathcal{M} \cup \mathcal{O}} Q_{input}$ is a non redundant mapping of Q_{input} w.r.t. $(\mathcal{O}, \mathcal{S}, \mathcal{M})$ iff Q is a maximally contained rewriting of Q_{input} using $\mathcal{M} \cup \mathcal{O}$.

The problem of computing maximally-contained rewriting of a query has been studied in the literature for different types of languages [4,11,17]. In the context of description logics, this problem has been shown to be decidable even for very expressive languages such as $\mathcal{ALCN}\mathcal{R}$ [4].

The next section is devoted to the problem $M_{min}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$.

4 The Problem $M_{min}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$

In this section, we formalize the problem $M_{min}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$ as a new form of query rewriting where the aim is to compute *minimal* rewritings of Q_{input} using $\mathcal{M} \cup \mathcal{O}$. More precisely, we consider two families of description logics: (i) DLs closed under negation, and (ii) DLs with the property of structural subsumption.

Case of DL Closed under Negation. A description logic \mathcal{L} is said to be closed under negation if for any description C in \mathcal{L} , $\neg C$ is also a description in \mathcal{L} . The following lemma says that in the particular case of such logics, the problem $M_{min}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$ can be reduced to a problem of computing maximally contained rewritings.

Lemma 2. *Let $\mathcal{O}, \mathcal{S}, \mathcal{M}$ and Q_{input} defined as previously. Then:*

$Q_{input} \sqsubseteq_{\mathcal{M} \cup \mathcal{O}} Q$ is a non redundant mapping of Q_{input} w.r.t. $(\mathcal{O}, \mathcal{S}, \mathcal{M})$ iff $\neg Q$ is a maximally contained rewriting of $\neg Q_{input}$ using $\mathcal{M} \cup \mathcal{O}$.

The demonstration of this lemma is based on the following axiom : $Q \sqsubseteq_{min} Q'$ is equivalent to $\neg Q' \sqsubseteq_{max} \neg Q$.

A good candidate logic for this case is the language $\mathcal{ALCN}\mathcal{R}$ which displays two interesting properties: it is closed under negation, and the problem of query rewriting using views is decidable for this language.

Case of DL with Structural Subsumption. We consider now a case of another family of logics, namely DL with the structural subsumption. We introduce first some basic notions that enable to define precisely such logics. Then, we show how to solve the $M_{min}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$ problem in this setting using hypergraph techniques.

Definition 3 (Reduced clause form). *Let \mathcal{L} be a description logic. A clause in \mathcal{L} is a description A with the following property:*

$$(A \equiv B \sqcap A') \Rightarrow (B \equiv \top) \wedge (A' \equiv A) \vee (A' \equiv \top) \wedge (B \equiv A).$$

Every conjunction $A_1 \sqcap \dots \sqcap A_n$ of clauses can be represented by the clause set $\hat{A} = \{A_1, \dots, A_n\}$.

$\hat{A} = \{A_1, \dots, A_n\}$ is called a reduced clause set if :

- either $n = 1$.
- or no clause subsumes the conjunction of the other clauses, i.e., $\forall i \mid i \in \{1, \dots, n\}, A_i \not\sqsupseteq (\hat{A} \setminus A_i)$.

The set \hat{A} is then called a reduced clause form (RCF).

Consider for example a description $D \equiv A \sqcap B \sqcap \forall R.C$, where A , B , and C are atomic concepts. Then, an RFC of D is $\hat{D} = \{A, B, \forall R.C\}$. Let us now introduce the notion of structural subsumption as defined in [13].

Definition 4 (Structural subsumption). *The subsumption relation in a description logic \mathcal{L} is said structural iff for any description $A \equiv A_1 \sqcap \dots \sqcap A_n$ and any description $B \equiv B_1 \sqcap \dots \sqcap B_m$ in \mathcal{L} which is given by its RCF, the following holds:*

$$B \sqsubseteq A \Leftrightarrow \forall A_j \in \hat{A}, \exists B_i \in \hat{B} \mid B_i \sqsubseteq A_j$$

In other words, if $B \sqsubseteq A$ then $\hat{A} \subseteq \hat{B}$ ($\{A_j\} \subseteq \{B_i\}$).

We consider now the problem $M_{min}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$ in the framework of DLs with structural subsumption. The following lemma provides a characterization of the solutions in terms of clause sets.

Lemma 3. *(Characterization of the minimal rewritings subsuming a query)*

Let \mathcal{O} , \mathcal{S} , \mathcal{M} and Q_{input} defined as previously but using a description logic with structural subsumption. Then:

$Q_{input} \sqsubseteq_{\mathcal{M} \cup \mathcal{O}} Q$ is a non redundant mapping of Q_{input} w.r.t. $(\mathcal{O}, \mathcal{S}, \mathcal{M})$ iff \hat{Q} is a maximal set of clauses such that $\hat{Q} \subseteq Q_{input}$.

The demonstration of this lemma is based on the structural subsumption propriety. Indeed, with the following descriptions given by their RCFs, if $A \sqcap B \sqcap C \sqsubseteq \mathbf{A} \sqcap \mathbf{B}$ and $A \sqcap B \sqcap C \sqsubseteq \mathbf{A}$, we note here that $\mathbf{A} \sqcap \mathbf{B} \sqsubseteq \mathbf{A}$. So we deduce that the description with the maximal set of clauses is the minimal.

Therefore, to solve a $M_{min}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$ in the context of DLs with structural subsumption, one has to look for descriptions Q such that their RFCs are made of maximal, with respect to set inclusion, sets of clauses that are included in the RFC of the input description. As shown below, this problem can be reduced to a problem of computing minimal transversals of a given hypergraph.

We recall first the definitions of hypergraphs and minimal transversals.

Definition 5. *(Hypergraph and transversals [6]) A hypergraph \mathcal{H} is a pair (Σ, Γ) of a finite set $\Sigma = \{V_1, \dots, V_n\}$ and a set $\Gamma = \{\epsilon_1, \dots, \epsilon_n\}$ of subsets of Σ . The elements of Σ are called vertices, and the elements of Γ are called edges.*

A set $\mathcal{R} \subseteq \Sigma$ is a transversal of \mathcal{H} if for each $\epsilon \in \Gamma$, $\mathcal{R} \cap \epsilon \neq \emptyset$. A transversal \mathcal{R} is minimal if no proper subset \mathcal{R}' of \mathcal{R} is a transversal. The set of the minimal transversals of an hypergraph \mathcal{H} is noted $Tr(\mathcal{H})$.

We show now how to map a $M_{min}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$ problem into a hypergraph based framework. Consider an instance of the problem $M_{min}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$ where all the descriptions are expressed using a description logic with structural subsumption. We assume that concept descriptions are represented by their RCFs. We note by $N_{\mathcal{M}}$ the set of defined concepts of \mathcal{M} that do not belong to one of the source schemas used in Q_{input} . Then, we build the associated hypergraph $\mathcal{H}_{\mathcal{M}_{min}} = (\Sigma, \Gamma)$ as follows.

- Each concept $C_i \in N_{\mathcal{M}}$ such that $\hat{C}_i \subseteq Q_{input}$, is associated with a vertex V_{C_i} in the hypergraph $\mathcal{H}_{\mathcal{M}_{min}}$. So $\Sigma = \{V_{C_i}, i \in \{1, \dots, l\}\}$.
- Each clause $A_j \in Q_{input}$, $j \in \{1, \dots, k\}$ is associated with an edge ϵ_{A_j} in $\mathcal{H}_{\mathcal{M}_{min}}$ such that $\epsilon_{A_j} = \{V_{C_i} \mid A_j \in \{\hat{C}_i \cap Q_{input}\}$.

Consider for example a $M_{min}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$ problem where \mathcal{O} , \mathcal{S} , \mathcal{M} are depicted by the Figure 2 and $Q_{input} \equiv S_2.FlightFromUSA$. The associated hypergraph $\mathcal{H}_{\mathcal{M}_{min}} = (\Sigma, \Gamma)$ is built as follows : The set of vertices is $\Sigma = \{V_{S_1.FlightToEU}, V_{S_3.USAFlightWithStop}\}$ and the set of edges is :

$$\Gamma = \{\epsilon_{(Flight)}, \epsilon_{(\forall Departure.USACity)}, \epsilon_{(\geq 1 Departure)}, \epsilon_{(\forall Destination.EuropeanCity)}, \epsilon_{(\geq 1 Destination)}, \epsilon_{(\geq 1 Stop)}\}.$$

Lemma 4. *Let $M_{min}(\mathcal{O}, \mathcal{M}, Q)$ be a P2P generation problem expressed in the context of a DL with structural subsumption. Then, Q a conjunction of $C_i, i \in \{1, \dots, n\}$ is a solution of $M_{min}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$ iff $\mathcal{R}_Q = \{V_{C_i}, i \in \{1, \dots, n\}\}$ is a minimal transversal of the hypergraph $\mathcal{H}_{\mathcal{M}_{min}}$.*

Proof : $Q \equiv \bigcap_{i=1}^n C_i$, $Q_{input} \equiv \bigcap_{j=1}^m A_j$. Q is a solution of $M_{min}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$. $\forall A_j \in \hat{Q} \cap Q_{input}, \exists C_i \mid A_j \in \hat{C}_i \cap Q_{input} \Leftrightarrow \forall \epsilon_{A_j} \in \mathcal{H}_{\mathcal{M}_{min}}, \exists V_{C_i} \in \mathcal{R}_Q \mid V_{C_i} \in \epsilon_{A_j} \Leftrightarrow \mathcal{R}_Q$ is a transversal of the hypergraph $\mathcal{H}_{\mathcal{M}_{min}}$. Q is a minimal rewriting of $Q_{input} \Leftrightarrow \nexists Q' \mid Q_{input} \sqsubseteq_{\mathcal{M} \cup \mathcal{O}} Q' \sqsubseteq_{\emptyset} Q \Leftrightarrow \mathcal{R}_Q$ is a minimal transversal of $\mathcal{H}_{\mathcal{M}_{min}}$.

This lemma provides a practical method to solve $M_{min}(\mathcal{O}, \mathcal{S}, \mathcal{M}, Q_{input})$. Indeed, it enables to reuse and adapt known techniques and algorithms for computing minimal transversals [5] to our context.

Continuing with the example, considering the hypergraph $\mathcal{H}_{\mathcal{M}_{min}}$, the minimal transversals are $\{V_{S_1.FlightToEU}, V_{S_3.USAFlightWithStop}\}$. So, in this case we have only one minimal rewriting that subsume Q_{input} which is $Q \equiv S_1.FlightToEU \sqcap S_3.USAFlightWithStop$.

5 Conclusion

In this paper, we considered the problem of the automatic generation of semantic P2P mappings between autonomous data source schemas. We formalized this problem using description logics and we showed that it can be reduced to a problem of rewriting queries using views. We have developed a prototype that implements our approach. The algorithms of query rewriting and the computation of the minimal transversals of the hypergraph are implemented using the JAVA language and it is based on the OWL language. The prototype is composed of three modules : Parser, GUI modules and Hypergraph. This latter module is devoted to the computation of minimal rewritings. Our future work will be devoted to exploration of other family of logics for which the investigated problem can be solved. We will also investigate the possibly of automatic generation of *input description*. Finally, quantitative experimentation and algorithmic optimization constitute also interesting future research directions.

References

1. Abiteboul, S., et al.: The lowell database research self-assessment (2003)
2. Adjiman, P., Chatalic, P., Goasdoué, F., Rousset, M.-C., Simon, L.: SomeWhere in the semantic web. In: Fages, F., Soliman, S. (eds.) PPSWR 2005. LNCS, vol. 3703, pp. 1–16. Springer, Heidelberg (2005)
3. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation and Applications. University Press, Cambridge (2003)
4. Beerl, C., Levy, A.Y., Rousset, M.C.: Rewriting queries using views in description logics. In: PODS 1997, pp. 99–108. ACM Press, New York (1997)
5. Berge, C.: Hypergraphs (1989)
6. Eiter, T., Gottlob, G.: Identifying the minimal transversals of an hypergraph and related problems. *SIAM of Computing* 24(6), 1278–1304 (1995)
7. Goasdoué, F., Rousset, M.C.: Answering queries using views: A KRDB perspective for the semantic web. *ACM Transactions on Internet Technology* 4(3), 255–288 (2004)
8. Halevy, A.Y., Ives, Z.G., Suciu, D., Tatarinov, I.: Schema mediation in peer data management systems. In: ICDE, pp. 505–516 (2003)
9. Halevy, A.Y., Ives, Z.G., Madhavan, J., Mork, P., Suciu, D., Tatarinov, I.: The piazza peer data management system. *IEEE Transactions on Knowledge and Data Engineering* 16(7), 787–798 (2004)
10. Lenzerini, M.: Data integration: a theoretical perspective. In: PODS 2002, pp. 233–246. ACM Press, New York (2002)
11. Levy, A.Y.: Answering queries using views: A survey. *The VLDB Journal* 10, 270–294 (2001)
12. Sarma, A.D., Dong, X., Halevy, A.: Bootstrapping pay-as-you-go data integration systems. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pp. 861–874. ACM, New York (2008)
13. Teege, G.: Making the difference: A subtraction operation for description logics. In: Doyle, J., Sandewall, E., Torasso, P. (eds.) Principles of Knowledge Representation and Reasoning: Proc. of the 4th International Conference (KR 1994), pp. 540–550. Morgan Kaufmann Publishers, San Francisco (1994)

An OWL Ontology for Fuzzy OWL 2

Fernando Bobillo¹ and Umberto Straccia²

¹ Dpt. of Computer Science & Systems Engineering, University of Zaragoza, Spain

² Istituto di Scienza e Tecnologie dell'Informazione (ISTI - CNR), Pisa, Italy
fbobillo@unizar.es, straccia@isti.cnr.it

Abstract. The need to deal with vague information in Semantic Web languages is rising in importance and, thus, calls for a standard way to represent such information. We may address this issue by either extending current Semantic Web languages to cope with vagueness, or by providing an ontology describing how to represent such information within Semantic Web languages. In this work, we follow the latter approach and propose and discuss an OWL ontology to represent important features of fuzzy OWL 2 statements.

1 Introduction

It is well-known that “classical” ontology languages are not appropriate to deal with *fuzzy/vague/imprecise knowledge*, which is inherent to several real world domains. Since fuzzy set theory and fuzzy logic [24] are suitable formalisms to handle these types of knowledge, fuzzy ontologies emerge as useful in several applications, such as (multimedia) information retrieval, image interpretation, ontology mapping, matchmaking and the Semantic Web.

Description Logics (DLs) are the basis of several ontology languages. The current standard for ontology representation is OWL (Web Ontology Language), which comprises three sublanguages (OWL Lite, OWL DL and OWL Full). OWL 2 is a recent extension which is currently being considered for standardization [10]. The logical counterparts of OWL Lite, OWL DL and OWL 2 are the DLs $\mathcal{SHIF}(\mathbf{D})$, $\mathcal{SHOIN}(\mathbf{D})$, and $\mathcal{SROIQ}(\mathbf{D})$, respectively. OWL Full does not correspond to any DL, and reasoning with it is undecidable.

Several fuzzy extensions of DLs can be found in the literature (see the survey in [15]) and some fuzzy DL reasoners have been implemented, such as FUZZYDL [8], DELOREAN [4] or FIRE [16]. Not surprisingly, each reasoner uses its own fuzzy DL language for representing fuzzy ontologies and, thus, there is a need for a standard way to represent such information. We may address this issue by either extending current Semantic Web languages to cope with vagueness, or by providing an ontology describing how to represent such information within current Semantic Web languages.

In this work, we follow the latter approach and propose and discuss an OWL ontology to represent some important features of fuzzy OWL 2 statements. We have also developed two open-source parsers that map fuzzy OWL 2 statements

expressed via this ontology into FUZZYDL and DELOREAN statements, respectively. Some appealing advantages of such an approach are that: (i) fuzzy OWL ontologies may easily be shared and reused according to the specified encoding; (ii) the ontology could easily be extended to include other types of fuzzy OWL 2 statements; (iii) current OWL editors can be used to encode a fuzzy ontology; and (iv) it can easily be translated into the syntax of other fuzzy DL reasoners.

The remainder of this paper is organized as follows. In Section 2 we present the definition of the DL $\mathcal{SROIQ}(\mathbf{D})$, the logic behind OWL 2, with fuzzy semantics. We also provide additional constructs, peculiar to fuzzy logic. Section 3 describes our OWL ontology, whereas Section 4 presents how to use it to represent two particular languages, those of FUZZYDL and DELOREAN fuzzy DL reasoners. Section 5 compares our proposal with the related work. Finally, Section 6 sets out some conclusions and ideas for future research.

2 The Fuzzy DL $\mathcal{SROIQ}(\mathbf{D})$

In this section we describe the fuzzy DL $\mathcal{SROIQ}(\mathbf{D})$, inspired by the logics presented in [3,8,21]. In the following, we assume $\bowtie \in \{\geq, >, \leq, <\}$, $\triangleright \in \{\geq, >\}$, $\triangleleft \in \{\leq, <\}$, $\alpha \in (0, 1]$, $\beta \in [0, 1)$, $\gamma \in [0, 1]$.

Syntax. Similarly as for its crisp counterpart, fuzzy $\mathcal{SROIQ}(\mathbf{D})$ assumes three alphabets of symbols, for concepts, roles and individuals. Apart from atomic concept and roles, complex concept and roles can be inductively built.

Let us introduce some notation. C, D are (possibly complex) concepts, A is an atomic concept, R is a (possibly complex) abstract role, R_A is an atomic role, S is a simple role¹, T is a concrete role, $a, b \in \Delta^{\mathcal{I}}$ are *abstract individuals* and $v \in \Delta_{\mathbf{D}}$ is a *concrete individual*.

The syntax of fuzzy concepts and roles is shown in Table 1. Note that the syntax extends the crisp case with salient features of fuzzy DLs [3,8]: fuzzy nominals $\{\alpha_1/o_1, \dots, \alpha_m/o_m\}$, fuzzy implication concepts $C \rightarrow D$, fuzzy weighted sums $\alpha_1 C_1 + \dots + \alpha_k C_k$, modified concept and roles $\text{mod}(C)$ and $\text{mod}(R)$, cut concept and roles $[C \triangleright \alpha]$ and $[R \triangleright \alpha]$, and fuzzy datatypes \mathbf{d} . Furthermore, for each of the connectives $\sqcap, \sqcup, \rightarrow$, we allow the connectives $\sqcap_X, \sqcup_X, \rightarrow_X$, where $X \in \{\text{Gödel}, \text{Lukasiewicz}, \text{Product}\}$, which are interpreted according to the semantics of the subscript.

As fuzzy concrete predicates we allow the following functions defined over $[k_1, k_2] \subseteq \mathbb{Q}^+ \cup \{0\}$: trapezoidal membership function (Fig. 1(a)), the triangular (Fig. 1(b)), the L -function (left-shoulder function, Fig. 1(c)) and the R -function (right-shoulder function, Fig. 1(d)) [20]. For instance, we may define $\text{Young}: \mathbb{N} \rightarrow [0, 1]$, denoting the degree of a person being young, as $\text{Young}(x) = L(10, 30)$. We also allow crisp intervals for backwards compatibility.

We further allow fuzzy modifiers, such as *very*. They are functions $f_m: [0, 1] \rightarrow [0, 1]$ which apply to fuzzy sets to change their membership function. We allow

¹ Simple roles are needed to guarantee the decidability of the logic. Intuitively, simple roles cannot take part in cyclic role inclusion axioms (see [7] for a formal definition).

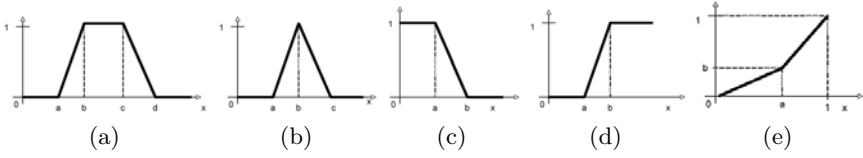


Fig. 1. (a) Trapezoidal function; (b) Triangular function; (c) L -function; (d) R -function; (f) Linear function

modifiers defined in terms of linear hedges (Fig. 1 (e)) and triangular functions (Fig. 1 (b)) [20]. For instance, $very(x) = linear(0.8)$.

Example 1. Concept $Human \sqcap \exists hasAge.L(10, 30)$ denotes the set of young humans, with an age given by $L(10, 30)$. If $linear(4)$ represents the modifier $very$, $Human \sqcap linear(4)(\exists hasAge.L(10, 30))$ denotes $very$ young humans. \square

A *Fuzzy Knowledge Base* (KB) contains a finite set of axioms organized in three parts: a fuzzy ABox \mathcal{A} (axioms about individuals), a fuzzy TBox \mathcal{T} (axioms about concepts) and a fuzzy RBox \mathcal{R} (axioms about roles). A *fuzzy axiom* is an axiom that has a truth degree in $[0,1]$. The axioms that are allowed in our logic are: $\langle a : C \bowtie \gamma \rangle$, $\langle (a, b) : R \bowtie \gamma \rangle$, $\langle (a, b) : \neg R \bowtie \gamma \rangle$, $\langle (a, v) : T \bowtie \gamma \rangle$, $\langle (a, v) : \neg T \bowtie \gamma \rangle$, $\langle a \neq b \rangle$, $\langle a = b \rangle$, $\langle C \sqsubseteq D \triangleright \gamma \rangle$, $\langle R_1 \dots R_m \sqsubseteq R \triangleright \gamma \rangle$, $\langle T_1 \sqsubseteq T_2 \triangleright \gamma \rangle$, $trans(R)$, $dis(S_1, S_2)$, $dis(T_1, T_2)$, $ref(R)$, $irr(S)$, $sym(R)$, and $asy(S)$.

Example 2. $\langle paul : Tall \geq 0.5 \rangle$ states that Paul is tall with at least degree 0.5. The fuzzy RIA $\langle isFriendOf isFriendOf \sqsubseteq isFriendOf \geq 0.75 \rangle$ states that the friends of my friends can also be considered my friends with degree 0.75. \square

Semantics. A fuzzy interpretation \mathcal{I} with respect to a fuzzy concrete domain \mathbf{D} is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a non empty set $\Delta^{\mathcal{I}}$ (the interpretation domain) disjoint with $\Delta_{\mathbf{D}}$ and a fuzzy interpretation function $\cdot^{\mathcal{I}}$ mapping:

- an *abstract individual* a onto an element $a^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$;
- a *concrete individual* v onto an element $v_{\mathbf{D}}$ of $\Delta_{\mathbf{D}}$;
- a *concept* C onto a function $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \rightarrow [0, 1]$;
- an *abstract role* R onto a function $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0, 1]$;
- a *concrete role* T onto a function $T^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}} \rightarrow [0, 1]$;
- an n -ary *concrete fuzzy predicate* \mathbf{d} onto the fuzzy relation $\mathbf{d}_{\mathbf{D}} : \Delta_{\mathbf{D}}^n \rightarrow [0, 1]$;
- a *modifier* mod onto a function $f_{mod} : [0, 1] \rightarrow [0, 1]$.

Given arbitrarities t-norm \otimes , t-conorm \oplus , negation function \ominus and implication function \Rightarrow (see [13] for properties and examples of these fuzzy operators), the fuzzy interpretation function is extended to fuzzy *complex concepts*, *roles* and *axioms* as shown in Table 1.

$C^{\mathcal{I}}$ denotes the membership function of the fuzzy concept C with respect to the fuzzy interpretation \mathcal{I} . $C^{\mathcal{I}}(x)$ gives us the degree of being the individual x an element of the fuzzy concept C under \mathcal{I} . Similarly, $R^{\mathcal{I}}$ denotes the membership

Table 1. Syntax and semantics of concepts, roles and axioms in fuzzy $SR\mathcal{OIQ}(\mathbf{D})$

Syntax (concept C)	Semantics of $C^{\mathcal{I}}(x)$
\top	1
\perp	0
A	$A^{\mathcal{I}}(x)$
$C \sqcap D$	$C^{\mathcal{I}}(x) \otimes D^{\mathcal{I}}(x)$
$C \sqcup D$	$C^{\mathcal{I}}(x) \oplus D^{\mathcal{I}}(x)$
$\neg C$	$\ominus C^{\mathcal{I}}(x)$
$\forall R.C$	$\inf_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x, y) \Rightarrow C^{\mathcal{I}}(y)\}$
$\exists R.C$	$\sup_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x, y) \otimes C^{\mathcal{I}}(y)\}$
$\forall T.d$	$\inf_{v \in \Delta_{\mathbf{D}}} \{T^{\mathcal{I}}(x, v) \Rightarrow d_{\mathbf{D}}(v)\}$
$\exists T.d$	$\sup_{v \in \Delta_{\mathbf{D}}} \{T^{\mathcal{I}}(x, v) \otimes d_{\mathbf{D}}(v)\}$
$\{\alpha_1/o_1, \dots, \alpha_m/o_m\}$	$\sup_{\{i \mid x = o_i^{\mathcal{I}}\}} \alpha_i$
$\geq m \ S.C$	$\sup_{y_1, \dots, y_m \in \Delta^{\mathcal{I}}} [(\min_{i=1}^m \{S^{\mathcal{I}}(x, y_i) \otimes C^{\mathcal{I}}(y_i)\}) \otimes (\otimes_{j < k} \{y_j \neq y_k\})]$
$\leq m \ S.C$	$\inf_{y_1, \dots, y_{m+1} \in \Delta^{\mathcal{I}}} [(\min_{i=1}^{m+1} \{S^{\mathcal{I}}(x, y_i) \otimes C^{\mathcal{I}}(y_i)\}) \Rightarrow (\oplus_{j < k} \{y_j = y_k\})]$
$\geq m \ T.d$	$\sup_{v_1, \dots, v_m \in \Delta_{\mathbf{D}}} [(\min_{i=1}^m \{T^{\mathcal{I}}(x, v_i) \otimes d_{\mathbf{D}}(v_i)\}) \otimes (\otimes_{j < k} \{v_j \neq v_k\})]$
$\leq n \ T.d$	$\inf_{v_1, \dots, v_{n+1} \in \Delta_{\mathbf{D}}} [(\min_{i=1}^{n+1} \{T^{\mathcal{I}}(x, v_i) \otimes d_{\mathbf{D}}(v_i)\}) \Rightarrow (\oplus_{j < k} \{v_j = v_k\})]$
$\exists S.Self$	$S^{\mathcal{I}}(x, x)$
$mod(C)$	$f_{mod}(C^{\mathcal{I}}(x))$
$[C \geq \alpha]$	1 if $C^{\mathcal{I}}(x) \geq \alpha$, 0 otherwise
$[C \leq \beta]$	1 if $C^{\mathcal{I}}(x) \leq \beta$, 0 otherwise
$\alpha_1 C_1 + \dots + \alpha_k C_k$	$\alpha_1 C_1^{\mathcal{I}}(x) + \dots + \alpha_k C_k^{\mathcal{I}}(x)$
$C \rightarrow D$	$C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x)$
Syntax (role R)	Semantics of $R^{\mathcal{I}}(x, y)$
R_A	$R_A^{\mathcal{I}}(x, y)$
U	1
R^-	$R^{\mathcal{I}}(y, x)$
$mod(R)$	$f_{mod}(R^{\mathcal{I}}(x, y))$
$[R \geq \alpha]$	1 if $R^{\mathcal{I}}(x, y) \geq \alpha$, 0 otherwise
T	$T^{\mathcal{I}}(x, v)$
Syntax (axiom τ)	Semantics of $\tau^{\mathcal{I}}$
$a : C$	$C^{\mathcal{I}}(a^{\mathcal{I}})$
$(a, b) : R$	$R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}})$
$(a, b) : \neg R$	$\ominus R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}})$
$(a, v) : T$	$T^{\mathcal{I}}(a^{\mathcal{I}}, v_{\mathbf{D}})$
$(a, v) : \neg T$	$\ominus T^{\mathcal{I}}(a^{\mathcal{I}}, v_{\mathbf{D}})$
$C \sqsubseteq D$	$\inf_{x \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x)$
$R_1 \dots R_m \sqsubseteq R$	$\inf_{x_1, x_{n+1}} \sup_{x_2, \dots, x_n} (R_1^{\mathcal{I}}(x_1, x_2) \otimes \dots \otimes R_n^{\mathcal{I}}(x_n, x_{n+1})) \Rightarrow R^{\mathcal{I}}(x_1, x_{n+1})$ where $x_1 \dots x_{n+1} \in \Delta^{\mathcal{I}}$
$T_1 \sqsubseteq T_2$	$\inf_{x \in \Delta^{\mathcal{I}}, v \in \Delta_{\mathbf{D}}} T_1^{\mathcal{I}}(x, v) \Rightarrow T_2^{\mathcal{I}}(x, v)$

function of the fuzzy role R with respect to \mathcal{I} . $R^{\mathcal{I}}(x, y)$ gives us the degree of being (x, y) an element of the fuzzy role R under \mathcal{I} .

Let $\phi \in \{a : C, (a, b) : R, (a, b) : \neg R, (a, v) : T, (a, v) : \neg T\}$ and $\psi \in \{C \sqsubseteq D, R_1 \dots R_m \sqsubseteq R, T_1 \sqsubseteq T_2\}$. $\phi^{\mathcal{I}}$ and $\psi^{\mathcal{I}}$ are defined in Table II. Then, a fuzzy interpretation \mathcal{I} satisfies (is a model of):

- $\langle \phi \boxtimes \gamma \rangle$ iff $\phi^{\mathcal{I}} \boxtimes \gamma$,
- $\langle a \neq b \rangle$ iff $a^{\mathcal{I}} \neq b^{\mathcal{I}}$,
- $\langle a = b \rangle$ iff $a^{\mathcal{I}} = b^{\mathcal{I}}$,
- $\langle \psi \triangleright \gamma \rangle$ iff $\psi^{\mathcal{I}} \triangleright \gamma$,
- **trans**(R) iff $\forall x, y \in \Delta^{\mathcal{I}}, R^{\mathcal{I}}(x, y) \geq \sup_{z \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, z) \otimes R^{\mathcal{I}}(z, y)$,
- **dis**(S_1, S_2) iff $\forall x, y \in \Delta^{\mathcal{I}}, S_1^{\mathcal{I}}(x, y) = 0$ or $S_2^{\mathcal{I}}(x, y) = 0$,
- **dis**(T_1, T_2) iff $\forall x \in \Delta^{\mathcal{I}}, v \in \Delta_{\mathbf{D}}, T_1^{\mathcal{I}}(x, v) = 0$ or $T_2^{\mathcal{I}}(x, v) = 0$,
- **ref**(R) iff $\forall x \in \Delta^{\mathcal{I}}, R^{\mathcal{I}}(x, x) = 1$,
- **irr**(S) iff $\forall x \in \Delta^{\mathcal{I}}, S^{\mathcal{I}}(x, x) = 0$,
- **sym**(R) iff $\forall x, y \in \Delta^{\mathcal{I}}, R^{\mathcal{I}}(x, y) = R^{\mathcal{I}}(y, x)$,

- $\text{asy}(S)$ iff $\forall x, y \in \Delta^{\mathcal{I}}$, if $S^{\mathcal{I}}(x, y) > 0$ then $S^{\mathcal{I}}(y, x) = 0$,
- a fuzzy KB iff it satisfies each of its axioms.

3 An OWL Ontology for Fuzzy OWL 2

In this section we describe FUZZYOWL2ONTOLOGY, the OWL ontology that we have developed with the aim of representing a fuzzy extension of OWL 2. An excerpt of the ontology is shown in Fig. 2.

FUZZYOWL2ONTOLOGY has 8 main classes representing different elements of a fuzzy ontology (of course, each of these classes has several subclasses):

- Individual simply represents an individual of the vocabulary.
- Concept, represents a fuzzy concept of the vocabulary. A concept can be an AbstractConcept or a ConcreteConcept. These two classes have several subclasses, covering the complex constructors already defined in Section 2.
- Property, represents a fuzzy role. A property can be concrete (DatatypeProperty) or abstract (ObjectProperty). These two classes have a lot of subclasses, covering the complex constructors already defined in Section 2.
- Axiom represents the axioms defined in Section 2. Axioms can be grouped in three categories: ABoxAxiom, TBoxAxiom and RBoxAxiom. Some of the

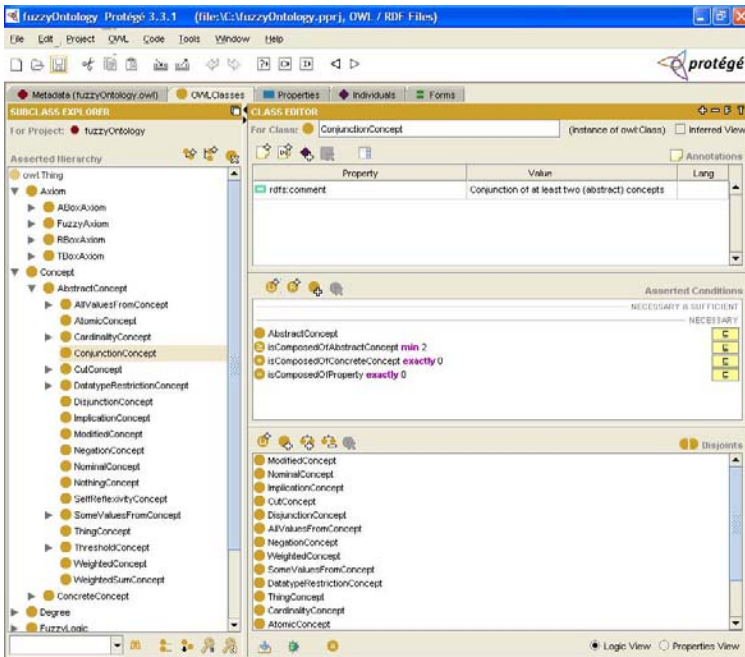


Fig. 2. An excerpt of FUZZYOWL2ONTOLOGY

axioms are subclasses of `FuzzyAxiom`, which indicates that the axiom is not either true or false, but that it is true to some extent.

- `Degree` represents a degree which can be added to an instance of `FuzzyAxiom`. `LinguisticDegree`, `ModifierDegree`, `NumericDegree` and `Variable` are subclasses.
- `Query` represents a special kind of axioms, queries to be submitted to a fuzzy DL reasoner. Current subclasses are `ConceptSatisfiabilityQuery`, `EntailmentQuery`, `GreatestConceptSatisfiabilityQuery`, `GreatestLowerBoundQuery`, `LowestUpperBoundQuery`, `OntologyConsistencyQuery` and `SubsumptionQuery`.
- `FuzzyLogic` represents different families of fuzzy operators which can be used to give different semantics to the logic. Current subclasses are `Zadeh`, `Goedel`, `Lukasiewicz` and `Product`. They can be linked via the property `hasSemantics`.
- `FuzzyModifier` represents a fuzzy modifier which can be used to modify the membership function of a fuzzy concept or a fuzzy role. Current subclasses are `LinearFuzzyModifier` and `TriangularFuzzyModifier`.

There are also some object properties, establishing relations between concepts, and datatype properties, defining attributes of them.

Example 3

- A `ConceptAssertion` axiom has several object properties: `isComposedOfAbstractConcept`, `isComposedOfAbstractIndividual` and `hasDegree`.
- A `TriangularConcreteFuzzyConcept` has several datatype properties representing the parameters k_1, k_2, a, b, c of the membership function, as shown in Section 2: `hasParameterA`, `hasParameterB`, `hasParameterC`, `hasParameterK1` and `hasParameterK2`. □

The integrity of the semantics is maintained with several domain, range, functionality and cardinality axioms.

Example 4

- The range of `isComposedOfIndividual` is `Individual`.
- `ConceptAssertion` has exactly one relation `isComposedOfAbstractIndividual`.
- `DisjointConceptAssertion` has at least two `isComposedOfConcept` relations. □

In some cases the order of the relations is important. For example, `ConceptInclusion` is related to two concepts, one being the subsumer and another being the subsumed. A similar situation happens with `PropertyAssertion` and `PropertyInclusion`. For this reason, there are two types of `isComposedOfConcept`, `isComposedOfAbstractIndividual` and `isComposedOfProperty` axioms. In the special case of `ComplexObjectPropertyInclusionAxiom` the subsumed role is related (via `hasChainProperty`) to another role and so on, forming a chain of roles.

Currently, our ontology has 168 classes, 28 object properties, 11 datatype properties and no instances. When a user wants to build a fuzzy ontology using `FUZZYOWL2ONTOLOGY`, he needs to populate our ontology with instances representing the axioms and the elements of its ontology.

Example 5. In order to represent that `car125` is an expensive Sedan car with at least degree 0.5, we proceed as follows:

- Create an instance of `Individual` called `car125`.
- Create an instance of `AtomicConcept` called `Sedan`.
- Create an instance of `AtomicConcept` called `ExpensiveCar`.
- Create an instance of `ConjunctionConcept`. Assume that the name is `conj1`.
- Create an instance of `NumericDegree`. Assume that the name is `deg1`.
- Create an instance of `ConceptAssertion`. Assume that the name is `ass1`.
- Create a datatype property `hasNumericValue` between `deg1` and “0.5”.
- Create an object property `isComposedOfAbstractConcept` between `conj1` and `Sedan`, and another one between `conj1` and `ExpensiveCar`.
- Create an object property `isComposedOfAbstractIndividual` between `ass1` and `car125`, and another one between `ass1` and `conj1`.
- Create an object property `hasNumericDegree` between `ass1` and `deg1`. □

Once the user has populated the ontology, it is possible to perform a consistency test over the OWL ontology, in order to check that all the axioms (for example, functionality of the roles) are verified, and thus, that the fuzzy ontology is syntactically correct. It is not possible though to check if the fuzzy ontology is consistent using standard reasoners for OWL.

4 FuzzyOWL2Ontology in Use

As an example of application of the `FUZZYOWL2ONTOLOGY`, we have developed two open-source parsers mapping fuzzy ontologies represented using this ontology into the syntax supported by different fuzzy DL reasoners, in particular `FUZZYDL` [8] and `DELOREAN` [4]. Currently, the parsers support fuzzy $SHIF(\mathbf{D})$, the common fragment to them.

The syntax of `FUZZYDL` can be found in [8], whereas the syntax of `DELOREAN` can be found in [2]. Both fuzzy DL reasoners have a similar Lisp-based syntax, but there are a lot of differences, which makes the manual codification of a fuzzy ontology in the two syntaxes a very tedious and error-prone task. This can be avoided by using `FUZZYOWL2ONTOLOGY` as an intermediate step.

The parsers have been developed in Java language using OWL API [2], which is an open-source API for manipulating OWL 2 ontologies [14] (we recall though that `FUZZYOWL2ONTOLOGY` is in OWL).

Each of these parsers works as follows (and consequently, similar parsers could be easily built). The input is a text file containing an ontology obtained after having populated `FUZZYOWL2ONTOLOGY` with OWL statements represented axioms in fuzzy $SRIOQ(\mathbf{D})$. To start with, OWL API is used to obtain a model representing the OWL ontology. Then, we iterate over the axioms and, for each of them, we compute the translation into the syntax of the particular fuzzy DL reasoner. We do not only have to translate the axioms, but also the elements (concepts, roles, individuals, fuzzy concrete domains ...) that take part in it.

Given an instance of the `Axiom` class, the parser navigates through its relation to obtain its components (for example, in a `ConceptAssertion` the parser gets the fillers for `isComposedOfAbstractConcept`, `isComposedOfAbstractIndividual` and `hasDegree`). A similar situation occurs with complex concepts and roles.

² <http://owlapi.sourceforge.net>

The parser also takes into account the fact that some of the axioms may need to introduce a previous definition. For instance, in FUZZYDL we need to define a trapezoidal fuzzy concept before using it.

FUZZYOWL2ONTOLOGY is very expressive, and no reasoner can currently support all of its constructors. Hence, if the reasoner does not support an OWL statement or one of the elements that take part in it, a warning message is shown and the axiom is skipped.

Example 6. As an example of the differences between them, assume that the age of a person ranges in $[0, 200]$ and consider the concept $\exists \text{hasAge.L}(10; 30)$.

In DELOREAN, it is represented as: `(some hasAge (trapezoidal 0 10 30 200))`.

In FUZZYDL, in addition to the axiom we also need a previous definition:

```
(define-fuzzy-concept trap left-shoulder(0, 200, 10, 30))
(some hasAge trap) □
```

In order to demonstrate the coverage of OWL 2, we have also developed a parser translating an OWL 2 ontology (for the moment without datatypes) into FUZZY-OWL2ONTOLOGY. This way, the user can import existing ontologies in an automatic way, as a previous step to extend them to the fuzzy case.

5 Discussion and Related Work

This is, to the best of our knowledge, the first ontology for fuzzy ontology representation. However, a similar idea has been presented in [1], where an OWL ontology is used to describe and build fuzzy relational databases.

The W3C Uncertainty Reasoning for the World Wide Web Incubator Group (URW3-XG) defined an ontology of uncertainty, a vocabulary which can be used to annotate different pieces of information with different types of uncertainty (e.g. vagueness, randomness or incompleteness), the nature of the uncertainty, etc. [23]. But unlike our ontology, it can only be used to identify some kind of uncertainty, and not to represent and manage uncertain pieces of information.

Fuzzy extensions of ontology languages have been presented, more precisely OWL [11][18] and OWL 2 [17], but they are obviously not complaint with the current standard.

A pattern for uncertainty representation in ontologies has also been presented [22]. However, it relies in OWL Full, thus not making possible for instance to check the syntactic correctness of the fuzzy ontology.

Our approach should not be confused with a series of works that describe, given a fuzzy ontology, how to obtain an equivalent OWL ontology (see for example [2,3,5,6,7,9,17,19]). In these works it is possible to reason using a crisp DL reasoner instead of a fuzzy DL reasoner, which is not our case. However, the obtained OWL ontologies cannot be easily understood by humans, as it happens under our approach.

Another approach to represent uncertainty without extending the standard language is to use annotation properties [12]. Despite the simplicity of this approach, it also has several disadvantages with respect to our approach. The

formal semantics of annotation properties is rather limited. More precisely, it is not possible to reason using standard tools with the fuzzy part of the ontology. The fact that an essential part of the ontology is not automatically understood is actually opposed to the philosophy of ontologies. Annotations are useful for “minimalist” extensions of the language, such as for example just adding a degree to an axiom. However, they are not so appropriate for new concept or role constructors. Furthermore, it uses OWL 2 annotation properties, whereas we are complaint with the current standard language OWL.

6 Conclusions and Future Work

In this paper we have proposed FUZZYOWL2ONTOLOGY, an OWL ontology to represent fuzzy extensions of the OWL and OWL 2 languages. The main advantages of our approach is that we are complaint with the standard ontology language and we can perform some reasoning with the meta-model by using standard OWL reasoners. We have proved its utility by means of a couple of parsers translating fuzzy ontologies represented with FUZZYOWL2ONTOLOGY into the common fragment of the languages supported by the fuzzy DL reasoners FUZZYDL and DELOREAN. We have also implemented a parser translating from OWL 2 into FUZZYOWL2ONTOLOGY statements.

Our approach is extensible, the ontology can easily be augmented to other fuzzy statements, and similar parsers could be built for other fuzzy DL reasoners. The parsers and the ontology are available from the FUZZYDL web site³.

In future work we plan to develop a graphical interface such as a Protégé plug-in to assist users in the population of FUZZYOWL2ONTOLOGY. We would also like to extend the parser to fully cover the languages supported by FUZZYDL and DELOREAN, and to cover the opposite directions of the translations.

References

1. Blanco, I.J., Vila, M.A., Martínez-Cruz, C.: The use of ontologies for representing database schemas of fuzzy information. *International Journal of Intelligent Systems* 23(4), 419–445 (2008)
2. Bobillo, F.: Managing vagueness in ontologies. PhD thesis, University of Granada, Spain (2008)
3. Bobillo, F., Delgado, M., Gómez-Romero, J.: A crisp representation for fuzzy *SHOIN* with fuzzy nominals and general concept inclusions. In: da Costa, P.C.G., d’Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) *URSW 2005 - 2007. LNCS (LNAI)*, vol. 5327, pp. 174–188. Springer, Heidelberg (2008)
4. Bobillo, F., Delgado, M., Gómez-Romero, J.: DeLorean: A reasoner for fuzzy OWL 1.1. In: *Proc. of the 4th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2008)*. *CEUR Workshop Proceedings*, vol. 423 (2008)
5. Bobillo, F., Delgado, M., Gómez-Romero, J.: Optimizing the crisp representation of the fuzzy description logic *SRDQ*. In: da Costa, P.C.G., d’Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) *URSW 2005 - 2007. LNCS (LNAI)*, vol. 5327, pp. 189–206. Springer, Heidelberg (2008)

³ <http://www.straccia.info/software/fuzzyDL/fuzzyDL.html>

6. Bobillo, F., Delgado, M., Gómez-Romero, J.: Crisp representations and reasoning for fuzzy ontologies. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 17(4) (2009)
7. Bobillo, F., Delgado, M., Gómez-Romero, J., Straccia, U.: Fuzzy Description Logics under Gödel semantics. *Int. J of Approximate Reasoning* 50(3), 494–514 (2009)
8. Bobillo, F., Straccia, U.: fuzzyDL: An expressive fuzzy Description Logic reasoner. In: *Proceedings of the 17th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2008)*, pp. 923–930. IEEE Computer Society, Los Alamitos (2008)
9. Bobillo, F., Straccia, U.: Towards a crisp representation of fuzzy Description Logics under Lukasiewicz semantics. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) *Foundations of Intelligent Systems. LNCS (LNAI)*, vol. 4994, pp. 309–318. Springer, Heidelberg (2008)
10. Cuenca-Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P.F., Sattler, U.: OWL 2: The next step for OWL. *Journal of Web Semantics* 6(4), 309–322 (2008)
11. Gao, M., Liu, C.: Extending OWL by fuzzy Description Logic. In: *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2005)*, pp. 562–567. IEEE Computer Society, Los Alamitos (2005)
12. Klinov, P., Parsia, B.: Optimization and evaluation of reasoning in probabilistic description logic: Towards a systematic approach. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) *ISWC 2008. LNCS*, vol. 5318, pp. 213–228. Springer, Heidelberg (2008)
13. Hájek, P.: *Metamathematics of Fuzzy Logic*. Kluwer, Dordrecht (1998)
14. Horridge, M., Bechhofer, S., Noppens, O.: Igniting the OWL 1.1 touch paper: The OWL API. In: *Proc. of the 3rd International Workshop on OWL: Experiences and Directions (OWLED 2007)*. *CEUR Workshop Proceedings*, vol. 258 (2007)
15. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in Description Logics for the Semantic Web. *Journal of Web Semantics* 6(4), 291–308 (2008)
16. Stoilos, G., Simou, N., Stamou, G., Kollias, S.: Uncertainty and the Semantic Web. *IEEE Intelligent Systems* 21(5), 84–87 (2006)
17. Stoilos, G., Stamou, G.: Extending fuzzy Description Logics for the Semantic Web. In: *Proc. of the 3rd International Workshop on OWL: Experience and Directions (OWLED 2007)*. *CEUR Workshop Proceedings*, vol. 258 (2007)
18. Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J.Z., Horrocks, I.: Fuzzy OWL: Uncertainty and the Semantic Web. In: *Proc. of the Workshop on OWL: Experience and Directions (OWLED 2005)*. *CEUR Workshop Proceedings*, vol. 188 (2005)
19. Straccia, U.: Transforming fuzzy Description Logics into classical description logics. In: Alferes, J.J., Leite, J. (eds.) *JELIA 2004. LNCS (LNAI)*, vol. 3229, pp. 385–399. Springer, Heidelberg (2004)
20. Straccia, U.: Description Logics with fuzzy concrete domains. In: *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, UAI 2005* (2005)
21. Straccia, U.: A fuzzy Description Logic for the Semantic Web. In: Sanchez, E. (ed.) *Fuzzy Logic and the Semantic Web. Capturing Intelligence*, vol. 1, pp. 73–90. Elsevier Science, Amsterdam (2006)
22. Vacura, M., Svátek, V., Smrž, P.: A pattern-based framework for representation of uncertainty in ontologies. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) *TSD 2008. LNCS (LNAI)*, vol. 5246, pp. 227–234. Springer, Heidelberg (2008)
23. W3C Incubator Group on Uncertainty Reasoning for the World Wide Web Final Report (2008), <http://www.w3.org/2005/Incubator/urw3/XGR-urw3>
24. Zadeh, L.A.: Fuzzy sets. *Information and Control* 8, 338–353 (1965)

Fuzzy Clustering for Categorical Spaces

An Application to Semantic Knowledge Bases

Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito

Dipartimento di Informatica, Università degli studi di Bari
Campus Universitario, Via Orabona 4, 70125 Bari, Italy
{fanizzi,claudia.damato,esposito}@di.uniba.it

Abstract. A multi-relational clustering method is presented which can be applied to complex knowledge bases storing resources expressed in the standard Semantic Web languages. It adopts effective and language-independent dissimilarity measures that are based on a finite number of dimensions corresponding to a committee of discriminating features (represented by concept descriptions). The clustering algorithm expresses the possible clusterings in tuples of central elements (medoids, w.r.t. the given metric) of variable length. It iteratively adjusts these centers following the rationale of fuzzy clustering approach, i.e. one where the membership to each cluster is not deterministic but rather ranges in the unit interval. An experimentation with some ontologies proves the feasibility of our method and its effectiveness in terms of clustering validity indices.

1 Clustering in Complex Domains

Recently, multi-relational learning methods are being devised for knowledge bases in the Semantic Web (henceforth SW), expressed in the standard representations. Indeed, the most burdensome related maintenance tasks, such as *ontology construction, refinement* and *evolution*, demand such automatization also to enable further SW applications.

In this work, we investigate on unsupervised learning for knowledge bases expressed in such standard languages. In particular, we focus on the problem of clustering semantically annotated resources. The benefits of clustering can be manifold. Clustering annotated resources enables the definition of new emerging concepts (*concept formation*) on the grounds of the concepts defined in a knowledge base; supervised methods can exploit these clusters to induce new concept definitions or to refining existing ones (*ontology evolution*); intensionally defined groupings may speed-up the task of approximate search and *discovery* [1]; a clustering may also suggest criteria for *ranking* the retrieved resources based on the distance from the cluster centers.

Most clustering methods are based on the application of similarity (or density) measures defined over a set of attributes of the domain objects. Classes of objects are taken as collections that exhibit low interclass similarity (density) and high intraclass similarity (density). Few methods are able to take into account some form of *background knowledge* that could characterize object configurations by means of global concepts and semantic relationships [2].

Specific approaches designed for terminological representations (*Description Logics* [3], henceforth DLs), have been introduced [4, 5]. These logic-based clustering methods

were devised for some specific DL languages of limited expressiveness. The main drawback of these methods is that they are language-dependent, which prevents them to scale to the standard SW representations that are mapped on complex DLs. Moreover, purely logic methods can hardly handle noisy data while distance-based ones may be more robust. Hence, from a technical viewpoint, suitable measures for concept representations and their semantics are to be devised. A further theoretical problem comes from the increased indeterminacy determined by the *Open-World semantics* that is adopted on the knowledge bases, differently from the *Closed-World semantics* which is more generally adopted in other contexts (e.g. databases).

These problems motivate the investigation on similarity-based clustering methods which can be more noise-tolerant and language-independent. Specifically, the extension of distance-based techniques is proposed, which can cope with the standard SW representations and profit by the benefits of a randomized search for optimal clusterings. Indeed, the method is intended for grouping similar resources w.r.t. a notion of similarity, coded in a distance measure, which fully complies with the semantics of knowledge bases expressed in DLs. The individuals are gathered around cluster centers according to their distance. The choice of the best centers (and their number) is performed through a fuzzy membership approach [6].

Although some structural dissimilarity measures have been proposed for some specific DLs of fair expressiveness [1], they are still partly based on structural criteria which makes them fail to fully grasp the underlying semantics and hardly scale to more complex ontology languages such as those backing the OWL ontology language¹. Therefore, we have devised a family of semi-distance measures for semantically annotated resources, which can overcome the aforementioned limitations [7, 8]. Following the criterion of semantic discernibility of individuals, a family of measures is derived that is suitable for a wide range of languages since it is merely based on the discernibility of the input individuals with respect to a fixed committee of features represented by a set of concept definitions. Hence, the new measures are not absolute, they rather depend on the knowledge base they are applied to. Thus, also the choice of good feature may deserve a preliminary optimization phase, which can be performed by means of a randomized search procedures [8].

In the target setting, the notion of *centroid* characterizing distance-based algorithms for numeric representations descending from K-MEANS [9], is replaced by the notion of *medoids* as cluster prototypes which fit better categorical representations [10]. Differently from these deterministic approaches, the proposed clustering algorithm employs a notion of fuzzy membership w.r.t. the current medoids computed according to the measure mentioned above. On each iteration, the choice of medoids evolves by adjusting the membership probability w.r.t. each medoid.

The paper is organized as follows. Sect. 2 presents the basics of the target representation and the semantic similarity measures adopted. This algorithm is presented and discussed in Sect. 3. We report in Sect. 4 an experiment aimed at assessing the validity of the method on some ontologies available in the Web. Conclusions and extensions are finally examined in Sect. 6.

¹ <http://www.w3.org/TR/owl-guide/>

2 Metrics for DL Representations

2.1 Preliminaries on the Representation

In the following, we assume that resources, concepts and their relationship may be defined in terms of a generic ontology language that may be mapped to some DL language with the standard model-theoretic semantics (see the DLs handbook [3] for a thorough reference). As mentioned in the previous section, one of the advantages of our method is that it does not depend on a specific language for semantic annotations based on DLs. However the implementation applies to OWL-DL knowledge bases.

In the reference DL framework, a *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ contains a *TBox* \mathcal{T} , an *RBox* \mathcal{R} and an *ABox* \mathcal{A} . \mathcal{T} is a set of concept definitions: $C \equiv D$, where C is the atom denoting the defined concept and D is a DL concept description specified by the application of the language constructors to *primitive concepts* and *roles*. The *RBox* \mathcal{R} contains similar axioms for specifying new roles by means of proper constructors. The complexity of such definitions depends on the specific DL language. \mathcal{A} contains *assertions* (ground facts) on *individuals* (domain objects) concerning the current world state, namely $C(a)$ (*class-membership*), a is an instance of concept C , and $R(a, b)$ (*relations*), a is R -related to b . The set of the individuals referenced in the assertions ABox \mathcal{A} is usually denoted with $\text{Ind}(\mathcal{A})$. Each individual can be assumed to be identified by a constant (or its own URI in OWL-DL), however this is not bound to be a one-to-one mapping (*unique names assumption*).

A set-theoretic semantics is generally adopted with these representations, with interpretations \mathcal{I} which map each concept description C to a subset of a domain of objects (*extension*) $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and each role description R to $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. This allows the formation of complex hierarchies of concept/roles.

In this context the most common inference is the computation of the *subsumption* relationship between concepts: given two concept descriptions C and D , D *subsumes* C , denoted by $C \sqsubseteq D$, iff for every interpretation \mathcal{I} it holds that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. The interpretations of interest are those that satisfy all axioms in the knowledge base \mathcal{K} , i.e. its *models*. Model-theoretic entailment will be denoted with \models .

Several other inference services are provided by the standard automated reasoners. Like all other instance-based methods, the measures proposed in this section require performing *instance-checking*, which amounts to determining whether an individual, say a , belongs to a concept extension, i.e. whether $\mathcal{K} \models C(a)$ holds for a certain concept C . In the simplest cases (primitive concepts) instance-checking requires simple ABox lookups, yet for defined concepts the reasoner may need to perform a number of inferences. It is worthwhile to recall that the *Open World Assumption* (OWA) is made. Thus, differently from the standard database framework, reasoning procedures might be unable to ascertain the class-membership or non-membership. Hence one has to cope with this form of uncertainty.

2.2 Comparing Individuals within Ontologies

In distance-based cluster analysis, a function for measuring the (dis)similarity of individuals is needed. It can be observed that individuals do not have a syntactic structure

that can be compared. This has led to lifting them to the concept level before comparing them [1] (resorting to the approximation of the *most specific concept* of an individual w.r.t. the ABox [3]).

Inspired from some techniques for distance construction and *Multi-dimensional Scaling* [6, 11], we have proposed the definition of totally semantic distance measures for individuals in the context of a knowledge base which is also able to cope with the OWA. On a semantic level, similar individuals should behave similarly with respect to the same concepts. We have introduced a novel measure, which is based on the idea of comparing their semantics along a number of dimensions represented by a committee of concept descriptions. Thus, the rationale of the new measure is to compare individuals on the grounds of their behavior w.r.t. a given collection of concept descriptions, say $F = \{F_1, F_2, \dots, F_m\}$, which stands as a group of discriminating *features* expressed in the considered DL language.

The general form of the family of dissimilarity measures for individuals inspired to the Minkowski's distances (L_p) can be defined as follows [7, 8]:

Definition 2.1 (family of dissimilarity measures). *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a knowledge base. Given a set of concept descriptions $F = \{F_1, F_2, \dots, F_m\}$ and a normalized vector of related weights w , a family of dissimilarity measures $\{d_p^F\}_{p \in \mathbb{N}}$, with $d_p^F : \text{Ind}(\mathcal{A}) \times \text{Ind}(\mathcal{A}) \mapsto [0, 1]$, is defined as follows:*

$$\forall a, b \in \text{Ind}(\mathcal{A}) \quad d_p^F(a, b) := \left[\sum_{i=1}^m |w_i \cdot (\pi_i(a) - \pi_i(b))|^p \right]^{\frac{1}{p}}$$

where the projection function vector π is defined $\forall i \in \{1, \dots, m\}$

$$\forall a \in \text{Ind}(\mathcal{A}) \quad \pi_i(a) = \begin{cases} 1 & \mathcal{K} \models F_i(a) & (\text{alt. } F_i(x) \in \mathcal{A}) \\ 0 & \mathcal{K} \models \neg F_i(a) & (\text{alt. } \neg F_i(x) \in \mathcal{A}) \\ 1/2 & \text{otherwise} \end{cases}$$

Note that the measure is efficiently computed when the feature concepts F_i are such that the KBMS can directly infer the truth of the assertions $F_i(a)$, ideally $\forall a \in \text{Ind}(\mathcal{A}) : F_i(a) \in \mathcal{A}$. This is very important for the measure integration in algorithms which massively use them, such as all instance-based methods. The presented method can be regarded as a form of propositionalization [12].

The given definition of the projection functions is basic. The case of $\pi_i(a) = 1/2$ corresponds to the case when a reasoner cannot give the truth value for a certain membership query. This is due to the open-world semantics adopted in this context.

An intermediate value is just a raw (uniform) estimate of the uncertainty related to the single feature. By properly assigning the weights to vector w it is possible to obtain a better measure which reflects the available knowledge [11].

3 Fuzzy Clustering in Complex Domains

The schemata of many similarity-based clustering algorithms (see [9] for a survey) can be adapted to more complex settings like the one of interest for this work, especially when similarity or similarity measures are available.

We focus on a generalization of distance-based methods adopting the notion of prototypes as cluster centers [10]. The method implements a fuzzy clustering scheme [6], where the representation for the clusterings that are iteratively adjusted is made up of tuples of prototypical individuals for the various clusters but, differently from the k -MEANS the membership of the instances to the various clusters is probabilistic rather than deterministic.

The algorithm searches the space of possible clusterings of individuals, optimizing a fitness function L based on the relative discernibility of the individuals of the different clusters (inter-cluster separation) and on the intra-cluster similarity measured in terms of the d_p^F pseudo-metric. Considered a set of cluster centers (prototypes) $\{\mu_1, \dots, \mu_k\}$, a notion of graded membership of an individual x_i w.r.t. a given cluster C_j is introduced ranging in $[0, 1]$. This corresponds to computing the probability $P(C_j|x_i, \theta)$.

The objective function to be minimized can be written:

$$L = \sum_{i=1}^N \sum_{j=1}^k (P(C_j|x_i, \theta))^b d(x_i, \mu_j)$$

Its minima are found solving the equations involving the partial derivatives w.r.t. the medoids $\partial L / \partial \mu_j = 0$ and of the probability $\partial L / \partial \hat{P}_j = 0$, yielding:

$$\mu_j = \frac{\sum_i (P(C_j|x_i))^b \cdot x_i}{\sum_i (P(C_j|x_i))^b} \quad \forall j \in \{1, \dots, k\} \tag{1}$$

and

$$P(C_j|x_i) = \frac{(1/d_{ij})^{\frac{1}{b-1}}}{\sum_r (1/d_{ir})^{\frac{1}{b-1}}} \quad \forall i \in \{1, \dots, N\} \quad \forall j \in \{1, \dots, k\} \tag{2}$$

where $d_{ij} = d(x_i, \mu_j)$.

In a categorical setting, the notion of *medoid* was introduced [10, 9] for categorical feature-spaces w.r.t. some distance measure. Namely, the medoid of a group of individuals is the individual that has the minimal average distance w.r.t. the others. Formally:

Definition 3.1 (medoid). *Given a set of individuals S and a dissimilarity measure d , the medoid of the set is defined:*

$$\mu_S = \text{medoid}(S) := \operatorname{argmin}_{a \in S} \frac{1}{|S|} \sum_{b \in S} d(a, b) \tag{3}$$

In the setting of interest, the prototypes are not numerical tuples but actual individuals (medoids). Eqs. 1 and 3 may be summed up in a single one as follows:

$$\mu_j = \operatorname{argmin}_{a \in C_j} \sum_{b \in C_j} d(a, b) \cdot P(C_j|a) \quad \forall j \in \{1, \dots, k\} \tag{4}$$

i.e. the medoids are determined by the individuals minimizing the distance to the other members of the cluster, weighted by their membership probability. Finally, a specific similarity measure for individuals like those defined in the previous section is needed: $d = d_p^F$ (for some F and p).

```

clustering FUZZY- $k$ -MEDOIDS( $k$ , individuals, maxIterations)
input:  $k$ : required number of clusters;
         individuals: individuals to be clustered;
         maxIterations: maximum number of iterations;
output: clustering: set of clusters
begin
Initialize iteration  $\leftarrow 0$ , random prototypes  $M = \{\mu_j\}_{j=1}^k$ 
Initialize uniform probabilities  $P(C_j|x_i)$ , for  $i = 1, \dots, N, j = 1, \dots, k$ 
repeat
  For each  $a \in$  individuals:
     $t \leftarrow \operatorname{argmin}_{j=1, \dots, k} d(a, \mu_j)$ 
     $C_t \leftarrow C_t \cup \{a\}$ 
  re-compute prototypes  $M = \{\mu_j\}_{j=1}^k$  according to eq. (4)
  re-compute all probabilities  $P(C_j|x_i)$ , using eq. (1)
  normalize the probabilities, for  $i = 1, \dots, N$ 
  ++iteration
until convergence or iteration = maxIterations
return  $\{C_j\}_{j=1, \dots, k}$ 
end

```

Fig. 1. The fuzzy clustering algorithm for categorical metric spaces

Fig. 1 reports a sketch of the FUZZY k -MEDOIDS algorithm. Note that the algorithm requires the number of clusters k as a parameter.

The representation of centers through medoids has two advantages. First, it presents no limitations on attributes types, and, second, the choice of medoids is dictated by the location of a predominant fraction of points inside a cluster and, therefore, it is less sensitive to the presence of outliers. This robustness is particularly important in the common context that many elements do not belong exactly to any cluster, which may be the case of the membership in DL knowledge bases, which may be not ascertained given the OWA. Algorithms where prototypes are represented by centroids, which are weighted averages of points within a cluster work conveniently only with numerical attributes and can be negatively affected even by a single outlier. An algorithm based on medoids allows for a more flexible definition of similarity. Many clustering algorithms work only after transforming symbolic into numeric attributes.

4 Evaluation

The clustering algorithm has been evaluated with an experimentation on various knowledge bases selected from standard repositories. The option of randomly generating assertions for artificial individuals was discarded for it might have biased the procedure. Only populated ontologies (which may be more difficult to find) were suitable for the experimentation.

A number of different knowledge bases represented in OWL were selected from various sources (the Protégé library² and the Swoogle³ search engine were used), namely: FINITESTATEMACHINES (FSM), SURFACEWATERMODEL (SWM), TRANSPORTATION, WINE, NEWTESTAMENTNAMES (NTN), FINANCIAL, the BioPax glycolysis ontology (BIOPAX), and one of the ontologies generated by the Lehigh University Benchmark (LUBM). Table 1 summarizes salient figures concerning these ontologies.

² <http://protege.stanford.edu/plugins/owl/owl-library>

³ <http://swoogle.umbc.edu>

Table 1. Ontologies employed in the experiments

Ontology	DL language	#concepts	#object props.	#data props.	#individuals
FSM	<i>SO\mathcal{F}(D)</i>	20	10	7	37
SWM	<i>ALCO\mathcal{F}(D)</i>	19	9	1	115
LUBM	<i>ALR$_{+}$HI(D)</i>	43	7	25	118
WINES	<i>ALC\mathcal{I}O(D)</i>	112	9	10	149
BioPAX	<i>ALC\mathcal{I}F(D)</i>	74	70	40	323
NTN	<i>SH\mathcal{L}F(D)</i>	47	27	8	676
FINANCIAL	<i>ALC\mathcal{I}F</i>	60	16	0	1000

In the computation of the proximity matrix (the most time-consuming operation) all named concepts in the knowledge base have been used for the committee of features, thus guaranteeing meaningful measures with high redundancy. The squared version of the measure has been adopted (d_2^F) with uniform weights. The PELLET reasoner⁴ (ver. 2.0rc4) was employed to perform the inferences that were necessary to compute the proximity matrices. The experimentation consisted of 50 runs of the algorithm per knowledge base. Each run took from a 1 to 5 mins on a QuadCore (2Gb RAM) Linux box, depending on the specific ontology processed. The indices which were chosen for the experimental evaluation of the outcomes were the following: an alternative R-Squared index (ranging in $[0, 1]$) [13] adopting medoids as cluster centers, Hubert's normalized Γ index [13] and the average Silhouette index [10], both ranging in $[-1, 1]$, with 1 indicating the best performance. We also considered the average number of clusters resulting from the runs on each knowledge base. It is also interesting to compare this number to the one of the primitive and defined concepts in each ontology (see Table 1, rightmost column).

For a comparison w.r.t. a different (stochastic) clustering procedure which is applicable to the same datasets employed in previous works (tables can be found in [8]), we opted for the average number of clusters found during the runs of the algorithm. Table 2 reports the average outcomes of these experiments. The table shows that the algorithm is quite stable in terms of all indices, as testified by the low variance of the results, despite its inherent randomized nature. As such, the optimization procedure does not seem to suffer from being caught in local minima.

Hubert's normalized Γ index measures both compactness and separation of the resulting clusters w.r.t. the proximity matrix. Results are generally good for the various ontologies. The R-Squared average values denote a good degree of separation between the various clusters. We may interpret the outcomes observing that clusters present a high degree of compactness. It should also be pointed out that flat clustering penalizes separation as the concepts in the knowledge base are seldom declared to be disjoint. Rather, they naturally tend to form subsumption hierarchies. As for the average Silhouette index the performance of the algorithm is generally very good with a slight degradation with the increase of individuals taken into account. Besides, note that the largest knowledge base (in terms of its population) is also the one with the maximal number of concepts which provided the features for the metric. Surprisingly, the number of clusters is limited w.r.t. the number of concepts in the KB, suggesting that many individuals gather around a restricted subset of the concepts, while the others are only

⁴ <http://clarkparsia.com/pellet>

Table 2. Results of the experiments with the FUZZY k -MEDOIDS algorithm

Ontology	Hubert's I^*	R-Squared	Silhouette	#clusters
FSM	.51 ($\pm 8.29e-2$) [.39,.72]	.81 ($\pm 4.98e-2$) [.74,.92]	.81 ($\pm 3.64e-2$) [.74,.89]	13
SWM	.77 ($\pm 4.99e-2$) [.63,.73]	.85 ($\pm 1.74e-2$) [.81,.89]	.88 ($\pm 6.49e-2$) [.81,.95]	14
LUBM	.60 ($\pm 9.14e-2$) [.48,.75]	.51 ($\pm 1.09e-1$) [.31,.69]	.85 ($\pm 2.05e-2$) [.75,.90]	12
WINE	.32 ($\pm 4.30e-2$) [.26,.41]	.98 ($\pm 6.56e-4$) [.982,.985]	.88 ($\pm 1.42e-2$) [.84,.90]	78
BIO-PAX	.59 ($\pm 7.77e-4$) [.45,.77]	.62 ($\pm 7.00e-2$) [.45,.78]	.88 ($\pm 1.46e-2$) [.85,.92]	16
NTN	.86 ($\pm 2.00e-2$) [.76,.88]	.83 ($\pm 3.35e-2$) [.65,.88]	.93 ($\pm 1.77e-2$) [.90,.95]	35
FINANCIAL	.44 ($\pm 1.36e-2$) [.42,.46]	.46 ($\pm 2.26e-2$) [.43,.45]	.89 ($\pm 3.26e-2$) [.85,.92]	27

complementary (they can be used to discern the various individuals). Such subgroups may be detected extending our method to perform hierarchical clustering.

5 Hierarchical Clustering

Some natural extensions may be foreseen for the presented algorithm. One regards upgrading the algorithm so that it may build *hierarchical* clusterings levelwise in order to produce (or reproduce) terminologies possibly introducing new concepts elicited from the ontology population. Hierarchical clustering methods may adopt agglomerative (*clumping*) or divisive (*splitting*) approaches and usually require distance functions for calculating distance between clusters.

Given the algorithm presented in section 4, it appears natural to focus on divisive methods. Whereas agglomerative clustering begins with each element a cluster and then combines clusters using a distance measure, divisive hierarchical clustering begins with one cluster and then continually breaks these clusters into smaller and smaller clusters until a stopping criterion is met (no quality improvement or singleton clusters reached). The clusters at each level are examined and the one containing objects that are the farthestmost according to the metric are broken apart.

The hierarchical extension of the algorithm implements a divisive method, starting with one universal cluster grouping all instances. Iteratively, it creates new clusters by applying the FUZZY k -MEDOIDS to the worst cluster and this continues until a stopping criterion is met, so that finally a *dendrogram* is produced. Fig. 2 reports a sketch of the algorithm. It essentially consists of a loop that computes a new level of the dendrogram until the stopping criterion is met; the inner call to FUZZY k -MEDOIDS returns a clustering of one cluster at the current level.

On each level, the worst cluster is selected (call to the SELECTWORSTCLUSTER function) on the grounds of its quality, e.g. the one endowed with the least average inner similarity (or cohesiveness). This cluster is candidate to being split. The partition is constructed by calling FUZZY k -MEDOIDS on the worst cluster (worstCluster). In the end, the candidate cluster is replaced by the newly found parts at the next level of the dendrogram (call to the REPLACE function).

```

clusterVector HIERARCHICALFCM(allIndividuals, params)
input allIndividuals: list of individuals
        params: other parameters for FUZZY-k-MEDOIDS
output clusterVector: array of lists of clusters

begin
level ← 0;
clusterVector[1] ← allIndividuals;
repeat
  level ← level + 1;
  worstCluster ← SELECTWORSTCLUSTER(clusterVector[level]);
  newClusters ← FUZZY-k-MEDOIDS(worstCluster,params);
  clusterVector[level+1] ← REPLACE(worstCluster,newClusters,clusterVector[level]);
until STOPCRITERION(clusterVector[level+1]);
return clusterVector
end

```

Fig. 2. The HIERARCHICAL FUZZY *k*-MEDOIDS algorithm

Two criteria have not been entirely specified: the function that determines the cluster quality and the stopping condition. As regards the cluster quality, there is a plethora of choices in the literature [14, 9]. Some of these functions have been recalled in section 4 for they determine validity measures. For example, the extent of a cluster diameter can be considered as a criterion for deciding which cluster has to be split. Alternatively, one may consider all clusters as candidates to the split, perform them and then evaluate the resulting new clustering level using the validity measures referred above. Although more computationally costly this may allow considering inter-cluster separation in the splitting criterion. As concerns the stopping criterion one may simply consider a maximum number of clusters to be produced. Again, a more costly way to determine the criterion would involve an evaluation of the gain yielded by a further level $l + 1$, in terms of the validity measure of choice (vm): $gain(l + 1) = vm(\text{clustering}[l + 1]) - vm(\text{clustering}[l])$. An insufficient or even negative improvement of the clustering quality (e.g. w.r.t. some given threshold) may determine the halting condition for the algorithm.

Alternative divisive methods based on hierarchical extensions of the PARTITION-AROUND-MEDOIDS algorithm have been considered [7].

6 Concluding Remarks

This work has presented a framework for fuzzy clustering that can be applied to standard multi-relational representations adopted for knowledge bases in the SW context. Its intended usage is for discovering interesting groupings of semantically annotated resources and can be applied to a wide range of concept languages. Besides, the induction of new concepts may follow from such clusters [7], which allows for accounting for them from an intensional viewpoint. In this paper we have also presented a possible extension to producing hierarchical clustering. A further natural extension of the clustering algorithm is towards incrementality.

The method exploits a dissimilarity measure that is based on the underlying resource semantics w.r.t. a number of dimensions corresponding to a committee of features represented by a group of concept descriptions in the language of choice. The algorithm is an extension of distance-based clustering procedures employing medoids as cluster

prototypes so to deal with complex representations of the target context. The distance measure may also serve as a ranking criterion.

As regards the optimization of the pseudo-metric, a promising research line, for extensions to matchmaking, retrieval and classification, is *retrieval by analogy* [11]: a search query may be issued by means of prototypical resources; answers may be retrieved based on local models (intensional concept descriptions) for the prototype constructed (on the fly) based on the most similar resources (w.r.t. some similarity measure). The presented algorithm may be the basis for the model construction activity.

References

- [1] d'Amato, C., Fanizzi, N., Esposito, F.: Analogical reasoning in description logics. In: da Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) URSW 2005 - 2007. LNCS (LNAI), vol. 5327, pp. 330–347. Springer, Heidelberg (2008)
- [2] Kirsten, M., Wrobel, S.: Relational distance-based clustering. In: Page, D.L. (ed.) ILP 1998. LNCS, vol. 1446, pp. 261–270. Springer, Heidelberg (1998)
- [3] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook. Cambridge University Press, Cambridge (2003)
- [4] Kietz, J.-U., Morik, K.: A polynomial approach to the constructive induction of structural knowledge. *Machine Learning* 14(2), 193–218 (1994)
- [5] Fanizzi, N., Iannone, L., Palmisano, I., Semeraro, G.: Concept formation in expressive description logics. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 99–110. Springer, Heidelberg (2004)
- [6] Duda, R., Hart, P., Stork, D.: *Pattern Classification*, 2nd edn. Wiley, Chichester (2001)
- [7] Fanizzi, N., d'Amato, C., Esposito, F.: Conceptual clustering for concept drift and novelty detection. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 318–332. Springer, Heidelberg (2008)
- [8] Fanizzi, N., d'Amato, C., Esposito, F.: Evolutionary conceptual clustering based on induced pseudo-metrics. *Semantic Web Information Systems* 4(3), 44–67 (2008)
- [9] Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. *ACM Computing Surveys* 31(3), 264–323 (1999)
- [10] Kaufman, L., Rousseeuw, P.J.: *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley & Sons, Chichester (1990)
- [11] Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning*. In: *Data Mining, Inference, and Prediction*. Springer, Heidelberg (2001)
- [12] Kramer, S., Lavrač, N., Flach, P.: Propositionalization approaches to relational data mining. In: Džeroski, S., Lavrač, N. (eds.) *Relational Data Mining*, pp. 262–286. Springer, Heidelberg (2001)
- [13] Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. *Journal of Intelligent Information Systems* 17(2-3), 107–145 (2001)
- [14] Bezdek, J.C., Pal, N.R.: Some new indexes of cluster validity. *IEEE Transactions on Systems, Man, and Cybernetics* 28(3), 301–315 (1998)

Reasoning about Relations with Dependent Types: Application to Context-Aware Applications

Richard Dapoigny and Patrick Barlatier

Université de Savoie, Laboratoire d'Informatique, Systèmes,
Traitement de l'Information et de la Connaissance

Po. Box 80439,

74944 Annecy-le-vieux cedex, France

Tel.: +33 450 096529; Fax: +33 450 096559

`richard.dapoigny@univ-savoie.fr`

Abstract. Generally, ontological relations are modeled using fragments of first order logic (FOL) and difficulties arise when meta-reasoning is done over ontological properties, leading to reason outside the logic. Moreover, when such systems are used to reason about knowledge and meta-knowledge, classical languages are not able to cope with different levels of abstraction in a clear and simple way. In order to address these problems, we suggest a formal framework using a dependent (higher order) type theory. It maximizes the expressiveness while preserving decidability of type checking and results in a coherent theory. Two examples of meta-reasoning with transitivity and distributivity and a case study illustrate this approach.

1 Introduction

A core problem in modeling knowledge-based intelligent systems is the ability to reason with and about relations. In [4], the authors develop a theory of part-hood, component hood, and containment relations in first order predicate logic and then discuss how description logics (DL) can be used to capture some aspects of the first order theory. They conclude that DL are not appropriate for formulating complex interrelations between relations. Moreover, the limited expressive power of FOL might result in reasoning problems that are known to be undecidable [1]. A possible solution consists in applying techniques of theorem proving that use higher-order knowledge about (or abstractions of) reasoning processes. This paper exploits an extension of type theory that treats meta reasoning in intelligent systems in a unified and mathematically clear manner. For that purpose we have investigated a constructive type theory and extended it with sub-typing and constant rules [2]. The constructive type theory relies on the Curry-Howard isomorphism giving us a way to capture the syntax-semantics interface in a clear way. It defines a formal system designed to act both as a proof system (constructive logic) and as a typed functional programming language (typed λ -calculus). The expressive power of a typed λ -calculus is thus

limited by its logic, but the positive counterpart is that the computation can be controlled by the logic. The main benefit for conceptual modeling is that the resulting language provides a common language both for modeling and for data processing. Due to space constraints, we restrict the analysis to the representation and reasoning about some part-whole relations.

2 The Dependent Type Framework

2.1 Overview of DTF

The logical background rooted in the Extended Calculus of Constructions (ECC) [11], uses a constructive type theory including a typing mechanism with polymorphism and functional dependency. The Dependent Type Framework (DTF) extends ECC with some useful rules whose basis are detailed in [6]. In this paper, we will demonstrate how relations and “meta” relations can be specified from ontological knowledge within DTF.

Type theory has explicit proof objects which are terms in an extension of the typed lambda-calculus while at the same time, provability corresponds to type inhabitation¹. Proofs in a computational setting can be the result of a database lookup, the existence of a function performing a given action or the result of a theorem prover given assumption about entities, properties or constraints. The expression $a : T$ is called a typing judgment and classifies an object a as being of type T . We call a an inhabitant of T , and we call T the type of a . Information states are formalized as sequences of judgments, built up according to a certain set of rules. Some types, that are always considered well-formed, are introduced by means of axioms. These special types are usually called sorts. ECC exhibits three levels of stratification, a sort level for universes, a type level and an object level (proofs).

The theory comprises both an impredicative universe noted *Prop* for logic (impredicative universe gives the system the expressiveness of intuitionist higher-order logic) and an infinite hierarchy of predicative type universes for data types denoted $Type_0, Type_1, \dots$. The hierarchy is cumulative, that is, $Type_i$ is contained in $Type_{i+1}$ for every i . We have objects of type 0, for individuals, objects of type 1, for classes of individuals, objects of type 2, for classes of classes of individuals, and so on.

Dependent type theories [8,5,13], introduce dependent types as types expressed in terms of data whereas their inhabitants are explicitly related to that data. A salient component of DTF is the dependent sum (or Σ -type) that models pairs in which the second component depends on the first. We will use $\langle a, b \rangle$ to denote pairs, which can be nested to describe more complex properties. Let us consider the dependent sum $\Sigma x : phone.mobile(x)$. A proof for that sum type is given for example by the instance $\langle MyNokia6125, q_1 \rangle$ indicating that for the individual *MyNokia6125*, the proposition is proved (q_1 is a proof of

¹ A proposition is true iff its set of proofs is inhabited.

$mobile(MyNokia6125)$). If we think of the set \mathcal{B} of all phones, then the proved pairs express a subset of \mathcal{B} , the subset of mobile phones such that:

$$\langle MyNokia6125, q_1 \rangle : \Sigma x \quad : \quad phone.mobile(x)$$

In addition, two projection operators π_1 and π_2 such that given a pair $\langle M, N \rangle$, we have $\pi_1 \langle M, N \rangle = M$ and $\pi_2 \langle M, N \rangle = N$. A proof $s : \Sigma x : T.P$ in a sum is a pair $s = \langle \pi_1 s, \pi_2 s \rangle$ that consists of an element $\pi_1 s : T$ of the domain type T together with a proof $\pi_2 s : P[\pi_1 s/x]$ stating that the proposition p is true for this element $\pi_1 s$.

2.2 Specification of Meta-properties

In what follows, the notation $\sum[x_1 : T_1, x_2 : T_2, \dots, x_n : T_n]$ will stand for the Σ -type $\Sigma x_1 : T_1. \Sigma x_2 : T_2. \dots. T_n$. A striking feature of type theory is that it allows for the so-called “meta” reasoning but without leaving the internal logic. For that purpose, a specification of some data structure is provided as the left member of a Σ -type, whereas the right member introduces the properties that the structure is supposed to fulfill.

Definition 1 (*Specification*). A specification S in DTF consists of a pair whose objects are *i*) available proofs that realize the specification of a structure $Struct[S]$ and *ii*) a predicate $Ax[S]$ over $Struc[S]$ specifying the properties that the realization of the specification should satisfy:

$$S \triangleq \sum [Struc : Type, Pr : Struc \rightarrow Prop]$$

In such a way, the computational contents (the structure type of the specification) is separated from the axiomatic requirements (proofs of correctness). If the structure exists (if we get some proof of it) and if the properties are fulfilled for that structure, then that structure satisfies the constraints given in Pr . Any binary relation can be expressed through the Σ -type:

$$Rel \triangleq \Sigma x : Type . \Sigma y : Type . R(x, y) \quad (1)$$

Expressing Transitivity. Then if we are for instance, interested in the specification of transitive relations, the following structure could be introduced:

$$Struc[Tr] \triangleq \sum \left[\begin{array}{l} Tr \quad : \quad Rel \\ Transitive : Rel \rightarrow Prop \end{array} \right]$$

and for any relation r of type $Struc[Tr]$ (where Tr abbreviates $Tr[r]$) :

$$\begin{aligned} & \forall u, u' : Tr . \\ Pr[Tr] \triangleq & (R_u = R_{u'} : Prop \ \& \ \pi_1 \pi_2 u = \pi_1 u' : Type) \rightarrow \\ & (R_u(\pi_1 u, \pi_1 \pi_2 u) \ \& \ R_{u'}(\pi_1 u', \pi_1 \pi_2 u') \rightarrow R_u(\pi_1 u, \pi_1 \pi_2 u')) \end{aligned}$$

with $R_u \triangleq \pi_2 \pi_2 u$ and $R_{u'} \triangleq \pi_2 \pi_2 u'$. The axiom Pr states that if the propositions in Rel structures are identical ($R_u = R_{u'}$) and if the relation is applied twice

with the second argument of the first relation being equal to the first argument of the second one, then the relation applies between the first argument of R_u and the second argument of $R_{u'}$. In other words, if we get a proof that a relation R is transitive (e.g., reading this information from a table), then a proof that the structure $Struc[Tr]$ exists, and then any relation of that type must satisfy the axioms of $Pr[Tr]$ in order for the specification to be fulfilled. A significant property of that mechanism is that a given specification can be extended and re-used in further specifications. This aspect is crucial for applying specifications to ontologies.

Expressing Distributivity. Another interesting case is that of the left- and right-distributing properties extensional relations such as the *part – whole* one. Above this assumption, the distributivity means that an intentional relation may distribute its related predicate to the parts of a whole. Let us consider collections as aggregates of individuals called members of the collection. The distributivity operates on the *has – part* relation². If a relation is left-distributive over a partonomic relation, then the relation which holds for the whole is also proved for the parts, i.e., more formally, the following structure holds provided that any relation DR is left-distributive with respect to the relation DR' (e.g., *has – part*):

$$Struc[DR, DR'] \triangleq \sum \begin{array}{l} DR \quad : Rel \\ DR' \quad : Rel \\ L - Distrib : Rel \rightarrow Rel \rightarrow Prop \end{array}$$

and for any pair of relations r, r' of type $Struc[DR, DR']$ (with DR, DR' the abbreviations for $DR[r], DR'[r']$):

$$Pr[DR, DR'] \triangleq \begin{array}{l} \forall u : DR, \forall u' : DR' . \\ (\pi_2\pi_2u = \pi_2\pi_2u' \rightarrow \perp \ \& \ \pi_1u = \pi_1u' : Type) \rightarrow \\ (R_u(\pi_1\pi_1u, \pi_1\pi_2u) \ \& \ R_{u'}(\pi_1\pi_1u', \pi_1\pi_2u') \rightarrow \\ R_u(\pi_1\pi_2u', \pi_1\pi_2u)) \end{array}$$

with $R_u \triangleq \pi_2\pi_2u$ and $R_{u'} \triangleq \pi_2\pi_2u'$. The axiom Pr says that, provided that each of the propositions corresponding to the relations DR and DR' are distinct ($R_u = R_{u'} \rightarrow \perp$), and that the first argument of the first relation is identical to the first argument of the second one, then the relation R_u is valid, having as respective arguments, the first argument of R_u and the second argument of $R_{u'}$. If we get a proof for the distributivity of the relation DR with respect to DR' , that is, a proof of $Struc[DR, DR']$, then any pair of relations of that type must satisfy the axioms in $Pr[DR, DR']$ in order to prove the specification.

3 Knowledge Representation with Dependent Types

3.1 Representing Ontological Concepts and Relations

Assuming that there exists a mapping between an ontology and a Type Theory, a concept hierarchy (e.g., a subsumption hierarchy) corresponds to a hierarchy

² Also called *has – item*, *has – participant*, *has – element*, ...

of types, that assigns each entity to a type. The type of an inhabitant constrains the types of other objects that we can combine with it and specifies the type of such a combination. Therefore, we can see a conceptualization as given by a set of types together with their constructors.

All simple types (i.e., non-dependent types) express *universals* in the ontological sense, while their related proof objects are individuals. They can be extracted from ontologies.

Definition 2. *In DTF any universal of the domain under consideration is represented as a non dependent type. For any basic object (individual), there exists a universal such that this object is a proof for it.*

Definition 3. *All properties of universals are captured by Σ -types.*

For instance, the universal “house” in $x : house$ has the proof (is instanced by): $x = MyHouse$. If one is interested in representing the property “LowCost” for houses, the corresponding Σ -type should be: $\Sigma x : house . LowCost(x)$. Complex types are described through Σ -types and express predicates. Predicates can be negated, stating that the referenced concepts are explicitly not in the relationship.

Definition 4. *An association between universals is formalized as a relation between these universals whose extension consists of all the proofs for that relation.*

Any n-ary relation can be expressed with a Σ -type. For example, the association *purchaseFrom* introduced in [7] within the scope of the General Ontological Language (GOL), relates three individuals, i.e., a person, an individual good and a shop. We get easily the corresponding DTF definition:

$$\Sigma x : Person . \Sigma y : Good . \Sigma z : Shop . purchaseFrom(x, y, z)$$

A proof for that ternary relation could be the tuple $\langle John, \langle PartsbySimons, Amazon, p_1 \rangle \rangle$ with $p_1 = purchaseFrom(John, PartsbySimons, Amazon)$ provided that this association exists in the database. Types are also able to constrain the scope of individuals that appear within relations. For example, *contained_in* does have its domain constrained to physical objects with the definition:

$$\Sigma x : Physical_object . \Sigma y : Physical_object . contained_in(x, y)$$

It means that only values of type *Physical_object*, or their sub-types in the hierarchy, are available. Of course, such a subtype specification requires commitment to a foundational ontology to ensure unambiguous definitions.

Pre-defined values can be introduced with the *Intensional Manifest Fields* (IMF) [12]. For example, with a maximum distance value d_m that is equal to 3, one can define a Σ -type comparing a distance value with that constant, seen as a unit type³ through the definition:

$$\sigma = \Sigma d : distance . \Sigma d_m \sim 3 : distance LowerThan(d, d_m)$$

³ The constant type.

which means that d_m is an IMF of the type *distance* and witnesses for a maximum value set to 3 miles.

DTF can be used to formalize the syntax of an ontology by defining rules of inference as inductive relations. Inferences are modeled with Π -types. For instance, the dependent product:

$$\Pi x : Phone . sendCalls(x)$$

in which $sendCalls(x)$ is a predicate that depends on x , represents the fact that any phone, e.g. *MyNokia6125*, is able to send calls. A proof of the Π -type is given by the β -reduction: $\lambda x : A.sendCalls(x)(MyNokia6125)$ that is, $sendCalls(MyNokia6125)$. In other words, $sendCalls$ is a predicate (i.e., a function $Phone \rightarrow Prop$) which for each instance x (e.g., *MyNokia6125*) of the type *Phone* yields a proof object for the proposition (e.g., $sendCalls(MyNokia6125)$). Since all phones send calls, it expresses the universal quantification \forall .

3.2 Representing Ontological “Meta” Properties

Due to its importance in ontological modeling, we focus on the representation of part-whole relations [10,16]. The existence of several types of part-whole relations, based on the type on ontological entities they relate, is extended by a number of meta-properties that part-whole can possess. Deriving implied relations, derived relations (transitivity), and satisfiability can aid correcting large conceptual models. For that purpose, specifications formalized in DTF enable reliable (automated type checking) and incremental (knowledge refinement) type structures. There is a need to define how important properties related to the part-whole relation such as transitivity and distributivity are expressed in DTF.

Transitivity. The dependent type theory can express transitivity as a property that depends on values corresponding to the different ways in which parts contribute to the structure of the whole. A typical example could be given with the spatial (or temporal) part-of relation that is, for these versions, transitive. A proof of $Struc[Tr](part - of)$ is given by checking the pair $\langle part - of, q_1 \rangle$ with q_1 a proof of $Transitive(part - of)$ ⁴. Since the *part - of* relation is transitive, let us consider that the terms u and u' from a knowledge base have the following contents:

$$\begin{aligned} u &: \Sigma x : soldier. \Sigma y : section. Part - of(x, y) \\ u' &: \Sigma x : section. \Sigma y : platoon. Part - of(x, y) \end{aligned}$$

If we obtain from a database, the respective proofs for the above relations: $\langle Paul, \langle sec35, p_1 \rangle \rangle$ and $\langle sec35, \langle P8, p_2 \rangle \rangle$ with p_1 and p_2 the respective proofs of $part - of(Paul, sec35)$ and $part - of(sec35, P8)$. From axiom $Pr[Tr](part - of)$, the premises are proved ($R_u = R_{u'} = part - of$ and

⁴ This kind of knowledge is needed in order to exploit re-usability and (meta)-reasoning.

$\pi_1\pi_2u = \pi_1u' = \text{sec35} : \text{section}$) then it yields a proof for $\text{part} - \text{of}(\text{Paul}, P8)$ since we have simultaneously $R_u(\pi_1u, \pi_1\pi_2u)$ (proof of $\text{part} - \text{of}(\text{Paul}, \text{sec35})$) and $R_{u'}(\pi_1u', \pi_1\pi_2u')$ (proof of $\text{part} - \text{of}(\text{sec35}, P8)$). In summary, with dependent types, transitivity is expressed like a property that depends on a value related to the different ways in which the components contribute to the whole's structure.

Distributivity. Predicates acting upon collections apply upon the articles that compose the collection. Let us show the expressive power of this property with some example. For instance, we may capture the fact, that the objectives of a group are the same as those of a member of the group. A proof of $\text{Struc}[DR, DR']$ ($\text{has_objective}, \text{has_part}$) is given according to the checking the nested pair $\langle \text{has_objective}, \langle \text{has_part}, q_1 \rangle \rangle$ with q_1 a proof of $L - \text{Distrib}(\text{has_objective}, \text{has_part})$. Provided that the relation has_objective is left-distributive over has_part , let us suppose that the knowledge base contains the terms u and u' with:

$$\begin{aligned} u &: \Sigma a : \text{association}. \Sigma b : \text{topic}. \text{has_objective}(a, b) \\ u' &: \Sigma x : \text{association}. \Sigma y : \text{person}. \text{has_Part}(x, y). \end{aligned}$$

Then, assuming the respective proofs that state i) the difference between propositions ($\text{has_objective} = \text{has_Part}$ is an absurd judgment) and ii) the identification of the head arguments (association is the common argument), the respective proofs of R_u and $R_{u'}$, e.g., $\text{has_objective}(\text{ACM} - \text{SIGART}, \text{AI})$ and $\text{has_part}(\text{ACM} - \text{SIGART}, \text{Patrick})$, yield a proof for $\text{has_objective}(\text{Patrick}, \text{AI})$. A similar rule can be defined for right-distributive relations.

Subsumption reasoning is at the heart of description logics (DLs). However, in DL subsumption holds among concepts, which are understood as unary predicates [3]. In DTF, the subsumption relation may hold between quantified types (such as Π -types) or subset types (such as Σ -types). Unlike hybrid formalisms dependent types provide a single framework for reasoning about ontological concepts and relations. Furthermore, it is worth noticing that many approaches in intelligent systems combine OWL-DL (i.e., DL-based) reasoning with logical programming to overcome DL-based restrictions [15,9]. By contrast, the current work provides a unified framework in which reasoning can be made at different abstraction levels while preserving tractability [2].

4 A Case Study

This case study is extracted from a scenario defined in the AWARENESS project [14]. A person owns a calendar (e.g., on a PDA) in which there are a number of meetings he can attend. Each of the meetings is provided with a start and stop time. The meeting is held in a particular room and the room is part of a building that has a geographical location or at least, an address. The objective is to determine the location of a person based on the meeting information of that person, and more precisely:

1. Determine the meeting that the person currently attends.
2. Assuming a known location for the meeting room, infer the location of the person.
3. Determine the location of the meeting room.

A person is in a meeting if there is a meeting that is attended by that person, and for this meeting the start time is smaller or equal to the current time, and the stop time is larger or equal to the current time. While OWL is unable to support comparison operations and parameterized definition of classes, the Dependent Type Framework allows to define a context related to a given meeting (considered as a process). Pieces of knowledge can be easily be built with the following predicates:

$$\begin{aligned}\sigma_1 &= \Sigma t : Time. \Sigma t_s \sim "11/11/08 : 14.00" : Time . GreaterThan(t, t_s) \\ \sigma_2 &= \Sigma s : \sigma_1. \Sigma t_e \sim "11/11/08 : 16.30" : Time . LowerThan(\pi_1 s, t_e) \\ \sigma_3 &= \Sigma x : Person. \Sigma m : meeting . attends(x, m)\end{aligned}$$

Then if the meeting process requires a knowledge in which the current time is within the predefined temporal limits of the event and the concerned person participates in that event, this contextual knowledge can be easily described with the above Σ -types:

$$K_0 = \sigma_2 \times \sigma_3$$

Two constant values define respectively the start and the stop time of the meeting. The two propositions (resp. *GreaterThan* and *LowerThan*) are both proved iff the current time $t : Time$ is between the start and the stop time. Notice that if σ_2 is proved, it means both that the current time is greater than and lower than the respective predefined limits. Each meeting can exploit the same context type provided that the constant values correspond to the correct values.

$$\Pi k : K_0 . T_part_of(\pi_1 \pi_2 k, \pi_1 \pi_2 \pi_2 k) \quad (2)$$

According to inference in eq. 2, a proof of the context K_0 yields a proof for the relation $T_part_of()$, where T_part_of stands for the temporal *part_of* relation. More complex contexts can be composed by adding more information about the meeting such as, for example, topics of the meeting, etc. Assuming a known location for the meeting room, could we infer the location of the person? In order to derive the result, a careful analysis of the ontological knowledge is required. First, a sum type reflects the contextual⁵ relation about meeting and room:

$$\begin{aligned}\tau_1 &= \Sigma x : meeting . \Sigma y : room . hasLocation(x, y) \\ \tau_2 &= \Sigma x : meeting . \Sigma y : Person . hasParticipant(x, y)\end{aligned}$$

Then, if we see a meeting as a (temporal) collection of persons, and if we have in the knowledge base a proof for the structure $Struc[DR, DR'](hasLocation, hasParticipant)$, i.e. in that particular case $\langle hasLocation, \langle hasParticipant, r_1 \rangle$

⁵ The context of the user's activity.

with r_1 a proof of $L_distrib(hasLocation, hasParticipant)$ then, the left distributivity can be applied to the $hasParticipant$ relation. For instance, if the following proofs for τ_1 and τ_2 , are respectively $\langle Ontologies\ and\ the\ Web, \langle AulaB7, p_1 \rangle \rangle$, and $\langle Ontologies\ and\ the\ Web, \langle Richard, p_2 \rangle \rangle$ with p_1 and p_2 the respective proofs of $hasLocation(Ontologies\ and\ the\ Web, AulaB7)$ and $hasParticipant(Ontologies\ and\ the\ Web, Richard)$. Then, since the propositions in the relation structures are different and the first argument in Σ -types τ_1 and τ_2 are identical, then it yields a proof for $hasLocation(Richard, AulaB7)$.

An identical process applies to the question 3). The knowledge base may include the following relations:

$$\begin{aligned}\tau_3 &= \Sigma x : building . \Sigma y : address . hasLocation(x, y) \\ \tau_4 &= \Sigma x : building . \Sigma y : room . has - part(x, y)\end{aligned}$$

Then, we conceive a building as a (spatial) collection of rooms, and assuming the respective proofs for these relations (e.g., proofs $\langle G - 10, \langle "B - 1050, Brussel, Belgium", p_1 \rangle \rangle$ and $\langle G - 10, \langle AulaB7, p_2 \rangle \rangle$ where p_1 and p_2 stand respectively for the proofs $hasLocation(G - 10, "B - 1050, Brussel, Belgium")$ and $has - part(G - 10, AulaB7)$) we can apply again the left distributivity and get easily a proof for $hasLocation(AulaB7, "B - 1050, Brussel, Belgium")$. Furthermore, this proof together with the previous one (i.e., $hasLocation(Richard, AulaB7)$) are the proofs for the respective Σ -types:

$$\begin{aligned}\tau_5 &= \Sigma x : room . \Sigma y : address . hasLocation(x, y) \\ \tau_6 &= \Sigma x : Person . \Sigma y : room . hasLocation(x, y)\end{aligned}$$

Provided that the relation $hasLocation$ is transitive, and applying the corresponding specification, it is easy to get a proof for $hasLocation(Richard, "B - 1050, Brussel, Belgium")$. These examples witness for the expressiveness of the specifications using ontological knowledge.

5 Conclusion

The approach proposed in this paper is independent of any environment and is therefore applicable to (or adaptable by) most ontologies with the objective of addressing important topics such as: (i) the general notion of types and their instances; (ii) distinctions among sorts of relational properties; (iii) Part-whole relations and iv) evaluation of the ontological correctness of current conceptual representations produced using the language. First, using dependent types leads to a more precise and concise modeling. Second, DTF can be seen as a theory based on a logic of dependence in which dependencies between terms allow for modular and inter-dependent theories to be interconnected. Furthermore, it enables the conceptual modeler to gradually develop models that are closer to the real-world semantics and thereby improve quality of the software.

Type theory can be used to formalize the syntax of an ontology. This could be generalized by defining the ontology as a (structured) type and defining rules

of inference as inductive relations so as to give a computational understanding of the ontology. Semantics could then be provided by defining a function from the ontology (or its substructures) to propositions (objects of type *Prop*). In this way, type theory would allow for syntactic and semantic properties to be analyzed all within DTF. Further work will be investigated in that direction.

Within DTF, classical problems of “meta” reasoning on relations receive precise solutions, in a coherent framework. Many theoretical problems raised by the untyped formalisms have their roots in the strong assumption of a universe of bare individuals.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook. Cambridge University Press, Cambridge (2003)
2. Barlatier, P., Dapoigny, R.: A Theorem Prover with Dependent Types for Reasoning about Actions. In: Procs. of STAIRS 2008 at ECAI 2008, pp. 12–23 (2008)
3. Brachman, R.J., Levesque, H.J.: Knowledge Representation and Reasoning. In: Procs. of IJCAI 2004. Morgan Kaufmann, San Francisco (2004)
4. Bittner, T., Donnelly, M.: Computational ontologies of parthood, componenthood, and containment. In: Procs. of the Nineteenth International Joint Conference on Artificial Intelligence, pp. 382–387 (2005)
5. Coquand, T., Huet, G.: The calculus of constructions. *Information and Computation* 76(2-3), 95–120 (1988)
6. Dapoigny, R., Barlatier, P.: Towards a Conceptual Structure based on Type Theory. In: ICCS 2008, pp. 107–114 (2008)
7. Guizzardi, G., Herre, H., Wagner, G.: On the General Ontological Foundations of Conceptual Modeling. In: Spaccapietra, S., March, S.T., Kambayashi, Y. (eds.) ER 2002. LNCS, vol. 2503, pp. 65–78. Springer, Heidelberg (2002)
8. Harper, R., Honsell, F., Plotkin, G.: A framework for defining logics. *Journal of the Association for Computing Machinery* 40(1), 143–184 (1993)
9. Horrocks, I., Patel-Schneider, P.F.: A Proposal for an OWL Rules Language. In: Proc. of the 13th Int World Wide Web Conf. (WWW 2004). ACM, New York (2004)
10. Keet, C.M.: Part-Whole relations in Object-Role Models. In: 2nd Int. Workshop on Object Role Modelling (ORM 2006), Montpellier (2006)
11. Luo, Z.: A Unifying Theory of Dependent Types: The Schematic Approach. In: Procs. of Logical Foundations of Computer Science, pp. 293–304 (1992)
12. Luo, Z.: Manifest fields and module mechanisms in intensional type theory. In: Procs. of TYPES 2008 (2008)
13. Martin-Löf, P.: Constructive Mathematics and Computer Programming. *Logic, Methodology and Philosophy of Sciences* 6, 153–175 (1982)
14. Ferreira Pires, L., van Sinderen, M., Munthe-Kaas, E., Prokaev, S., Hutschemaekers, M., Plas, D.-J.: Techniques for describing and manipulating context information. Lucent Technologies, Freeband/A_MUSE, report D3.5v2.0 (2005)
15. Toninelli, A., Montanari, R., Kagal, A., Lassila, O.: A semantic Context-aware Access Control Framework for Secure Collaborations in Pervasive Computing Environments. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 473–486. Springer, Heidelberg (2006)
16. Varzi, A.C.: Parts, Wholes and Part-Whole Relations. *The Prospects of Mereotopology. Data and Knowledge Engineering* 20, 259–286 (1996)

Quasi-Classical Model Semantics for Logic Programs – A Paraconsistent Approach

Zhihu Zhang, Zuoquan Lin, and Shuang Ren

School of Mathematical Sciences,
Peking University,
Beijing, 100871, China
{zhzhang, lzq, shuangren}@is.pku.edu.cn

Abstract. We present a new paraconsistent approach to logic programming, called Quasi-classical (QC for short) model semantics. The basic idea is the following. We define the QC base as a set of all atoms and their complements, which decouples the link between an atom and its complement at the level of interpretation. Then we define QC models for positive logic programs. The QC model semantics actually effecting on disjunctive programs imposes the link between each disjunction occurring in the head of a rule and its complement disjunct. This enhances the ability of paraconsistent reasoning. We also define weak satisfaction to perform reasoning under our approach. The fixpoint semantics with respect to the QC model semantics is also presented in the paper.

Keywords: Quasi-classical logic, paraconsistent semantics, disjunctive logic program, fixpoint semantics.

1 Introduction

Logic programming especially disjunctive logic programming has been considered as a powerful tool for knowledge representation and automated reasoning [1]. Since classical semantics for logic programs can not deal with inconsistent information paraconsistently, the idea of the paraconsistent logics has been introduced to logic programming. Paraconsistent logics ([2,3,4]) in which a complementary pair a and $\neg a$, where a is an atom, does not deduce any arbitrary formula, have been considered as a fundamental theory for understanding human cognitive processes. One of paraconsistent logics namely four-valued logic [5] has first been introduced to the logic programming setting as bottom inference systems by Blair and Subrahmanian [6]. Later, various paraconsistent semantics [7,8,9,10] have been proposed for logic programs including disjunctive logic programs. Most of these paraconsistent semantics are based on Belnap's algebraic structure [5] since the four-valued structure is intuitive for representing paraconsistent inconsistent information. However, Disjunctive Syllogism [2] fails for multi-valued logics, i.e., $\frac{\alpha \vee \beta, \neg \alpha}{\beta}$ fails. Thus, it results in that these paraconsistent semantics are weak in deriving non-trivial classical conclusions.

The simplified form of the example [11] shows the aspect of weak reasoning that arises in using multi-valued logics in paraconsistent logic programming.

Example 1. *An intelligent agent has the knowledge base: $\{\text{lefthand} - \text{broken} \vee \text{righthand} - \text{broken} \leftarrow\}$ which means that one of my hands is broken, but not both. Once additional information about that my left hand is not broken is learned by the agent, then an additional rule $\{\neg \text{lefthand} - \text{broken} \leftarrow\}$ is added. Based on the current knowledge base, the intelligent agent based on four-valued logic will get two results $\{\text{lefthand} - \text{broken}, \neg \text{lefthand} - \text{broken}\}$ and $\{\text{righthand} - \text{broken}, \neg \text{lefthand} - \text{broken}\}$. Under the cautious reasoning¹, the agent will only entail that my left hand is not broken, and the question about which hand is broken is unknown to the agent. However, doing such paraconsistent reasoning is not satisfactory enough for an intelligent agent. When considering the brave reasoning, unfortunately, the agent may derive the inconsistent information about that my left hand is broken and my left hand is not broken, contrary to what we would expect.*

As illustrated above, most of the well-known paraconsistent semantics based on the four-valued structure do not behave properly in the presence of such classically consistent logic programs. Thus, we need to seek other paraconsistent approaches to enhance the ability of paraconsistent reasoning. Fortunately, Quasi-classical logic (QC for short) [12] provides a nice approach to resolve this problem. Hence, in this paper, we introduce the idea of QC logic to logic programming. In our approach, we first define the QC base as a set of all atoms and their complement, which decouples the link between an atom and its complement at the level of model. By introducing the QC base, we actually define a basis for paraconsistent reasoning. As we will see in the later part of this paper, the intuition about the interpretation coincides with four-valued logic [5]. However, we do not follow the philosophy of four-valued logic. We define QC models for positive logic programs. The QC model semantics actually effecting on disjunctive programs impose the link between each disjunction occurring in the head of a rule and its complementary disjuncts, which makes our semantics behave differently while comparing to paraconsistent semantics for logic programs such as paraconsistent stable model semantics [13]. By imposing the link, more non-trivial classical conclusions are obtained under our semantics. In this paper, we also present the corresponding fixpoint semantics to characterize the QC model semantics.

The rest of paper is organized as follows: the next section briefly reviews the background. In Section 3, we define QC models and minimal QC models for positive extended disjunctive programs. Then, a fixpoint semantics is proposed to characterize the QC model semantics. In Section 4, based on the definition of weak satisfaction, we define two main entailment problems under our semantics. Section 5 briefly discusses its relationship with the related work including the answer set semantics [14] and the paraconsistent minimal model semantics [13]. In the last section, we discuss the future work and conclude the paper.

¹ The definitions of cautious reasoning and brave reasoning can be referred to the paper [1].

2 Background

We assume that readers are familiar with the theory of the first order logic. A classical *literal* is either a *positive literal* a or *negative literal* $\neg a$, where a is an atom (\neg is the classical negation symbol).

Syntax. Given the first order language \mathcal{L} over the symbol Φ defined in this section, a positive extended disjunctive program [15] Π consists of the rules of the form:

$$r = a_1 \vee \dots \vee a_n \leftarrow b_1, \dots, b_k \quad (n \geq 0, k \geq 0)^2 \tag{1}$$

where $a_1, \dots, a_n, b_1, \dots, b_k$ are literals, a_i is also called a disjunct. The left-hand side of (1) is called the *head* of the rule, and the right-hand side of (1) is called the *body* of the rule. A rule of (1) can be classified as follows,

- a rule is called a fact if the body is empty;
- a rule is called an integrity constraint if the head is empty and the body is not empty;
- a rule is called a disjunctive rule if the head contains more than one literal;
- a rule is called a non-disjunctive rule if the head contains only one literal;

Then, a program is called an extended program if it contains no disjunctive rule; a program is called a disjunctive program if $a_1, \dots, a_n, b_1, \dots, b_k$ in all of its rules are atoms;

Semantics. An expression is called *ground* if it does not contain any variables. The *Herbrand base* of a program Π , denoted by HB_Π is the set of all ground literals of Π . By $ground(\Pi)$ we mean the grounding of Π . An answer set of a positive disjunctive program Π is any minimal³ subset I of HB_Π such that (1) for each rule in $ground(\Pi)$, if $b_1, \dots, b_k \in I$, then for some $i = 1, \dots, n, a_i \in I$; (2) if I contains a pair of complementary literals, then $I = HB_\Pi$. A positive extended disjunctive program is consistent if it has a consistent answer set, otherwise, it is inconsistent.

3 QC Semantics for Positive Extended Programs

3.1 Model Semantics

Quasi-classical (QC for short) logic has been developed to handle inconsistent information [12]. The essential idea of QC logic is using *focus* to constitute the basis of useful paraconsistent reasoning. The definition of *focus* is defined as follows,

Definition 1 [12]. *Let a_1, \dots, a_n be literals. The focus of $a_1 \vee \dots \vee a_n$ by a_i , denoted by $\otimes(a_1 \vee \dots \vee a_n, a_i)$, is defined as follows,*

$$\otimes(a_1 \vee \dots \vee a_n, a_i) = a_1 \vee \dots \vee a_{i-1} \vee a_{i+1} \vee \dots \vee a_n$$

² In [14], an extended disjunctive program use $|$ to denote \vee .

³ Minimality is defined in terms of set inclusion.

Based on the essential idea of QC logic, we introduce QC model semantics for positive extended disjunctive programs. For convenience, a program means a positive extended disjunctive program, and we also semantically identify a program with its ground program. For the reason that we introduce QC model semantics to handle inconsistency in programs, we first introduce the definition of QC base.

Definition 2. Let \mathcal{A}_Π be the set of all ground atoms used in a program Π . The QC base for Π (denoted by QCB_Π) is defined as follows, where the negative literal $\neg a$ is envisaged as a new atom.

$$QCB_\Pi = \mathcal{A}_\Pi \cup \{\neg a \mid a \in \mathcal{A}_\Pi\}.$$

\mathcal{I}_Π denotes the set of all subsets of QCB_Π . We call any $I \in \mathcal{I}_\Pi$ a QC interpretation. So, I can contain both a and $\neg a$ for some a .

The above definition of QC interpretation gives the following meaning with respect to literals: Let $a \in \mathcal{A}_\Pi$,

- $a \in I$ means that a is “satisfied” in I .
- $\neg a \in I$ means that $\neg a$ is “satisfied” in I .
- $a \notin I$ means that a is not “satisfied” in I .
- $\neg a \notin I$ means that $\neg a$ is not “satisfied” in I .

Accordingly, the above semantics can be regarded as giving one of the four truth values \top (both), t (true), f (false), \perp (unknown) to all ground atoms in \mathcal{A}_Π , as in Belnap’s four-valued logic [5]:

- a is \top if both a and $\neg a$ are “satisfied”;
- a is t if a is “satisfied” and $\neg a$ is not “satisfied”;
- a is f if a is not “satisfied” and $\neg a$ is “satisfied”;
- a is \perp if neither a nor $\neg a$ are “satisfied”;

Once this is stated, we indeed relax the link between an atom and its complement at the level of the model. In contrast, in the classical answer set semantics, if an answer set contains a pair of complementary literals, then it is restricted to be the set of all ground literals. So, in order to express the complement of a literal, we give the following definition.

Definition 3. Let a be an atom, and let \sim be a complementation operation such that $\sim a$ is $\neg a$ and $\sim(\neg a)$ is a . So, we say that two literals are complementary if one is the complementation operation of the other, such as l and $\sim l$.

The \sim operation is not in the object language, but it makes the following definitions clear.

Now, we are ready to inductively define satisfaction (denoted by \models) in logic programs as follows,

Definition 4. Let I be an interpretation and Π be a program. a_1, \dots, a_m are literals.

- (1) If $a \in \mathcal{A}_\Pi$, then $I \models a$ iff $a \in I$;
- (2) If $a \in \mathcal{A}_\Pi$, then $I \models \neg a$ iff $\neg a \in I$;
- (3) $I \models a_1 \vee \dots \vee a_n$ iff $I \models a_1$ or ... or $I \models a_n$ and $\forall i$ s.t. $1 \leq i \leq n$, $I \models \sim a_i$ implies $I \models \otimes(a_1 \vee \dots \vee a_n, a_i)$, where \otimes is defined in Definition 7 ;
- (4) $I \models a_1 \wedge \dots \wedge a_n$ iff $\forall i(1 \leq i \leq n)$, $I \models a_i$;
- (5) for any rule $r = F \leftarrow G$, $I \models r$ iff $I \models F$ or $I \not\models G$. In particular, $I \models \leftarrow G$ iff $I \not\models G$, and $I \models F \leftarrow$ iff $I \models F$.

An interpretation I is a QC model of a program Π if I satisfies every ground rule from Π . We use $QCM(\Pi)$ to denote the set of all QC models of a program Π . That is, $QCM(\Pi) = \{I \in \mathcal{I}_\Pi \mid I \models r, \text{ for every rule } r \in \Pi\}$. We say that a QC model I is inconsistent if I contains both a and $\sim a$ for some literal a , consistent otherwise.

It should be noted that this definition for disjunction is more restricted than the classical definition adopted by Gelfond and Lifschitz in [14]. For instance, if given a program $\{a \vee b \leftarrow, \neg a \leftarrow\}$, by the definition, $\{\neg a, b\}$ is a QC model of the program, while $\{\neg a, b, \neg b\}$ is not a QC model of the program. The reason that we need such restricted definition for disjunction is that, in order to provide a meaning for resolution that is a very powerful means for reasoning, we need to impose the link between each disjunction occurring in the head of a rule and its complement disjunct. As a result, to guarantee the satisfaction of disjunction, we need to guarantee the satisfaction of every more focused disjunction if necessary. In other words, we have the following theorem,

Proposition 1. Let Π be a logic program, $I \in QCM(\Pi)$ be an interpretation, and a_1, a_2 be literals. $I \models a_1 \vee a_2$ iff (1) $a_1 \in I$ and $\sim a_1 \notin I$ or (2) $a_2 \in I$ and $\sim a_2 \notin I$ or (3) $a_1 \in I$ and $\sim a_1 \in I$ and $a_2 \in I$ and $\sim a_2 \in I$.

Corollary 1 follows immediately from Proposition 1.

Corollary 1. Let Π be a logic program, $I \in QCM(\Pi)$ be an interpretation, and a_1, \dots, a_n be literals. $I \models a_1 \vee \dots \vee a_n$ iff (1) for some $a_i \in \{a_1, \dots, a_n\}$, $a_i \in I$ and $\sim a_i \notin I$ or (2) for all $a_i \in \{a_1, \dots, a_n\}$, $a_i \in I$ and $\sim a_i \in I$.

Since there are many QC models for a program, in order to catch the intended meaning of our programs, we define minimal models as follows,

Definition 5. Let Π be a program. Let $MQCM(\Pi) \subseteq QCM(\Pi)$ be the set of minimal QC models of Π , then,

$$MQCM(\Pi) = \{I \in QCM(\Pi) \mid \text{if } I' \subset I, \text{ then } I' \notin QCM(\Pi)\}.$$

Example 2. Reconsider Example 1 introduced in introduction section. The example can be formalized into a program $\Pi_1 = \{lh - broken \vee rh - broken \leftarrow, \neg lh - broken \leftarrow\}$. Then $MQCM(\Pi_1) = \{\{\neg lh - broken, rh - broken\}\}$.

Note that the QC model semantics is also suitable for positive extended logic programs although we consider positive extended disjunctive programs here.

When restricted to positive extended programs⁴, it is clear that the following theorem holds.

Theorem 1. *Let Π be a positive extended logic program. I is a minimal QC model of Π iff I is a \mathcal{M}_Π Model [7] iff I is a paraconsistent minimal model [13].*

Example 3. *Consider the classical birds example taken from [7]:*

$$\Pi_2 = \{ \text{flies}(X) \leftarrow \text{bird}(X), \neg \text{flies}(X) \leftarrow \text{penguin}(X), \\ \text{bird}(X) \leftarrow \text{penguin}(X), \text{penguin}(\text{tweety}) \leftarrow, \text{bird}(\text{fried}) \leftarrow \}$$

The \mathcal{M}_{Π_2} model, paraconsistent minimal model and minimal QC model are the same, namely $\{\text{flies}(\text{tweety}), \neg \text{flies}(\text{tweety}), \text{bird}(\text{tweety}), \text{penguin}(\text{tweety}), \text{bird}(\text{fried}), \text{flies}(\text{fried})\}$.

3.2 Fixpoint Semantics

In this section, we define the corresponding fixpoint semantics of positive extended disjunctive programs related to the minimal QC model semantics presented in the previous section. We first need introduce the definition of focusing disjunction by an interpretation⁵.

Definition 6. *Let a_1, \dots, a_n be literals and I be an interpretation. The focus of $a_1 \vee \dots \vee a_n$ by I , denoted by $f_{QC}(a_1 \vee \dots \vee a_n, I)$ is defined as follows,*

$$f_{QC}(a_1 \vee \dots \vee a_n, I) = \begin{cases} a_1 \vee \dots \vee a_n, & \text{if } I = \emptyset; \\ a_{l_1} \vee \dots \vee a_{l_i} \forall j \text{ s.t. } 1 \leq l_1 \leq j \leq l_i \leq n, \sim a_j \notin I, \text{ if } l_i \geq 1 \text{ and } I \neq \emptyset; \\ a_1 \wedge \dots \wedge a_n, & \text{otherwise.} \end{cases}$$

As characterized in Corollary [1], if the complements of all disjuncts in a disjunction belong to an interpretation I , then the disjuncts must be in I . Hence, we represent the idea of such constraint in the third condition of the above definition. For instance, if $a_1 \vee a_2$ is a disjunction, and $I = \{\neg a_1, \neg a_2\}$ where a_1, a_2 are literals, then $f_{QC}(a_1 \vee a_2, I) = a_1 \wedge a_2$.

The following operation *Lits* corresponding to Definition [6] is defined to classify the results of focusing a disjunction by an interpretation.

Definition 7. *Let a_1, \dots, a_n be literals, the operation *Lits* is defined as follows,*

$$\begin{aligned} \text{Lits}(a_1 \vee \dots \vee a_n) &= \{\{a_1\}, \dots, \{a_n\}\} \\ \text{Lits}(a_1 \wedge \dots \wedge a_n) &= \{\{a_1, \dots, a_n\}\} \end{aligned}$$

Based on Definition [6] and Definition [7], we are ready to give the fixpoint semantics. In order to describe non-deterministic property of disjunctive programs, we define the closure operator $\mathcal{T}_\Pi : \mathcal{I}_\Pi \rightarrow \mathcal{I}_\Pi$ as follows,

⁴ This kind of programs is also called definite extended logic programs.

⁵ Definition [6] and Definition [7] are defined below just for the convenience of describing the fixpoint semantics.

Definition 8. Let Π be a positive extended disjunctive program and \mathcal{I} be a set of interpretations, then the closure operator $\mathcal{T}_\Pi : \mathcal{I}_\Pi \rightarrow \mathcal{I}_\Pi$ is defined as,

$$\mathcal{T}_\Pi(\mathcal{I}) = \bigcup_{I \in \mathcal{I}} T_\Pi(I)$$

where the intermediate operator $T_\Pi : \mathcal{I}_\Pi \rightarrow \mathcal{I}_\Pi$ is defined as follows,

$$T_\Pi(I) = \begin{cases} \emptyset, & \text{if } \{b_1, \dots, b_k\} \subseteq I \text{ for some ground integrity constraint} \\ & \leftarrow b_1, \dots, b_k \text{ from } \Pi; \\ \{ J \mid \text{for each ground rule } r_i : a_{i_1} \vee \dots \vee a_{i_n} \leftarrow b_{i_1}, \dots, b_{i_k} \text{ such that} \\ & \{b_{i_1}, \dots, b_{i_k}\} \subseteq I, J = I \cup \bigcup_{r_i} J', \\ & \text{where } J' \in \text{Lits}(f_{QC}(a_{i_1} \vee \dots \vee a_{i_n}, I)) \}, & \text{otherwise.} \end{cases}$$

Hence, $T_\Pi(I)$ is the set of interpretations J 's such that for each rule C_i whose body is satisfied by I , I is expanded into J by adding one element from the classification of the focused version of head of every such C_i . Particularly, if I does not satisfy an integrity constraint from Π , I is removed from $T_\Pi(I)$.

Since we want to find the models of the program Π from the empty interpretation, we introduce the definition of the ordinal powers of \mathcal{T}_Π ,

Definition 9. The ordinal powers of \mathcal{T}_Π is inductively defined as follows,

$$\begin{aligned} \mathcal{T}_\Pi \uparrow 0 &= \{\emptyset\}, \\ \mathcal{T}_\Pi \uparrow n + 1 &= \mathcal{T}_\Pi(\mathcal{T}_\Pi \uparrow n), \\ \mathcal{T}_\Pi \uparrow \omega &= \bigcup_{\alpha < \omega} \bigcap_{\alpha \leq n < \omega} \mathcal{T}_\Pi \uparrow n. \end{aligned}$$

where n is a successor ordinal and ω is a limit ordinal.

The definition of the ordinal powers for zero and successor is canonically defined. This definition is also used by Lloyd [16], while the definition of limit ordinal is similar to that used in [13]. It states that we retain interpretations which are persistent in the preceding iterations. That is, for any interpretation I in $\mathcal{T}_\Pi \uparrow \omega$, there is an ordinal α smaller than ω such that, for every $n \in [\alpha, \omega)$, I is an element of $\mathcal{T}_\Pi \uparrow n$.

Theorem 2 (Fixpoint Theorem). $\mathcal{T}_\Pi \uparrow \omega$ is a fixpoint.

Example 4. Reconsider the program Π_1 in Example 2. It is easy to verify that $\mathcal{T}_{\Pi_1} \uparrow \omega = \mathcal{T}_{\Pi_1} \uparrow 2 = \{\{lh - broken, \neg lh - broken, rh - broken\}, \{\neg lh - broken, rh - broken\}\}$.

By the definition of fixpoint closure, it is clear that the fixpoint closure exists for any programs and is unique determined. Intuitively, by iterating from empty set of interpretation, we get the fixpoint closure. Indeed, the fixpoint closure contains what we need.

Lemma 1. Let Π be a positive extended disjunctive program. Then, an interpretation I is a QC model of Π iff $I \in \mathcal{T}_\Pi(\{I\})$.

Lemma 2. If I is a minimal QC model of Π , then for each literal a in I , there is a ground rule r in the set of such rules r_i of the form $r_i = a_1 \vee \dots \vee a_{n_i} \leftarrow b_1, \dots, b_{k_i}$ from Π that $\{b_1, \dots, b_{k_i}\} \subseteq I$, $a_i \in J$, where $J \in \text{Lits}(f_{QC}(a_1 \vee \dots \vee a_{n_i}, I))$ and $a = a_i$ for some i ($1 \leq i \leq n_i$).

From Lemma 2, It is clear to have the following result,

Corollary 2. *If I is a minimal QC model of Π , then for each literal a in I , there is a ground rule $a_1 \vee \dots \vee a_n \leftarrow b_1, \dots, b_k$ from $\Pi, \{b_1, \dots, b_k\} \subseteq I$ and $a = a_i$ for some i ($1 \leq i \leq n$).*

Next, we establish the equivalence between the fixpoint semantics and the model semantics of positive extended disjunctive programs. Before doing this, we need introduce some notations. Let $\mu(\mathcal{T}_\Pi \uparrow \omega) = \{I \mid I \in \mathcal{T}_\Pi \uparrow \omega \text{ and } I \in \mathcal{T}_\Pi(\{I\})\}$. Then by Lemma 1, $\mu(\mathcal{T}_\Pi \uparrow \omega)$ represents the set of QC models of Π included in the fixpoint closure. In addition, we let $\min(I) = \{I \in \mathcal{I} \mid \text{if } I' \subset I, \text{ then } I' \notin \mathcal{I}\}$. Then we get the following result,

Theorem 3. *Let Π be a positive extended logic program and $MQCM(\Pi)$ be the set of all minimal QC models of Π .*

$$MQCM(\Pi) = \min(\mu(\mathcal{T}_\Pi \uparrow \omega))$$

Example 5. *In Example 4, $\min(\mu(\mathcal{T}_{\Pi_1} \uparrow \omega)) = \{\{-lh - broken, rh - broken\}\}$, and $MQCM(\Pi_1) = \{\{-lh - broken, rh - broken\}\}$. Hence, $MQCM(\Pi_1) = \min(\mu(\mathcal{T}_{\Pi_1} \uparrow \omega))$.*

4 Entailment under the QC Model Semantics

The constraint imposed on the satisfaction of disjunction is so strong that disjunction introduction rule [12] can not be satisfied, i.e., $\frac{\alpha}{\alpha \vee \beta}$ fails. For instance, $a \vee b$ can not be entailed by the program $\{\neg a \leftarrow, a \leftarrow, b \leftarrow c\}$ under the above satisfaction for disjunction, which is not intuitive. Hence, we need define the notion of weak satisfaction.⁶

Definition 10. *Let I be an interpretation and Π be a program. a_1, \dots, a_m are literals.*

- (1) *If $a \in \mathcal{A}_\Pi$, then $I \models_w a$ iff $a \in I$;*
- (2) *If $a \in \mathcal{A}_\Pi$, then $I \models_w \neg a$ iff $\neg a \in I$;*
- (3) *$I \models_w a_1 \vee \dots \vee a_n$ iff $I \models_w a_1$ or ... or $I \models_w a_n$;*
- (4) *$I \models_w a_1 \wedge \dots \wedge a_n$ iff $\forall i(1 \leq i \leq n), I \models_w a_i$.*

Hence, we are ready to define the two main entailment problems in our approach.

Definition 11. *Let Π be a program and F be a propositional formula. Then,*

- *Π bravely entails F ($\Pi \models_B F$) iff $I \models_w F$ for some $I \in MQCM(\Pi)$.*
- *Π cautiously entails F ($\Pi \models_C F$) iff $I \models_w F$ for every $I \in MQCM(\Pi)$.*

Example 6. *Reconsider Example 1, under the QC model semantics, righthand - broken and \neg lefthand - broken can be entailed by both brave reasoning and cautious reasoning.*

⁶ If we consider the satisfaction defined in Definition 4 is a strong one, then the satisfaction here is a weak one.

5 Related Work

One related work is the answer set semantics illustrated in the background section defined by Gelfond and Lifschitz [14]. The main difference between the answer set semantics and our semantics is how to deal with inconsistent information. Gelfond and Lifschitz adopt the way of trivializing the results while we tolerate the contradiction. However, when a positive extended disjunctive program is classically consistent, there is a close connection between these two semantics.

Theorem 4. *Let Π be a consistent positive extended disjunctive program. Then I is a consistent minimal QC model of Π iff I is an answer set of Π .*

The other closely related work is the paraconsistent minimal model (p-minimal model for short) semantics proposed by Sakama and Inoue [13]. The p-minimal model semantics is based on four-valued logic, which makes the semantics to behave paraconsistently. However, such semantics is a little weak to derive consistent information when the program is indeed classically consistent. Recall Example 1, we can get two p-minimal models $\{lh - broken, \neg lh - broken\}$ and $\{\neg lh - broken, rh - broken\}$. In fact, a rational behavior for an automated system should present only the latter model for users but not give users two models and leave the work of selecting which model is reasonable to users.

6 Conclusion and Future Work

In this paper, we have proposed a new paraconsistent semantics for logic programs, namely the QC model semantics. Such semantics is an alternative approach to the traditional paraconsistent semantics: it enhances the ability of paraconsistent reasoning by imposing the strong constraint on the satisfaction of disjunction. In other words, by putting the link between a disjunction and its negative disjunct, more non-trivial classical conclusions can be captured by the QC model semantics. The QC model semantics is also a paraconsistent generalization of the traditional answer set semantics [14]. Our approach decouples the link between a literal and its complement which makes it exhibit the nice feature of paraconsistent reasoning. Based on the notion of weak satisfaction, we have also defined how to perform entailment under our semantics.

Indeed, our current work can be easily generalized to programs containing default negation, which will be our future work. To decide the complexity of the main entailment problems and implement our semantics are also left to be the future work.

References

1. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The dlv system for knowledge representation and reasoning. *ACM Transactions on Computational Logic* 7(3), 499–562 (2006)

2. Hunter, A.: Paraconsistent logics. In: Handbook of defeasible reasoning and uncertainty management systems. Reasoning with actual and potential contradictions, vol. 2, pp. 11–36. Kluwer Academic Publishers, Norwell (1998)
3. Carnielli, W., Coniglio, M., Marcos, J.: Logics of formal inconsistency. In: Handbook of Philosophical Logic, 2nd edn., vol. 14, pp. 1–93. Springer, Heidelberg (2007)
4. Arieli, O.: Distance-based paraconsistent logics. *International Journal of Approximate Reasoning* 48(3), 766–783 (2008)
5. Belnap, N.D.: A useful four-valued logic. In: Epstein, G., Dunn, J.M. (eds.) *Modern Uses of Multiple-Valued Logic*, pp. 7–37. Reidel Publishing Company, Boston (1977)
6. Blair, H.A., Subrahmanian, V.S.: Paraconsistent logic programming. *Theoretical Computer Science* 68(2), 135–154 (1989)
7. Damásio, C.V., Pereira, L.M.: A survey of paraconsistent semantics for logic programs. In: Handbook of defeasible reasoning and uncertainty management systems. Reasoning with actual and potential contradictions, vol. 2, pp. 241–320. Kluwer Academic Publishers, Norwell (1998)
8. Arieli, O.: Paraconsistent declarative semantics for extended logic programs. *Annals of Mathematics and Artificial Intelligence* 36(4), 381–417 (2002)
9. Alcântara, J., Damásio, C.V., Pereira, L.M.: Paraconsistent logic programs. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) *JELIA 2002. LNCS (LNAI)*, vol. 2424, pp. 345–356. Springer, Heidelberg (2002)
10. Alcântara, J., Damásio, C.V., Pereira, L.M.: A declarative characterization of disjunctive paraconsistent answer sets. In: de Mántaras, R.L., Saitta, L. (eds.) *ECAI 2004*. pp. 951–952. IOS Press, Amsterdam (2004)
11. Poole, D.: What the lottery paradox tells us about default reasoning. In: *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning*. pp. 333–340. Morgan Kaufmann, San Francisco (1989)
12. Hunter, A.: Reasoning with contradictory information using quasi-classical logic. *Journal of Logic and Computation* 10(5), 677–703 (2000)
13. Sakama, C., Inoue, K.: Paraconsistent stable semantics for extended disjunctive programs. *Journal of Logic and Computation* 5(3), 265–285 (1995)
14. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9(3/4), 365–386 (1991)
15. Minker, J., Seipel, D.: Disjunctive logic programming: A survey and assessment. In: Kakas, A.C., Sadri, F. (eds.) *Computational Logic: Logic Programming and Beyond. LNCS (LNAI)*, vol. 2407, pp. 472–511. Springer, Heidelberg (2002)
16. Lloyd, J.W.: *Foundations of logic programming*, 2nd extended edn. Springer, New York (1987)
17. Emden, M.H.V., Kowalski, R.A.: The semantics of predicate logic as a programming language. *Journal of the ACM* 23, 569–574 (1976)

Prime Implicates and Reduced Implicate Tries^{*}

Neil V. Murray¹ and Erik Rosenthal²

¹ Department of Computer Science, State University of New York, Albany, NY 12222, USA
nvm@cs.albany.edu

² Department of Mathematics, University of New Haven, West Haven, CT 06516, USA
erosenthal@newhaven.edu

Abstract. The reduced implicate trie (*ri*-trie) is a data structure that was introduced as a target language for knowledge compilation. It has the property that, even when large, it guarantees fast response to queries. It also has the property that each prime implicate of the formula it represents corresponds to a branch. In this paper, those prime branches are characterized, and a technique is developed for marking nodes to identify branches that correspond to non-prime implicates. This marking technique is enhanced to allow discovery of prime implicate subsets of queries that are answered affirmatively.

1 Introduction

Consequences expressed as minimal clauses that are implied by a formula are its *prime implicates*, while minimal conjunctions of literals that imply a formula are its *prime implicants*. Implicates are useful in certain approaches to non-monotonic reasoning [13,21,23], where all consequences of a formula — for example, the support set for a proposed common-sense conclusion — are required. Another application is error analysis during hardware verification, where satisfying models are desired. Many algorithms have been developed to compute the prime implicates (or implicants) of a propositional boolean formula — see, for example, [1,2,5,9,10,12,15,20,22,24,25].

Several investigators have represented knowledge bases as propositional theories, typically as sets of clauses. However, since the question, Does $\mathcal{NP} = \mathcal{P}$? remains open — i.e., there are no known polynomial algorithms for problems in the class \mathcal{NP} — the time to answer queries is (in the worst case) exponential. The *reduced implicate trie* (*ri*-trie) was developed [17,18] as a solution to a problem posed by Kautz and Selman [11] in 1991. Their idea, known as *knowledge compilation*, was to pay the exponential penalty once by compiling the knowledge base into a *target language* that would guarantee fast response to queries.

In [17] *ri*-tries are shown to guarantee response time *linear in the size of the query*. Thus queries of any knowledge base that can be “practically compiled” — i.e., can be built in reasonable time and space¹ — can always be answered quickly. These tries contain all prime implicates among their branches. In this paper, those branches are identified.

^{*} This research was supported in part by the National Science Foundation under grants IIS-0712849 and IIS-0712752.

¹ *Reasonable* is a subjective term, presumably defined by the end user.

Implicates are of interest with respect to propositional knowledge bases because a typical query has the form, *Is a clause logically entailed by the theory?* which is equivalent to the question, *Is the clause — i.e., the query — an implicate of the formula representing the theory?*

Reduced implicate tries are reviewed in Section 1.2; new results are presented in Sections 2 and 3. In Section 2, prime implicate branches are made recognizable by characterizing and marking the non-prime implicate branches. In Section 3, the marking technique is enhanced to allow discovery of prime implicate subsets of queries that are answered affirmatively.

As usual, an *atom* is a propositional variable, a *literal* is an atom or the negation of an atom, and a *clause* is a disjunction of literals.² Clauses are often referred to as sets of literals.

Proofs for the results described in this section can be found in [17|18|19].

1.1 Complete Implicate Tries

The trie is a well-known data structure introduced by Fredkin in 1960 [6]; a variation was introduced by Morrison in 1968 [16]. It is a tree in which each branch represents the sequence of symbols labeling the nodes³ on that branch, in descending order. Tries have been used to represent logical formulas, including sets of prime implicates [23]. The nodes along each branch represent the literals of a clause, and the conjunction of all such clauses is a CNF equivalent of the formula represented by the trie. In general, the CNF formula can be significantly larger than the corresponding trie. Tries that represent logical formulas can be interpreted directly as formulas in negation normal form (NNF): A trie consisting of a single node represents the label of that node. Otherwise, the trie represents the disjunction of the label of the root with the conjunction of the formulas represented by the tries rooted at its children.

A trie that stores all (non-tautological) implicates of a formula is called a *complete implicate trie*. For a formal definition and its properties, see [18].

1.2 Reduced Implicate Tries

Recall that for any logical formulas \mathcal{F} and α and subformula \mathcal{G} of \mathcal{F} , $\mathcal{F}[\alpha/\mathcal{G}]$ denotes the formula produced by substituting α for every occurrence of \mathcal{G} in \mathcal{F} . If α is a truth functional constant 0 or 1 (*false* or *true*), and if p is a negative literal, we will slightly abuse this notation by interpreting the substitution $[0/p]$ to mean that 1 is substituted for the atom that p negates.

The following simplification rules, when applied to a complete implicate trie, will produce a *reduced implicate trie* (*ri-trie*).

$$\begin{array}{ll} \text{SR1. } \mathcal{F} \rightarrow \mathcal{F}[\mathcal{G}/\mathcal{G} \vee 0] & \mathcal{F} \rightarrow \mathcal{F}[\mathcal{G}/\mathcal{G} \wedge 1] \\ \text{SR2. } \mathcal{F} \rightarrow \mathcal{F}[0/\mathcal{G} \wedge 0] & \mathcal{F} \rightarrow \mathcal{F}[1/\mathcal{G} \vee 1] \\ \text{SR3. } \mathcal{F} \rightarrow \mathcal{F}[0/p \wedge \neg p] & \mathcal{F} \rightarrow \mathcal{F}[1/p \vee \neg p] \end{array}$$

² The term *clause* is sometimes used for a conjunction of literals, especially with *disjunctive normal form*.

³ Many variations have been proposed in which arcs rather than nodes are labeled, and the labels are sometimes strings rather than single symbols.

The branches of an *ri*-trie represent the *relatively prime implicates* [18]: If \mathcal{F} is a logical formula, and if the variables of \mathcal{F} are ordered, then a relatively prime implicate is one for which no proper prefix is also an implicate. If the leaf node of a branch in an *ri*-trie is labeled p_i , then every extension with variables of index greater than i is a branch in the complete implicate trie of \mathcal{F} . These extensions correspond to implicates of \mathcal{F} that are not relatively prime and that are represented implicitly by that branch in the *ri*-trie.

Theorem 1. Given a logical formula \mathcal{F} and an ordering of the variables of \mathcal{F} , then the branches of the corresponding *ri*-trie represent precisely the relatively prime implicates. In particular, since prime implicates are relatively prime, each is represented by a branch in the trie. \square

1.3 Computing Reduced Implicate Tries

Let \mathcal{F} be a logical formula, and let the variables of \mathcal{F} be $V = \{v_1, v_2, \dots, v_n\}$. Then the *ri*-trie of \mathcal{F} can be obtained by applying the recursively defined RIT operator (introduced in [17]):

$$\text{RIT}(\mathcal{F}, V) = \begin{cases} \mathcal{F} & V = \emptyset \\ \left(\begin{array}{c} v_i \vee \text{RIT}(\mathcal{F}[0/v_i], V - \{v_i\}) \\ \wedge \\ \neg v_i \vee \text{RIT}(\mathcal{F}[1/v_i], V - \{v_i\}) \\ \wedge \\ \text{RIT}((\mathcal{F}[0/v_i] \vee \mathcal{F}[1/v_i]), V - \{v_i\}) \end{array} \right) & v_i \in V \end{cases}$$

where v_i is the variable of lowest index in V .

Implicit is the use of simplification rules **SR1**, **SR2**, and **SR3**.

Theorem 2. If \mathcal{F} is a logical formula with variable set V , then $\text{RIT}(\mathcal{F}, V)$ is logically equivalent to \mathcal{F} . \square

Let $\text{Imp}(\mathcal{F})$ denote the set of all implicates of \mathcal{F} .

Lemma 1. Given logical formulas \mathcal{F} and \mathcal{G} , $\text{Imp}(\mathcal{F}) \cap \text{Imp}(\mathcal{G}) = \text{Imp}(\mathcal{F} \vee \mathcal{G})$. \square

The last lemma means that the implicates being computed in the third conjunct of the RIT operator are precisely those that occur in both of the first two (ignoring, of course, the root labels v_i and $\neg v_i$). This third conjunct can thus be computed from the first two, and the direct recursive call on $(\mathcal{F}[0/v_i] \vee \mathcal{F}[1/v_i])$ can be avoided. This is significant because in this call, the size of the argument essentially doubles.

Theorem 3. Let \mathcal{F} be a logical formula with variable set V , and let C be an implicate of \mathcal{F} . Then there is a unique branch of $\text{RIT}(\mathcal{F}, V)$ that is a prefix of C , and every branch is a relatively prime implicate. \square

1.4 Ternary Representation

Observe that the RIT operator essentially produces a conjunction of three tries. It is therefore natural to represent an *ri*-trie as a ternary trie. The root of the third subtrie is labeled 0. One advantage of this representation is that the *i*th variable appears only at level *i*. Another is that any subtrie (including the entire trie) is easily expressed as a four-tuple consisting of its root and the three subtrees. For example, for a subtrie \mathcal{T} we might write $\langle r, \mathcal{T}^+, \mathcal{T}^-, \mathcal{T}^0 \rangle$, where r is the root label of \mathcal{T} , and \mathcal{T}^+ , \mathcal{T}^- , and \mathcal{T}^0 are the three subtrees.

A trivial technical difficulty arises with the ternary representation: The zeroes along branches interfere with the prefix property of Theorem 3. But this is easily dealt with by interpreting the statement, *A branch B is a prefix of a clause C*, to mean *The clause represented by B with zeroes simplified away is a prefix of C*. The zeroes cause no difficulty when traversing branches in the trie.

Obtaining the ternary representation with the RIT operator requires only a minor change: disjoining 0 to the third conjunct. The notation $ri(\mathcal{F}, V) = 0 \vee \text{RIT}(\mathcal{F}, V)$ will be used for the ternary *ri*-trie of \mathcal{F} with variable ordering V . For the remainder of this paper, we will generally assume this ternary representation. As a result, the forest denoted by $\text{RIT}(\mathcal{F}, V)$ will contain three tries whose roots are labeled by a variable, its complement, and zero.

Theorem 4. Let \mathcal{F} and \mathcal{G} be logically equivalent formulas. Then, with respect to a fixed variable ordering V , $ri(\mathcal{F}, V)$ is isomorphic to $ri(\mathcal{G}, V)$. □

1.5 Intersecting *ri*-Tries

Given two formulas \mathcal{F} and \mathcal{G} , fix an ordering of the union of their variable sets, and let $\mathcal{T}_{\mathcal{F}}$ and $\mathcal{T}_{\mathcal{G}}$ be the corresponding *ri*-tries. The *intersection* of $\mathcal{T}_{\mathcal{F}}$ and $\mathcal{T}_{\mathcal{G}}$ is defined to be the *ri*-trie (with respect to the given variable ordering) that represents the intersection of the implicate sets. By Theorems 2 and 4 and Lemma 1, this is the *ri*-trie for $\mathcal{F} \vee \mathcal{G}$.

The intersection of two tries (with the same variable ordering) is produced by the INT operator introduced in [19].

Theorem 5. Let $\mathcal{T}_{\mathcal{F}}$ and $\mathcal{T}_{\mathcal{G}}$ be the respective *ri*-tries of \mathcal{F} and \mathcal{G} (with the same variable ordering). Then $\text{INT}(\mathcal{T}_{\mathcal{F}}, \mathcal{T}_{\mathcal{G}})$ is the intersection of $\mathcal{T}_{\mathcal{F}}$ and $\mathcal{T}_{\mathcal{G}}$; in particular, $\text{INT}(\mathcal{T}_{\mathcal{F}}, \mathcal{T}_{\mathcal{G}})$ is the *ri*-trie of $\mathcal{F} \vee \mathcal{G}$ (with respect to the given variable ordering). □

Theorem 5 provides a formal basis for a definition of the RIT operator that produces *ri*-tries using intersection. It is obtained from the earlier definition essentially by replacing the third conjunct by $\text{INT}(\text{RIT}(\mathcal{F}[0/v_i], V - \{v_i\}), \text{RIT}(\mathcal{F}[1/v_i], V - \{v_i\}))$.

2 Prime Implicate Branches

There is a one-to-one correspondence, by Theorem 3, between relatively prime implicates of a formula \mathcal{F} and branches in $\mathcal{T}_{\mathcal{F}}$. Since every prime implicate is relatively prime, there is a one-to-one correspondence between the prime implicates of \mathcal{F} and

the branches of a subtree of $\mathcal{T}_{\mathcal{F}}$. Finding non-prime branches amounts to subsumption checking, and there are several methods available for doing so. However, the construction of an ri -trie provides information that can be employed to recognize branches whose labels form non-prime implicates.

2.1 Characterizing Non-prime Branches

The next lemma offers one characterization of branches in an ri -trie that represent non-prime implicates. The existence of $\overline{p_i}$ is not surprising; the significance of the lemma is that $i < m$.

Theorem 6. Let $C = \{p_1, \dots, p_m\}$ be a relatively prime implicate of a logical formula \mathcal{F} . Then C is not a prime implicate iff there is a relatively prime implicate $\tilde{C}_k = \{p_1, \dots, \overline{p_i}, \dots, p_k\}$ of \mathcal{F} , where $p_i \in C$, $1 \leq i \leq k$, $i < m$ and $k \leq m$. In that case, $C - \{p_i\}$ is itself a relatively prime implicate. \square

The following terminology will be useful in the sequel. A *suffix* of a branch $B = \{p_1, \dots, p_n\}$ in an ri -trie is a sub-branch of B labeled $\{p_j, p_{j+1}, \dots, p_n\}$, $1 \leq j \leq n$. This suffix will be referred to as the *suffix of B beginning with p_j* . A relatively prime implicate $C = \{p_1, \dots, p_m\}$ is called *i -redundant* if for some i , $1 \leq i < m$, $C - \{p_i\}$ is an implicate.

2.2 Identifying Non-prime Branches

If $C = \{p_1, \dots, p_m\}$ is a relatively prime implicate of \mathcal{F} , then there is a unique branch B corresponding to C in the ri -trie $\mathcal{T}_{\mathcal{F}}$ of \mathcal{F} . We often will not distinguish between the clause C and the branch B (as long as there is no possibility of confusion). Theorem 6 provides a characterization of relatively prime implicates that are not prime: If C is not prime, then there is a relatively prime implicate $\tilde{C}_k = \{p_1, \dots, \overline{p_i}, \dots, p_k\}$ of \mathcal{F} , where $p_i \in C$, $1 \leq i \leq k$, $i < m$ and $k \leq m$. Since $\mathcal{T}_{\mathcal{F}}$ is the ri -trie for \mathcal{F} , the branch \tilde{C}_k shares with the branch C precisely the nodes labeled p_1, \dots, p_{i-1} — see Figure 11.4

The identification and marking of nodes that determine non-prime branches is accomplished by the MARK operator; this is performed after the entire ri -trie has been constructed. MARK is defined as a function; it takes an ri -trie (represented by the root node) as argument and returns the same trie but with the side effect of marking non-prime branches. The primary task of MARK is to install marks in the three subtrees of its argument and then employ the SCAN procedure on those subtrees.

It is in SCAN that the conditions of Theorem 6 are recognized and nodes are marked. Observe that these conditions can occur only if all three subtree arguments of SCAN are non-empty — i.e., the branches C , \tilde{C}_k , and C' in Figure 11 must be non-empty. The test $T^0 \neq \emptyset$ is sufficient since it is the intersection of the first two. (Note that T^0 may be empty even if the first two are not.)

It is convenient to assume a primitive procedure $npm(< arg >)$ that merely marks its ri -trie node argument as “non-prime.” Recall that ternary notation is assumed, and that $T = \langle r, T^+, T^-, T^0 \rangle$.

⁴ The figure is drawn as if p_i is positive (so $\overline{p_i} = \neg p_i$); the dual case is straightforward.

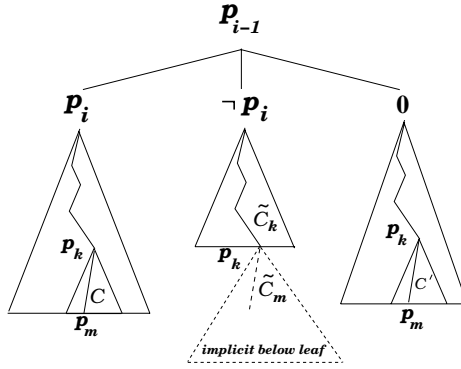


Fig. 1. A Non-Prime Relatively Prime Implicate

```

define MARK ( $T$ :trinode);
if not(leaf( $T$ )) then
    |   SCAN (MARK ( $T^+$ ),
    |       MARK ( $T^-$ ),
    |       MARK ( $T^0$ ))
end
return ( $T$ );
end MARK
    
```

```

define SCAN ( $\mathcal{P}$ ,  $\mathcal{M}$ ,  $\mathcal{I}$ :trinode);
if  $\mathcal{I} \neq \emptyset$  then
    |   if leaf( $\mathcal{P}$ ) then
    |       |   npm( $\mathcal{M}$ );
    |       |   if leaf( $\mathcal{M}$ ) then
    |       |       |   npm( $\mathcal{P}$ );
    |       |   else
    |       |       |   if leaf( $\mathcal{M}$ ) then
    |       |           |   npm( $\mathcal{P}$ )
    |       |           |   else
    |       |               |   SCAN ( $\mathcal{P}^+$ ,  $\mathcal{M}^+$ ,  $\mathcal{I}^+$ );
    |       |               |   SCAN ( $\mathcal{P}^-$ ,  $\mathcal{M}^-$ ,  $\mathcal{I}^-$ );
    |       |               |   SCAN ( $\mathcal{P}^0$ ,  $\mathcal{M}^0$ ,  $\mathcal{I}^0$ );
    |       |           end
    |       |       end
    |       |   end
    |       |   end
end
end SCAN
    
```

The ri -trie for a formula \mathcal{F} is produced by computing $ri(\mathcal{F}) = 0 \vee \text{RIT}(\mathcal{F})$. A trie with non-prime marks can be obtained from $ri(\mathcal{F})$ with $ri^{np}(\mathcal{F}) = \text{MARK}(ri(\mathcal{F}))$; the result will be referred to as an ri^{np} -trie. It is evident from the definitions of MARK and SCAN that MARK is invoked on every subtree, and SCAN is invoked on every triplet in which at least one sibling is not empty. Moreover, unless none of the siblings is empty, SCAN will immediately exit. In particular, for any subtree of the form shown in Figure 1, SCAN will be invoked on the three main subtrees.

In any such invocation, if C is not prime, Theorem 6 applies, and we may assume that $C = \{p_1, \dots, p_m\}$, and that $\tilde{C}_k = \{p_1, \dots, \bar{p}_i, \dots, p_k\}$, $i < m$, $k \leq m$, as pictured in Figure 1. As the paths are being traced, Theorem 6 ensures that the end of branch \tilde{C}_k will be encountered by the time the end of C is. One leaf test (both if the end of branch C is reached) by SCAN will evaluate to true. If the trace along branch C does not reach

a leaf — i.e., if $k < m$, so that p_k is not the leaf of C — then several branches have $C_k = \{p_1, p_2, \dots, p_k\}$ as a prefix. If D is such a branch, resolving D and \tilde{C}_k amounts to deleting p_i from D and thus produces an implicate that is a proper subset of D , which is to say, D is not a prime implicate.⁵

Suppose now that p_k is the leaf of C . Then $C - \{p_i\}$ is the resolvent of C and \tilde{C}_k , so $C - \{p_i\}$ is an implicate — i.e., both C and \tilde{C}_k are not prime. In that case, both nodes labeled p_k are marked not prime by SCAN. This analysis, along with Theorem 6, reveals that when the ri^{np} algorithm terminates, all non-prime branches contain a non-prime mark. This proves

Theorem 7. Let $\mathcal{T} = ri(\mathcal{F})$. Then the computation of $ri^{np}(\mathcal{F}) = \text{MARK}(\mathcal{T})$ invokes MARK on every node of \mathcal{T} and invokes SCAN on every triplet of non-empty siblings in \mathcal{T} . Furthermore,

1. If $C = \{p_1, \dots, p_m\}$ is a non-prime relatively prime implicate of \mathcal{F} , then for some $i, 1 \leq i < m$, $C - \{p_i\}$ is relatively prime.
2. The suffix of $C - \{p_i\}$ beginning with p_{i+1} is in the zero sibling of p_i in C .
3. $\text{SCAN}(p_i, \bar{p}_i, 0)$ terminates on some leaf descendant p_k of \bar{p}_i , $i \leq k \leq m$.
4. All branches containing suffixes that begin with $p_k \in C$ are not prime.
5. If p_k is the leaf of C , then $C - \{p_i\} \cup \{\bar{p}_i\}$ is a non-prime branch.
6. When the ri^{np} algorithm terminates, all non-prime branches are so marked. \square

2.3 Early Recognition

Theorem 7 guarantees that ri -tries can be constructed so that every branch can be recognized as a prime implicate or not. This is completely adequate for answering queries and for determining whether an implicate is prime. The reason is that a search that produces an affirmative answer traverses a branch to its leaf. Thus, if the searched branch is non-prime, the marked node on it will be encountered.

Suppose, however, that an ri -trie is being explored for prime implicates without reference to any particular query. If a node p_i in the trie had only non-prime implicates as its suffixes, it would be advantageous if p_i were marked. Theorem 7 does not guarantee this. What is known is that every extension from p_i contains a node with a non-prime mark.

Consider, for example, the ri -trie in Figure 2. The solid circles identify nodes that would be marked as non-prime in compiling this ri -trie with ri^{np} . Any node with no unmarked children can itself be marked. Were this done in Figure 2 the leftmost 0, $\neg p$, and the two occurrences of q having a zero parent would be marked non-prime; this is indicated by the dashed circles.

Propagating non-prime marks toward the root can be accomplished entirely within npm by extending the marking side effects as follows: If the node being marked has no unmarked siblings, mark the parent, and recursively apply this analysis to the parent. Assuming that npm has been so defined, ri^{np} will produce ri -tries in which each non-prime branch has been marked at the earliest possible node, and prime implicates can be found by traversing branches with no marked nodes.

⁵ Recall that resolving two clauses of a formula produces an implicate of the formula. Of course, D is relatively prime since it is a branch in the trie.

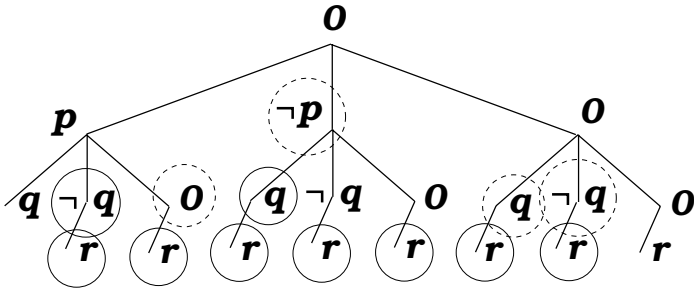


Fig. 2. Propagating Non-Prime Marks

3 Obtaining Prime Implicate Branches

For each query of a knowledge base with an affirmative answer, there is a unique prefix that is relatively prime and represented by a branch in the ri^{np} -trie. If this prefix is not prime, then this branch will contain a marked node, and a more refined query — i.e., a proper subset — would also secure a *yes* answer. In the case that a query is “prime” — i.e., every proper subset leads to a negative answer — then the traversed branch in the ri^{np} -trie will have no marked node. Thus the ri^{np} -trie can also determine whether a query is “prime.”

In the case of a not prime, yes answer, finding all prime implicates that are subsets of the query cannot be accomplished efficiently, since there can be exponentially many of them. However, one prime subset can be found in time polynomial in the size of the query. The key is observing how a node gets marked as non-prime. This occurs precisely in the base cases of SCAN, that is, whenever one of the leaf tests is satisfied. And this occurs whenever “parallel” suffixes have been traced until a leaf node is encountered in one or both. These suffixes are identical except that they begin with complementary labels. Placement of a non-prime mark occurs because Theorem 6 applies, and it is the literal beginning the suffix that is redundant. Furthermore, this redundant literal has a zero sibling that begins a suffix that is otherwise identical to the suffix(es) on which the mark is placed. (Recall that the zero subtree is the intersection of the subtrees containing the “parallel” suffixes.)

One way to account for a redundancy is to mark the node in question with a pointer to the zero sibling of the redundant node. When such a marked node is encountered answering a query, the pointer to the zero sibling can be followed, noting that the redundant node was unnecessary to verify the query. Processing the query would then continue from the zero sibling. Employing this strategy whenever a non-prime pointer is encountered would not alter the answer, but in the case when the query is indeed an implicate, the traced branch will correspond to a prime implicate subset of the query.

The SCAN and npm operators require only an extra parameter to accommodate the addition of the zero sibling pointer. Their new definitions are presented below; the definition of ri^{np} is unchanged: $ri^{np}(\mathcal{F}) = \text{MARK}(ri(\mathcal{F}))$.

```

define MARK ( $\mathcal{T}$ :trinode);
if not(leaf( $\mathcal{T}$ )) then
  | SCAN (MARK ( $\mathcal{T}^+$ ),
  |   MARK ( $\mathcal{T}^-$ ),
  |   MARK ( $\mathcal{T}^0$ ),
  |   ↑  $\mathcal{T}^0$ )
end
return ( $\mathcal{T}$ );
end MARK

define SCAN ( $\mathcal{P}$ ,  $\mathcal{M}$ ,  $\mathcal{I}$ :trinode,
               $z$ sib:↑trinode);
if  $\mathcal{I} \neq \emptyset$  then
  | if leaf( $\mathcal{P}$ ) then
  | |  $npm(\mathcal{M}, z$ sib);
  | | if leaf( $\mathcal{M}$ ) then  $npm(\mathcal{P}, z$ sib);
  | | else
  | | | if leaf( $\mathcal{M}$ ) then
  | | | |  $npm(\mathcal{P}, z$ sib)
  | | | else
  | | | | SCAN ( $\mathcal{P}^+$ ,  $\mathcal{M}^+$ ,  $\mathcal{I}^+$ ,  $z$ sib);
  | | | | SCAN ( $\mathcal{P}^-$ ,  $\mathcal{M}^-$ ,  $\mathcal{I}^-$ ,  $z$ sib);
  | | | | SCAN ( $\mathcal{P}^0$ ,  $\mathcal{M}^0$ ,  $\mathcal{I}^0$ ,  $z$ sib);
  | | | end
  | | end
  | end
end
end SCAN

```

If \mathcal{F} is compiled to ri^{np} -trie $\mathcal{T}_{\mathcal{F}}$ using ri^{np} with pointer marks, then a prime implicate subset can be determined for any query to which the answer is yes. As with the unmarked ri -trie, the response time is linear in the size of the branch traversed. But this is no longer bound by the length of the query. Suppose query $Q = \{q_1, \dots, q_m\}$ is posed. In the worst case, a non-prime pointer is discovered at q_m , and it leads to the zero sibling of q_1 . This may happen again after tracing $\{q_2, \dots, q_m\}$. Therefore, discovering that $\{q_m\}$ (or some small subset of Q consisting of variables comparatively late in the order) is a prime implicate subset of Q could require $\mathbf{O}(m^2)$ time. However, a simple three-way response of *yes-prime*, *yes-non-prime*, or *no*, can still be obtained in time linear in the size of the query.

References

1. Bittencourt, G.: Combining syntax and semantics through prime form representation. *Journal of Logic and Computation* 18, 13–33 (2008)
2. Coudert, O., Madre, J.: Implicit and incremental computation of primes and essential implicant primes of boolean functions. In: 29th ACM/IEEE Design Automation Conference, pp. 36–39 (1992)
3. Darwiche, A.: Decomposable negation normal form. *Journal of the ACM* 48(4), 608–647 (2001)
4. Darwiche, A., Marquis, P.: A knowledge compilation map. *Journal of AI Research* 17, 229–264 (2002)
5. de Kleer, J.: An improved incremental algorithm for computing prime implicants. In: Proc. AAAI 1992, San Jose, CA, pp. 780–785 (1992)
6. Fredkin, E.: Trie memory. *Communications of the ACM* 3(9), 490–499 (1960)
7. Hähnle, R., Murray, N.V., Rosenthal, E.: Normal forms for knowledge compilation. In: Proc. International Symposium on Methodologies for Intelligent Systems - ISMIS, Saratoga Springs, NY, pp. 304–313 (2005)

8. Hai, L., Jigui, S.: Knowledge compilation using the extension rule. *Journal of Automated Reasoning*, 93–102 (2004)
9. Jackson, P.: Computing prime implicants incrementally. In: Kapur, D. (ed.) *CADE 1992. LNCS (LNAI)*, vol. 607, pp. 253–267. Springer, Heidelberg (1992)
10. Jackson, P., Pais, J.: Computing prime implicants. In: Stickel, M.E. (ed.) *CADE 1990. LNCS (LNAI)*, vol. 449, pp. 543–557. Springer, Heidelberg (1990)
11. Kautz, H., Selman, B.: A general framework for knowledge compilation. In: *Proc. International Workshop on Processing Declarative Knowledge (PDK)*, Kaiserslautern, Germany (July 1991)
12. Kean, A., Tsiknis, G.: An incremental method for generating prime implicants/implicates. *Journal of Symbolic Computation* 9, 185–206 (1990)
13. Kean, A., Tsiknis, G.: Assumption based reasoning and clause management systems. *Computational Intelligence* 8(1), 1–24 (1992)
14. Liberatore, P., Schaerf, M.: Compilability of propositional abduction. *ACM Transactions on Computational Logic* 8(1), Article 2 (2007)
15. Manquinho, V.M., Flores, P.F., Silva, J.P.M., Oliveira, A.L.: Prime implicant computation using satisfiability algorithms. In: *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, Newport Beach, US A, November 1997, pp. 232–239 (1997)
16. Morrison, D.R.: Patricia — practical algorithm to retrieve information coded in alphanumeric. *Journal of the ACM* 15(4), 514–534 (1968)
17. Murray, N.V., Rosenthal, E.: Efficient query processing with compiled knowledge bases. In: Beckert, B. (ed.) *TABLEAUX 2005. LNCS (LNAI)*, vol. 3702, pp. 231–244. Springer, Heidelberg (2005)
18. Murray, N.V., Rosenthal, E.: Efficient query processing with reduced implicate tries. *Journal of Automated Reasoning* 38(1-3), 155–172 (2007)
19. Murray, N.V., Rosenthal, E.: Updating reduced implicate tries. In: Olivetti, N. (ed.) *TABLEAUX 2007. LNCS (LNAI)*, vol. 4548, pp. 183–198. Springer, Heidelberg (2007)
20. Ngair, T.: A new algorithm for incremental prime implicate generation. In: *Proc. IJCAI 1993*, Chambéry, France (1993)
21. Przymusiński, T.C.: An algorithm to compute circumscription. *Artificial Intelligence* 38, 49–73 (1989)
22. Ramesh, A., Becker, G., Murray, N.V.: Cnf and dnf considered harmful for computing prime implicants/implicates. *Journal of Automated Reasoning* 18(3), 337–356 (1997)
23. Reiter, R., de Kleer, J.: Foundations of assumption-based truth maintenance systems: preliminary report. In: *Proc. 6th National Conference on Artificial Intelligence*, Seattle, WA, July 12–17, 1987, pp. 183–188 (1987)
24. Slagle, J.R., Chang, C.L., Lee, R.C.T.: A new algorithm for generating prime implicants. *IEEE transactions on Computers* C-19(4), 304–310 (1970)
25. Strzemecki, T.: Polynomial-time algorithm for generation of prime implicants. *Journal of Complexity* 8, 37–63 (1992)

Logic for Reasoning about Components of Persuasive Actions

Katarzyna Budzynska¹, Magdalena Kacprzak^{2,*}, and Pawel Rembelski³

¹ Institute of Philosophy, Cardinal Stefan Wyszyński University in Warsaw, Poland

² Faculty of Computer Science, Białystok University of Technology, Poland

³ Faculty of Computer Science, Polish-Japanese Institute of Information Technology

<http://perseus.ovh.org/>

Abstract. The aim of the paper is to propose an extension for a model of logic \mathcal{AG}_n . Thus far, \mathcal{AG}_n was applied for reasoning about persuasiveness of actions in multi-agent systems, i.e., we examined which arguments, provided by agents, are successful and how big such a success is. Now we enrich our approach in order to study why these arguments are efficient and what attributes cause their success. Therefore, we propose to specify persuasive actions with three parameters: content, goal and means of sending messages. As a result, we can formally express what an agent wants to achieve by executing an action, whether this action can be successful, and if not, recognize the reasons which can cause the failure.

Keywords: Success of persuasion, nonverbal arguments, formal models of persuasive actions.

1 Introduction

Our work is focused on **persuasion** processes which can be conducted in artificial societies such as multi-agent systems. Therefore we analyze scenarios in which one party (an agent) tries to persuade another party (another agent) to adopt a belief or point of view he or she does not currently hold or influence the activity of this party (actions she or he performs). However we do not limit to persuasion dialogs, like almost all the works in this field, but we take into account all factors which can change beliefs, behavior or attitudes of artificial agents.

In the psychological models [13], one of the important elements of the persuasion is its *goal*. Moreover, they investigate into the different means of social influence, one of which is *nonverbal communication*. In the **formal models** these aspects are rarely discussed. The formal model that includes the persuasion's **goal** is a proposal by K. Atkinson, T. Bench-Capon and P. McBurney [1]. However, they focus on the goal of an action other than persuasion itself (or - in other words - on the goal of an agent engaged in the persuasion). They consider practical reasoning such as: "I'm to be in London at 4.15" and "If I catch the

* The author acknowledges support from Ministry of Science and Higher Education under Białystok University of Technology (grant W/WI/3/07).

2.30 train, I'll be in London at 4.15" therefore "I'll catch the 2.30 train". This is a reasoning about what should be done to achieve a given *goal* according to some criterion. In the above example, the goal of an agent is being in London at 4.15, and the action which allows to achieve this goal is catching the 2.30 train. In this paper, we are not interested in this type of goals. Another models we want to mention here is the most widespread formal approach to persuasion - i.e. the models which are built within the dialog and game-theoretic framework (see [9,10,11,14]). These models consider the goal of persuasion which is the resolution of a conflict or winning an adversary. This goal remains unchanged during the whole process of persuasion. However, they do not discuss goals of individual actions and cannot express them directly in the language of the logic they use.

The **nonverbal means** of persuasion are present in a few frameworks. The nonverbal actions can be verbal as well as nonverbal in the model of persuasive negotiation proposed by S. Ramchurn, N. Jennings and C. Sierra [15]. They assume that threats and rewards can be performed by illocutionary (verbal) actions, e.g. while saying "I will give you money", or environmental (nonverbal) actions, e.g. when giving money by showing a gun. Nevertheless, the focus of their approach is still on the persuasive illocutions. The other model, in which reasoning about nonverbal actions is possible, is multimodal logic of actions and graded beliefs \mathcal{AG}_n proposed by K. Budzynska and M. Kacprzak (see e.g. [3,4]). We interpret persuasive arguments as actions in terms of Algorithmic Logic and Dynamic Logic. As a result, the nature of arguments is not predetermined to be verbal such as it assumed when arguments are treated as sentences. Observe that the persuasion dialog systems [14] do not address the issue of nonverbal arguments since they limit their considerations to the persuasion which is a dialog.

Identifying goal and means of persuasion is especially important when we study the process of persuasion from the point of view of its effectiveness. The persuasion is successful when a persuader achieves the *goal* he pursued. Obviously, some persuasive actions can be stronger than other ones, in particular - depending on the *way the action is performed*. It may be more persuasive to give money than promising "I will give you money" when a negotiator wants to make somebody agree to something. However, thus far there is no stress that the same **content** of a message can have a different impact on the audience depending on the means in which this message is sent. In our example, both in the verbal action (saying "I will give you money") and in the nonverbal action (giving money) the negotiator sends the same content of a message, i.e., a reward (see e.g. [15]). Observe that the second - nonverbal - strategy may turn out to be more persuasive.

The **contribution** of the paper is to propose an extension for a model of logic \mathcal{AG}_n which enables to represent those three aspects of messages sent during the persuasion. \mathcal{AG}_n was designed to study the persuasion systems of agents with respect to the strength of arguments and their influence on the uncertainty

of agents' beliefs. In this paper, we enrich \mathcal{AG}_n expressivity. Furthermore, we present how the proposed model can be included into our software tool Perseus.

The remainder of this paper is structured as follows. In Section 2, we give the general idea of our approach and formalization of \mathcal{AG}_n logic. Section 3 presents the extension of our model. That is, we propose to specify the persuasive actions in terms of content, goal and means of sending messages. In Section 4, we show how the extension of \mathcal{AG}_n enables to enrich the computer system Perseus.

2 \mathcal{AG}_n Logic

In this section we show the logic \mathcal{AG}_n proposed by K. Budzynska and M. Kacprzak [3,4].

2.1 Degrees of Uncertainty

Before we formally define degrees of beliefs first we show some intuitions. We refer to the resource re-allocation problem, i.e. the process of re-distributing resources amongst agents. This problem attracts a lot of attention in the literature, since it has many practical applications (e.g. in the logistic [8], in the informatics technology [6], in the industry and commerce [12]). In [7], agents are able to perform the actions of information-seeking and negotiation. In [5], we extend this approach by allowing agents to persuade each other and give diverse arguments in order to achieve desired resources.

In this framework, agents begin with beliefs specifying which resources they have and which resources they would like to have. Next, they communicate in order to establish which agent has the desired resources and then they start negotiation or persuasion. Consider the following example. Assume that there are five keys and one of them opens a safe, call it 3. At the beginning of persuasion John has two keys 1 and 3 and Peter has three keys 2, 4, and 5. A state of a system is represented by three sets. The first one indicates keys belonging to John, the second indicates keys belonging to Peter, and the third points at the key opening the safe. So, the initial situation is denoted by $(1, 3|2, 4, 5|3)$. John wants to open the safe. He knows what keys are in the system, what keys he has, and knows also that other keys Peter has. Yet he does not know the most important information, i.e., which key allows to open the safe. However he expects that this key has odd number. Therefore he considers three situations as his doxastic alternatives: $(1, 3|2, 4, 5|1)$, $(1, 3|2, 4, 5|3)$, $(1, 3|2, 4, 5|5)$. Formally these visions of the current state are represented by a doxastic relation denoted by $RB(John)$ (see Fig. 1). Notice that John has three visions of the current state, but in only one of them it is true that the key number 3 opens the safe. Thus we say that John believes that the desired key is 3 with the **degree of uncertainty** $\frac{1}{3}$. In all John's visions it holds that the key opening the safe has odd number. So we say that John believes that the desired key has odd number with the degree $\frac{3}{3}$. Since this value is equal to 1 (the highest possible to reach) we can say that John is absolutely sure about this fact.

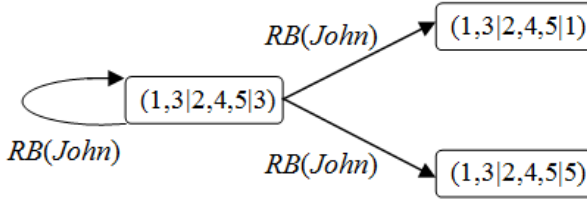


Fig. 1. John’s doxastic relation

In \mathcal{AG}_n logic there are several **doxastic modalities** expressing some meanings of uncertainty. The main operator is $M_i^k(T)$, e.g., $M_{John}^0(three)$ what intuitively means that John considers **more than 0** visions of the current state in which *three* is true (*three* expresses that the key number 3 opens the safe). A dual operator is $B_i^k(T)$, e.g., $B_{John}^2(three)$ what intuitively expresses that John considers **at most 2** visions of the current state in which the thesis *three* does not hold. The next operator is $M_i^1(T)$, e.g., $M_{John}^1(three)$ what intuitively says that John considers **exactly 1** vision in which the thesis *three* holds. Observe that all of these operators do not provide information about the number of all visions considered by an agent. The only operator which gives such an information is the operator we introduced, i.e., $M_i^{k_1, k_2}(T)$. For example, $M_{John}^{1,3}(three)$ means that there is 1 vision satisfying *three* compared to 3 all visions of John (c.f. Fig. II).

2.2 Persuade to Change Uncertainty

Degrees of beliefs are valuable tools that allow for examining whether and how much persuasion is successful. Let us come back to the example where John believes that the key 3 opens the safe with the degree $\frac{1}{3}$. Now, assume that Peter wants to exchange keys with John. Therefore he proposes to give John the key 4 in return for the key number 3. In order to do this he tries to convince John that such an exchange is profitable. The argument, call it *a*, is: “I have heard that a key with even number opens the safe and it is the key number 4. I can give it to you in return for the key number 3”. In consequence of such an argumentation, John allows for a possibility that key 4 is the desired key, but he still believes that the key with odd number may be the right one. Thereby he has four visions of the current state, three old ones: $(1, 3|2, 4, 5|1)$, $(1, 3|2, 4, 5|3)$, $(1, 3|2, 4, 5|5)$ and a new one: $(1, 3|2, 4, 5|4)$. The degree of his belief that 3 is a right key decreases from $\frac{1}{3}$ to $\frac{1}{4}$. As a result, John may suppose that the key number 3 is useless and agree for the exchange.

Now we can evaluate **effectiveness** of argument *a*. If the persuader is satisfied only when John is absolutely sure that the key 3 does not open the safe then the argument is not successful. It could be also the case that any fall of beliefs’ degree is desired. Then, the argument is successful. Obviously, in our example - the lower degree, the better result is. So, if there exists an argument which causes the larger

fall, e.g. to $\frac{0}{4}$, then we consider it as more persuasive. Degrees of uncertainty make it possible to **describe** effects of arguments as well as **compare** their success.

In \mathcal{AG}_n logic, the changes caused by actions are described by operator \diamond . For example, the formula $\diamond(a : Peter)M_{John}^{1,4}(three)$ means that if Peter provides the argument a then John may believe with the degree $\frac{1}{4}$ that the key 3 opens the safe.

2.3 Formal Framework

Now we shortly describe formal syntax and semantics of \mathcal{AG}_n . For more details see [3,4]. Let $Agt = \{1, \dots, n\}$ be a set of names of *agents*, V_0 be a set of *propositional variables*, and Π_0 a set of *program variables*. Further, let $;$ denote a programme connective which is a sequential composition operator. It enables to compose *schemes of programs* defined as the finite sequences of atomic **actions**: $a_1; \dots; a_k$. Intuitively, the program $a_1; a_2$ for $a_1, a_2 \in \Pi_0$ means “Do a_1 , then do a_2 ”. The set of all schemes of programs we denote by Π . The set of all **well-formed expressions** of \mathcal{AG}_n is given by the following Backus-Naur form:

$$\alpha ::= p | \neg\alpha | \alpha \vee \alpha | M_i^d \alpha | \diamond(i : P)\alpha,$$

where $p \in V_0$, $d \in \mathbb{N}$, $P \in \Pi$, $i \in Agt$.

Other Boolean connectives are defined from \neg and \vee in the standard way. We use also the following abbreviations: $\Box(i : P)\alpha$ for $\neg\diamond(i : P)\neg\alpha$, $B_i^d\alpha$ for $\neg M_i^d\neg\alpha$, $M_i^d\alpha$ where $M_i^0\alpha \Leftrightarrow \neg M_i^0\neg\alpha$, $M_i^d\alpha \Leftrightarrow M_i^{d-1}\alpha \wedge \neg M_i^d\neg\alpha$, if $d > 0$, and $M_i^{d_1, d_2}\alpha$ for $M_i^{d_1}\alpha \wedge M_i^{d_2}(\alpha \vee \neg\alpha)$. Formula $\Box(i : P)\alpha$ is defined as $\neg\diamond(i : P)\neg\alpha$. We use also the formula $M_i^d\alpha$ where $M_i^0\alpha \Leftrightarrow \neg M_i^0\neg\alpha$, $M_i^d\alpha \Leftrightarrow M_i^{d-1}\alpha \wedge \neg M_i^d\neg\alpha$, if $d > 0$. Moreover, we introduce the formula $M_i^{d_1, d_2}\alpha$ which is an abbreviation for $M_i^{d_1}\alpha \wedge M_i^{d_2}(\alpha \vee \neg\alpha)$. It should be read as “ i believes α with the degree $\frac{d_1}{d_2}$ ”. Thereby, by a **degree of beliefs** of agents we mean the ratio of d_1 to d_2 , i.e. the ratio of the number of states which are considered by an agent i and verify α to the number of all states which are considered by this agent. It is easy to observe that $0 \leq \frac{d_1}{d_2} \leq 1$.

Definition 1. Let Agt be a finite set of names of agents. By a semantic model we mean a Kripke structure $\mathcal{M} = (S, RB, I, v)$ where

- S is a non-empty set of states (the universe of the structure),
- RB is a doxastic function which assigns to every agent a binary relation, $RB : Agt \longrightarrow 2^{S \times S}$,
- I is an interpretation of the program variables, $I : \Pi_0 \longrightarrow (Agt \longrightarrow 2^{S \times S})$,
- v is a valuation function, $v : S \longrightarrow \{\mathbf{0}, \mathbf{1}\}^{V_0}$.

Function I can be extended in a simple way to define interpretation of any program scheme. Let $I_\Pi : \Pi \longrightarrow (Agt \longrightarrow 2^{S \times S})$ be a function defined by mutual induction on the structure of $P \in \Pi$ as follows: $I_\Pi(a)(i) = I(a)(i)$ for $a \in \Pi_0$ and $i \in Agt$, $I_\Pi(P_1; P_2)(i) = I_\Pi(P_1)(i) \circ I_\Pi(P_2)(i) = \{(s, s') \in S \times S : \exists s'' \in S ((s, s'') \in I_\Pi(P_1)(i) \text{ and } (s'', s') \in I_\Pi(P_2)(i))\}$ for $P_1, P_2 \in \Pi$ and $i \in Agt$.

The **semantics** of formulas is defined with respect to a Kripke structure \mathcal{M} .

Definition 2. For a given structure $\mathcal{M} = (S, RB, I, v)$ and a given state $s \in S$ the Boolean value of the formula α is denoted by $\mathcal{M}, s \models \alpha$ and is defined inductively as follows:

$$\begin{aligned} \mathcal{M}, s \models p & \quad \text{iff } v(s)(p) = \mathbf{1}, \text{ for } p \in V_0, \\ \mathcal{M}, s \models \neg\alpha & \quad \text{iff } \mathcal{M}, s \not\models \alpha, \\ \mathcal{M}, s \models \alpha \vee \beta & \quad \text{iff } \mathcal{M}, s \models \alpha \text{ or } \mathcal{M}, s \models \beta, \\ \mathcal{M}, s \models M_i^d \alpha & \quad \text{iff } |\{s' \in S : (s, s') \in RB(i) \text{ and } \mathcal{M}, s' \models \alpha\}| > d, d \in \mathbb{N}, \\ \mathcal{M}, s \models \diamond(i : P)\alpha & \quad \text{iff } \exists s' \in S ((s, s') \in I_{\Pi}(P)(i) \text{ and } \mathcal{M}, s' \models \alpha). \end{aligned}$$

We say that α is true in a model \mathcal{M} at a state s if $\mathcal{M}, s \models \alpha$.

In [3] we showed the sound axiomatization of the logic \mathcal{AG}_n and proved its completeness.

3 Extension of \mathcal{AG}_n

In this section we show the extension for the model of \mathcal{AG}_n .

3.1 Content and Goal of Persuasive Actions

In formal models, persuasive actions are often represented in terms of **illocutionary acts** [2,16]. For instance, the game-theoretic models such as [9,10,11,14] allows different types of speech acts that can be executed during the persuasion dialogs e.g. claims, questions, argumentations (“since”), etc. In the model of persuasive negotiation [15], there are two types of persuasive actions - illocutionary and environmental ones.

Each illocutionary act consists of its **content** and force. In the resource re-allocation scenario, when Peter says to John that the key 3 does not open the safe, he may send this propositional content (i.e. “the key 3 does not open the safe”) with a different illocutionary force (intention): informing, assuring, conjecturing, warning, persuading, etc. According to the Searle and Vanderveken’s account, there are seven components of the illocutionary force [17]. However, the most important is its **goal** called illocutionary point. For example, the characteristic aim of an assuring is to describe how things are, while the goal of arguing is both to assure and give reasons which support what is assured. For instance, when Peter sends a message “the key 3 does not open the safe”, he may have different goals. He may want John to believe that the key 3 is not the right one, or he may intend to make him trust in Peter’s cooperation, or to discourage John from asking about this key. Depending on the particular applications, we could need to be able to express different goals that agents may have in persuasion, in particular when they communicate and negotiate about the desired resources.

3.2 Verbal and Nonverbal Arguments

When we research the effectiveness of persuasive arguments, we can be interested in differences in the result of a message that depends on the means by which

this message was sent. In particular, we can focus on the differences between verbal and nonverbal arguments. We understand **verbal communication** as communication executed with use of words (not as communication executed orally). Generally, there are two types of **nonverbal communication**: messages produced by the body (general appearance and dress, body movement, posture, gestures, facial expressions, touch, etc.) and messages produced by the environment (time, space, silence, etc.).

Consider the problem of resource re-allocation. Depending on the application, we may want to differentiate reactions of agents in respect of the way the persuasive action is executed. That is, Peter can send the propositional content “I can give you the key number 4” in verbal or nonverbal manner. We could want to express that the result of the verbal promise is different (e.g. less persuasive) than the result of the actual act of giving the key to John. Observe that the content and the goal of those persuasive actions may remain the same. The only parameter that can change is the means of sending the message.

3.3 Which vs. Why Successful

\mathcal{AG}_n logic was applied for reasoning about persuasiveness of actions, i.e., we examined **which** arguments are successful and how big such a success is. Now we propose an extension of our approach in order to study **why** these arguments are efficient and what attributes cause their success.

Assume that every persuasive action is represented by 3-tuple (m, β, δ) which fixes a content of a message m sent in the action, a goal α of executing action and the way it is performed. Formally, we define the set of persuasive actions $\Pi_p \subseteq \Pi_0$ as follows:

$$\Pi_p = \{(m, \beta, \delta) : m \in C, \beta \in F, \delta \in \Delta\}$$

where C is a set of contents, F is a set of formulas of \mathcal{AG}_n and Δ is a set of symbols representing means of actions, i.e., ways they can be performed. This set can consist of the elements such as: *ver* - for verbal actions, *nver* - for nonverbal actions.

A goal of an action is expressed by a formula of \mathcal{AG}_n . We say that the goal is **achieved** if after execution of the action, a state in which it is satisfied is reached (a formula expressing the goal is true in this state). For example, let a goal of an action $a = (m, \beta, \delta)$ executed by Peter be to convince John with the degree $\frac{1}{4}$ that the right key is 3. That is, β is a formula $M_{John}^{1,4}(three)$. So, the goal of the action is achieved if there exists a state s such that s is a result of a and $s \models M_{John}^{1,4}(three)$.

The **success** of an action strongly depends on its content. For instance, to achieve the goal $\beta = M_{John}^{1,4}(three)$ Peter can say “I have heard that a key with an even number opens the safe and it is the key number 4” or he can say “I know that the key number 5 does not open the safe”. As we showed in section 2.2 the action with the first content may cause that β will be true. If we now assume that words “key number 5 does not open the safe” are disregarded

by John then the action with this content will not be successful. Formally, the action $a = (\mathbf{m}, \beta, \delta)$ with $m =$ “a key with an even number opens the safe and it is the key number 4”, $\beta = M_{John}^{1,4}(three)$ and $\delta = ver$ can achieve its goal while the action $a_1 = (\mathbf{m}_1, \beta, \delta)$ with $m_1 =$ “the key number 5 does not open the safe”, $\beta = M_{John}^{1,4}(three)$ and $\delta = ver$ can not achieve the goal. In this way, we can analyze how different contents of actions may influence the result of agents activity.

The other advantage of our approach is that we can distinguish effectiveness of actions with the **same content** but sent by **different means**. Suppose that the goal of Peter is to reach a situation in which John is convinced that the key number 5 does not open the safe. He may send this content in different ways. He may say “the key 5 does not open the safe” or do a nonverbal action in which he takes the key and tries to open the safe. The first action is $a_1 = (m_1, \beta_1, \delta_1)$, where m_1 expresses “the key 5 does not open the safe”, β_1 is a formula $M_{John}^{1,4}(five)$ (*five* means that the key 5 opens the safe), and δ_1 is a symbol *ver* pointing that it is a verbal action. The second action is $a_2 = (m_1, \beta_1, \delta_2)$, where m_1 and β_1 are as above and δ_2 is a symbol *nver* pointing that it is a nonverbal action. Verbal argument a_1 can bring a poor result - John disregards Peter’s words and still believes that 5 may be the right key. Nonverbal argument with the same content is very effective - when John seeing that the key 5 does not fit the safe he must believe that 5 is not the right key. Formally, the difference is in a state of a multiagent system which is reached depending on the way the action is performed.

The representation of arguments by identifying their content, goal and means significantly facilitates the recognition of the attributes which influence the success the most.

4 Software Tool Perseus

Using the \mathcal{AG}_n logic’s formalism, a computer tool called **Perseus** is being developed. The aim of Perseus system is to analyze properties of multiagent systems concerning persuasion process. In this case the **parametric verification** of an input question is done [5]. This approach enables to find an answer to three basic types of questions: how a given argumentation can influence a degree of someone’s beliefs about a given thesis, whose beliefs are changing during a given argumentation and finally which argumentation should be executed to change someone’s beliefs in a specific way. As soon as the Perseus tool gives an answer to a question, we can carry out more detailed research. For example, we can check whether an argumentation is successful, how big such a success is or determine optimal argumentation with regard to its length.

Now we widen the capabilities of the Perseus system with the enriched \mathcal{AG}_n formalism. In the extended model, a set of persuasive actions is specified. Therefore more information can be received from answers of questions. Furthermore, we can verify new properties of multiagent systems. For example, it is possible to test if an action may be **successful** (may satisfy its goal):

$$\diamond((m, \beta, \delta) : i)\beta.$$

Informally the formula says that it is possible that the result of an action (m, β, δ) , the goal of which is β , performed by an agent i is β , i.e., if the agent i executes the action (m, β, δ) in a state s then the system moves to a state s' such that $s' \models \beta$. For instance,

$$\diamond((m, M!_{John}^{1,4}(three), ver) : Peter)M!_{John}^{1,4}(three),$$

where content m is “the key 4 opens the safe”. The formula means that Peter can convince John that the key number 3 opens the safe with degree $\frac{1}{4}$. If we want to exchange the “possibility” into “necessity” it is sufficient to exchange the operator \diamond into the operator \square .

Another property we can verify is that the **same content** sending with the same goal but by **different means** brings about two different results:

$$\begin{aligned} &\diamond((m, M!_{John}^{1,4}(three), \mathbf{nver}) : Peter)M!_{John}^{1,4}(three) \wedge \\ &\neg\diamond((m, M!_{John}^{1,4}(three), \mathbf{ver}) : Peter)M!_{John}^{1,4}(three). \end{aligned}$$

Informally the property expresses that Peter’s nonverbal action of sending m is successful, while verbal action of sending m is not. Analogously, it is possible to test which content can cause the success assuming that the goal and means are the same:

$$\diamond((\mathbf{m}_1, \beta, \delta) : i)\beta \wedge \neg\diamond((\mathbf{m}_2, \beta, \delta) : i)\beta.$$

5 Conclusions

The extension of \mathcal{AG}_n logic proposed in this paper enables to investigate into the wider class of research questions concerning content, goal and means of sending messages in persuasive actions. In the persuasion dialog systems as well as in the pure \mathcal{AG}_n logic we can examine properties of multiagent systems referring to those parameters of persuasion. Nevertheless, the detailed analysis can be conducted only in the metalanguage or the natural language. This means that when an action (argument) terminates with success it is possible to indicate its various attributes and study interactions among them. However, it could be done from the meta-level. The strong point of the extended \mathcal{AG}_n logic is that goal, content and means are directly included in its language. As a result we can express properties as formulas and next we can automatically verify these properties by means of software tool such as Perseus.

Furthermore, observe that in the persuasion dialog systems the goal of persuasion is either resolution of a conflict or persuading an adversary. Assume that Paul and Olga discuss the topic of safety of Paul’s car (the example given in [14]). The only goal, which is considered in this approach, is Paul and Olga’s cognitive attitude towards the topic, i.e., whether they believe that his car is safe or not. However, the real goal of the dialog could be that Paul wants to influence Olga’s behavior, e.g., to take her for a ride or make her buy the car. This could be accomplished with the use of the extended \mathcal{AG}_n logic.

References

1. Atkinson, K., Bench-Capon, T., McBurney, P.: Towards a computational account of persuasion in law. In: Proceedings of the Ninth International Conference on AI and Law (ICAIL 2003), pp. 22–31 (2003)
2. Austin, J.: *How to Do Things with Words*. Clarendon, Oxford (1962)
3. Budzynańska, K., Kacprzak, M.: A logic for reasoning about persuasion. *Fundamenta Informaticae* 85, 51–65 (2008)
4. Budzynańska, K., Kacprzak, M., Rembelski, P.: Modeling persuasiveness: change of uncertainty through agents' interactions. In: Proc. of COMMA. *Frontiers in Artificial Intelligence and Applications*. IOS Press, Amsterdam (2008)
5. Budzynańska, K., Kacprzak, M., Rembelski, P.: Perseus. software for analyzing persuasion process. *Fundamenta Informaticae* (2009)
6. Feldmann, R., Gairing, M., Lücking, T., Monien, B., Rode, M.: Selfish routing in non-cooperative networks: A survey. In: Rovan, B., Vojtáš, P. (eds.) *MFCS 2003*. LNCS, vol. 2747, pp. 21–45. Springer, Heidelberg (2003)
7. Hussain, A., Toni, F.: Bilateral agent negotiation with information-seeking. In: Proc. of the 5th European Workshop on Multi-Agent Systems (2007)
8. Jonker, G.M., Meyer, J.J., Dignum, F.: Market mechanisms in airport traffic control. In: Proc. of the Second European Workshop on Multiagent Systems, EUMAS 2004 (2004)
9. McBurney, P., Parsons, S.: Games that agents play: A formal framework for dialogues between autonomous agents. *Journal of Logic, Language and Information* (13), 315–343 (2002)
10. Parsons, S., Sierra, C., Jennings, N.R.: Agents that reason and negotiate by arguing. *Journal of Logic and Computation* 8(3), 261–292 (1998)
11. Parsons, S., Wooldridge, M., Amgoud, L.: An analysis of formal interagent dialogues. In: Proc. of AAMAS, pp. 394–401 (2002)
12. Parunak, H.V.D.: *Applications of distributed artificial intelligence in industry*. John Wiley and Sons, Inc., Chichester (1996)
13. Petty, R.E., Cacioppo, J.T.: *Attitudes and Persuasion: Classic and Contemporary Approaches*. Westview Press (1996)
14. Prakken, H.: Formal systems for persuasion dialogue. *The Knowledge Engineering Review* 21, 163–188 (2006)
15. Ramchurn, S.D., Jennings, N.R., Sierra, C.: Persuasive negotiation for autonomous agents: A rhetorical approach. In: *IJCAI Workshop on Computational Models of Natural Argument*, Acapulco, Mexico (2003)
16. Searle, J.: *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge (1969)
17. Searle, J., Vanderveken, D.: *Foundations of Illocutionary Logic*. Cambridge University Press, Cambridge (1985)

A Hybrid Method of Indexing Multiple-Inheritance Hierarchies

Jacek Lewandowski and Henryk Rybinski

Institute of Computer Science, Warsaw University of Technology
{j.lewandowski,hrb}@ii.pw.edu.pl

Abstract. The problem of efficient processing of the basic operations on ontologies, such as subsumption checking, or finding all subtypes of a given type, becomes of a very high importance. In the paper we present a hybrid approach of organizing multi-hierarchical structures, combining numbering schemes [1], [13] with “gene-based” methods [10], [17]. The proposed method generalizes earlier solutions and inherits advantages of earlier approaches. The obtained structure preserves the feature of incremental changes of the ontology structure. The experiments performed show significant efficiency results in accessing ontology resources for performing processes specific for semantic web.

1 Introduction

Ontologies have shown their usefulness in many application areas, especially in intelligent information integration, information retrieval and brokering or natural-language processing, including knowledge discovery and text mining. In all these applications one of the main issue is to reuse domain-specific knowledge coded in a multiple inheritance hierarchy. An efficient encoding of the hierarchical dependencies is of paramount importance in these applications.

Multiple-inheritance hierarchies can be found in class-based object programming languages, as the type-subtype principle for modeling inheritance, and to more extent in object oriented databases [12]. Traversing the graph for finding a path from a computational point of view is an expensive operation. To this end the problem of fast type-subtype checking was considered in many publications [1,2,3,4,6,10,13,17,20].

From the mathematical point of view such hierarchies are simple directed acyclic graphs (DAG) [5]. The aim of indexing is to assign a special code to each node, which allows to effectively: (1) find out if there is a path from one node to another, (2) retrieve all nodes which a path exists to and (3) find the least common ancestor of the nodes. A typical way to store such relationship is enumerating parents of each node. For most hierarchies this method is memory efficient – $O(n)$, however it is not time efficient in general – the worst-case reaches $O(n)$ for single check [3].

In the area of the object oriented programming or even database modeling the multi-hierarchical structures are not that large as in the case of ontologies. This is therefore much more important to find out methods for fast type-subtype checking relevant for the ontology sized structures. Ontologies are knowledge

bases which may contain huge number of connected elements. A common type of such connection is a taxonomy relation [15]. Recently, growing interest in this area have been shown [11,13,17].

In this paper we present a method suitable for indexing the ontology taxonomies structures. The next section provides a related work on encoding hierarchies. Then we introduce some basic notions and describe our hybrid approach. In Section 4 we present results of the experiments we have done. We finish with the conclusions.

2 Related Work

The interest in methods of indexing hierarchical structures goes back to 70-ties of the previous century [14]. In [7,11] a method for static tree-like hierarchies is presented. It is an interval based approach, which involves creating two indexes that somehow constrain two-dimensional plane. The inheritance relation is checked by inclusion testing of areas represented by the two nodes. The solution is very attractive especially in static tree-like hierarchies – Dietz [7] has shown that we can create such indexes by pre-order traversing a hierarchy to create the first index, and then post-order traversing to create the second one. Though the usefulness for single inheritance hierarchies has been proven, these numbering schemes are not sufficient for multiple inheritance hierarchies. The approach was further developed towards the multi-hierarchical applications in [13,13,20].

In [2,6,10,17] a new approach to an efficient organization of the hierarchical structures is considered. Opposite to [7], these methods tend to encode the set inclusion relation, based on inheriting some properties by the "child nodes" from their "parents". An important feature is that the methods already allow multi-hierarchy structures. In particular, the method described in [10] is closer to the efficiency requirements for taxonomies in ontologies. Recently, a more efficient method has been described in [17], where an idea of using prime number as "genes" has been used.

In the sequel we will present a combination of the methods based on numbering schemes [17] and the ones based on the genes inheritance. The approach generalizes both types of methods in the sense that the node code may consists of an arbitrary number of indexes and "genes", so that the coding structure can be adjusted to the specific needs of the hierarchy.

3 Hierarchy Encoding

3.1 Basic Concepts

We present an ontology as a directed acyclic graph $G(\Omega, \prec_d)$ with Ω as the set of classes (concepts) and \prec_d the set of direct ISA relationships between the classes. We define a multiple inheritance hierarchy $H(\Omega, \prec)$, where \prec is an inheritance relation defined as follows: $x \prec y$ if $x \prec_d y$ or for some x_1, \dots, x_n , $n > 0$, we have $x \prec_d x_1 \prec_d \dots \prec_d x_n \prec_d y$. The inheritance relation is a strict partial

order, so it is transitive, asymmetric, and irreflexive. We also define the relation \preceq as follows: $x \preceq y$ iff $x = y$ or $x \prec y$. Given x , we also define descendants, ancestors, parents and children of x (and denote them by $D(x)$, $A(x)$, $P(x)$, and $C(x)$ respectively) as follows:

$$\begin{aligned} D(x) &= \{y : y \in \Omega \wedge y \prec x\} \\ A(x) &= \{y : y \in \Omega \wedge x \prec y\} \\ P(x) &= \{y : y \in \Omega \wedge x \prec_d y\} \\ C(x) &= \{y : y \in \Omega \wedge y \prec_d x\} \end{aligned}$$

In addition for $G = (\Omega, \prec_d)$ we define the notions of roots and leaves (R and L respectively):

$$\begin{aligned} R &= \{x : x \in \Omega \wedge A(x) = \emptyset\} \\ L &= \{x : x \in \Omega \wedge D(x) = \emptyset\} \end{aligned}$$

Let us introduce the notion of index function. Given a total function $f : \Omega \rightarrow \mathbb{R}$, \mathbb{R} , is a set of real numbers, we call it index if it preserves the order in hierarchy H , i.e. $a \prec b \Rightarrow f(a) > f(b)$.

3.2 The Idea

The presented approach combines two ways of encoding, i.e. the one based on using indexes (generalizing the Dietz approach [7], and another one based on coding the set-theoretic inclusion relation [10]. We will show that combining these two ways of encoding we can reduce the size of coding the hierarchy, and on the other hand we can speed up all the basic operations on the hierarchy [2].

First, let us consider the indexing approach. The best solution would be to create an indexing system f for H , so that the relationship between the indexes reflects the order of the relation \prec , i.e. $f(a) > f(b) \Leftrightarrow a \prec b$.

Unfortunately for (poly)hierarchical structures there is no such an indexing system. We can however find a number of indexing systems f_i so, that for each of them we have

$$f_i(a) > f_i(b) \Leftrightarrow a \prec b \tag{1}$$

So with one index only, given $f_i(a) > f_i(b)$, we cannot decide if the relation $a \prec b$ holds. On the other hand we can build more indexes, say f_i and f_j , such that

$$a \not\prec b \wedge b \not\prec a \Leftrightarrow f_i(a) > f_i(b) \wedge f_j(a) < f_j(b) \tag{2}$$

To this end, we should have a set of indexes \mathcal{F} , satisfying (2), so that the following is hold:

$$\forall a, b \in \Omega (\forall f \in \mathcal{F} : f(a) > f(b)) \Leftrightarrow a \prec b \tag{3}$$

Some numbering schemes addressing this problem, are presented in [7][8]. Dietz has shown a method to encode tree-like structures with 2 indexes. Still the method is not able to handle multi-inheritance hierarchies. Formally we may not fulfill (3) with $\overline{\mathcal{F}} = 2$. The example below clarifies the problem.

In the Dietz's numbering scheme we conclude that node a is ancestor of b iff $f_1(a) \leq f_1(b) \wedge f_2(a) \leq f_2(b)$. If though we look at nodes C and F on Fig. 1 we see that the above rule leads us to the conclusion $F \prec C$, which is false.

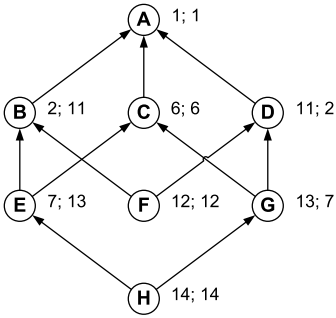


Fig. 1. Dietz numbering

Such pairs of nodes which invalidate (3) will be call unresolved false-positive conflicts or simply - unresolved conflicts. We provide a formal definition below.

Definition 1: Given a set of indexes \mathcal{F} , a pair of nodes invalidating (3) is called F -conflicting. If \mathcal{F} is obvious we also call it unresolved conflict.

Some workarounds of the conflict problem were presented in [1,13,20]. In the sequel we propose a more general solution. Our numbering scheme does not impose number of indexes. The only requirement is that they must correspond to the

topological order of the hierarchy. In other words, we devise a method to create a family of indexes \mathcal{F} that satisfy (3). Moreover, to achieve an efficient encoding, we want also the number of indexes to be as low as possible, that is, we want to maximize the number of unresolved conflicts covered by each index.

Assume that there are no indexes assigned to any node. So we have to create indexes which allow determining ancestor – descendant relations and also as we put on before, indexes have to reflect a topological order of the hierarchy. A simple graph can be used to represent our assumptions - Fig. 2. Edges are directed from the nodes which must be numbered with lower value to the ones with higher value. First, we create edges that reflect topological constraints (dotted lines on the picture). As the hierarchy is a directed acyclic graph, the edges cannot make any cycle. Next, according to formula (1) we create edges in both directions between the nodes, which are not in relation in any direction we call them conflict edges (thin solid lines on the picture). Now, to each node we have to assign a consecutive number, so that the created sequence reflects a topological order of the created graph – we can use depth first traversal algorithm. As the next step, we remove the cycles resulting from the above procedure by deleting particular conflict edges but all edges reflecting the topological order must remain. After removing all the cycles we can number the nodes in the modified graph (dotted and thick solid lines on Fig. 2). The obtained sequence constitutes the first index. To create the next one, again we start with edges reflecting the topological order of the hierarchy. Then we add those conflict edges which have been removed before, and try to eliminate all cycles. We repeat the steps until there are no remaining conflict edges.

The algorithm sketched above can solve all the conflicts. However, we can leave a number of conflicts for solving them by coloring the graph. In the next subsection we will show how to encode poly-hierarchic structures by coloring the graph, and then we show how to combine the two methods.

3.3 Hierarchy Coloring

We consider the methods, which generate codes consisting of some elements (colors or genes), and for which the inheritance relation is determined by the

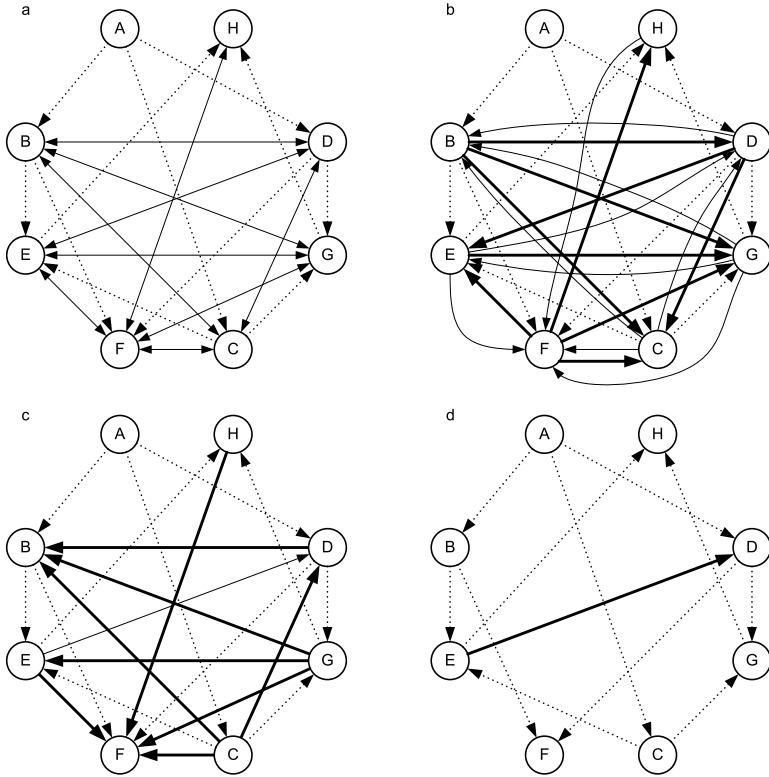


Fig. 2. A sequence of graphs used to create three indexes; (a) an initial state (b) a graph for the first index; (c) a graph for the second index; (d) a graph for the third index

set-theoretic inclusion test. Generally, these methods can be divided into two groups – conflict-free encodings and non-conflict-free encodings [17].

Let us define them formally. Given a basic set Φ of colors (*genes*), we define the color function $g : \Omega \rightarrow \Phi$, which assigns colors to individual nodes. Now we also define the *inheritance code function* $\gamma : \Omega \rightarrow 2^\Phi$, so that the following condition holds:

$$d \preceq a \Leftrightarrow \gamma(d) \supseteq \gamma(a) \tag{4}$$

Given d , we call $\gamma(d)$ the code of d . To achieve (4) we construct the (total) function γ in the way that the colors assigned by g are propagated down to the all subsequent descendants, i.e.

$$\gamma(x) = \{g(y) : y \in \Omega, x \preceq y\} \tag{5}$$

This gives a simple way to check if d inherits from a , or not. By rewriting (4) we have:

$$d \not\preceq a \Leftrightarrow \gamma(d) \not\supseteq \gamma(a) \tag{6}$$

$$\gamma(d) \not\supseteq \gamma(a) \Leftrightarrow \exists \varphi \in \gamma(a) : \varphi \notin \gamma(d) \tag{7}$$

and by inserting (5) to (7) we obtain:

$$d \not\preceq a \Leftrightarrow \exists \varphi \in \gamma(a) \forall x, d \preceq x : \varphi \neq g(x) \tag{8}$$

So, to be able to apply (8) for a the value of $\gamma(a)$ cannot be empty set. This can be assured by defining g for those nodes. So, the domain $\Gamma, \Gamma \subseteq \Omega$, of the coloring function g can be defined, as follows:

$$\Gamma = \{x : \exists d \in \Omega : d \not\preceq x\} \tag{9}$$

There are some scenarios of constructing the function g affecting its properties:

- g is a total one-to-one mapping – a conflict-free encoding that is used in binary matrices, and in [17]; codes are longer, but it is easier to encode new nodes incrementally;
- g is a partial many-to-one mapping – a non-conflict-free encoding, presented in [10]; short codes, but incremental encoding is much more complex;
- g is a partial one-to-one mapping – possibly a compromise between the cases mentioned above, however we have not found any of its applications;
- g is a total many-to-one mapping – probably the least efficient solution, giving no advantages of the first two cases.

We will focus on the second solution, as it is most efficient: (1) it can reuse some colors for the nodes; and (2) the function g is not total, so not all nodes have to have assigned colors. Below we explain how to create the required mapping. As mentioned above, the right side of (8) may be satisfied by assigning a color φ to each node a from Γ :

$$\forall a, d : d \not\preceq a \forall x, d \preceq x : g(a) \neq g(x) \tag{10}$$

Now, based on (10), we define a conflict graph whose edges connect nodes which a value of the function must be different for:

Definition 2: Given a graph $G_C(\Gamma, E)$, where $E = \{(x, y) \in \Gamma \times \Gamma : \exists z : z \not\preceq x \wedge z \preceq y\}$, we call it a conflict graph.

To construct g we have to color vertices of G_C , and use the obtained colors as values of g . Then, we compute γ for each node, according to (5). The presented method is a top-down encoding, as each child node inherits all colors of its parents. It is easy to transform the above formulas to achieve bottom-up encoding, where a parent node inherits colors of its children.

3.4 Hybrid Approach

The presented encodings hierarchy numbering and hierarchy coloring – may coincide. We can create a number of indexes according to the first solution, leaving some conflicts unresolved, and then solve them with the other method. It can be particularly useful, as each type of encoding has specific features. With a hybrid approach we can reduce drawbacks while keeping profits.

In our method the rule for inheritance testing has the following form: $a \preceq b \Leftrightarrow \gamma(b) \subseteq \gamma(a) \wedge \forall f_i \in \mathcal{F} : f_i(a) > f_i(b)$. As we negate it we will have: $a \not\preceq b \Leftrightarrow \gamma(b) \not\subseteq \gamma(a) \vee \exists f_i \in \mathcal{F} : f_i(a) \leq f_i(b)$, so (6) has to hold in a limited domain:

$$\forall a, b \in \Omega \forall f_i \in \mathcal{F} : f_i(a) > f_i(b) : a \not\preceq b \Leftrightarrow \gamma(b) \not\subseteq \gamma(a) \tag{11}$$

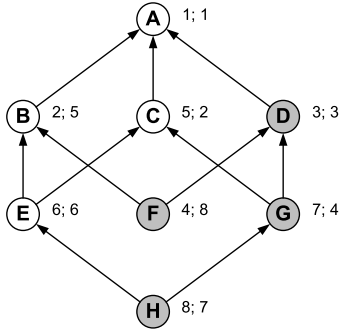


Fig. 3. A hierarchy encoded with hybrid method

Next, we follow the rules from the previous subsection to achieve a conflict graph. Formula (11) is a clue of the method. It shows how to make use of both types of codes. We will show how we can leverage the approach to encode a sample hierarchy.

Given a hierarchy as on Fig. 1, encoded with two indexes, we still have unresolved conflict between E and D . We solve it by coloring the hierarchy. According to Definition 2 we create a set Γ , which must be colored. In this particular case $\Gamma = \{D\}$. So we have to color it and compute.

4 Testing

We have tested the algorithm on several hierarchies – both random and real ones. Our aim was to explore encoding capabilities such as minimum, maximum and average number of bits required to create a code for each node. The testing environment was implemented on a PC (Intel Core2Duo). The algorithm is programmed with Java and for a taxonomy with about 32000 nodes it can perform 10^7 inheritance tests per second (so the platform is very likely to run significantly faster when implemented in C or C++).

4.1 Random Hierarchies

Random hierarchies have been prepared as follows. As an input for it we provide the following parameters: (1) a starting number of nodes, (2) minimum, (3) maximum, (4) average and (5) standard deviation of children per node, (6) an average number of parents per node and (7) the number of levels. With these parameters the structures were generated randomly – we used a Gaussian distribution with parameters (4) and (5), bounded by (2) and (3), making the appropriate connections. For the generated graph all the parent-less nodes have been finally connected to the single root node, added to the hierarchy.

4.2 Tests Results

Number of parents

First, we have tried to examine how number of parents impacts average encoding length. We have created 12 random hierarchies, with 6 levels and 3906 nodes. They differ in average number of parents/children of a single node.

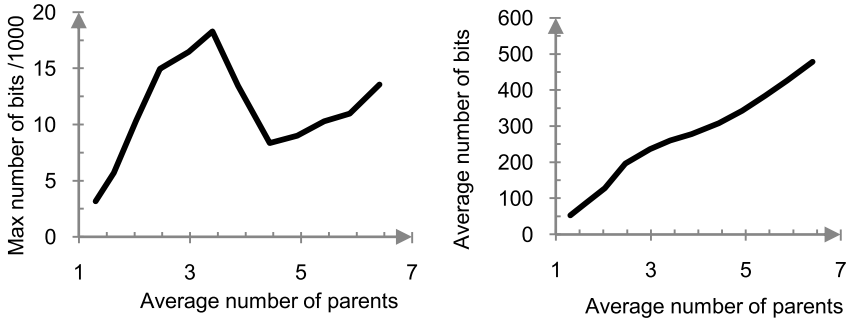


Fig. 4. Maximum and average number of bits in respect to average number of parents

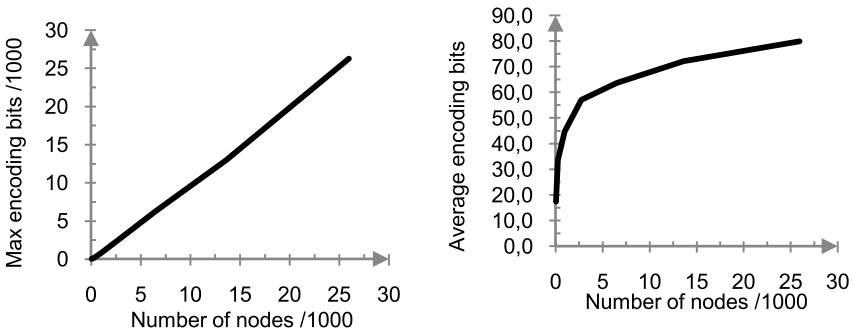


Fig. 5. Maximum and average number of encoding bits in respect to number of nodes

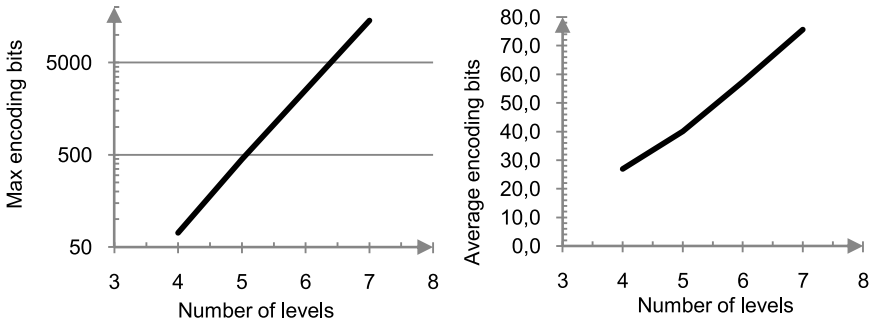


Fig. 6. Maximum and average number of nodes in respect to number of levels in hierarchy

Number of nodes

The second test has been performed to find a relationship between the total number of nodes and the average number of bits needed to encode each node. All hierarchies have 6 levels and approximately constant number of parents of a single node.

Number of levels

The last test of random hierarchies has been made to discover a relationship between a number of levels and an average encoding length. Every node in each tested hierarchy has more or less the same number of parents and about the same number of children.

Real world hierarchies

We have also tested some real world taxonomies, from such ontologies like Pizza [16], SUMO [19] and Gene Ontology [8]. The results are shown below. As far as Pizza and Gene ontologies are concerned, we could not compare the results because we had used different versions, though for the SUMO ontology our results outperforms the results of the other methods in particular the one presented in [17].

Name	Number of nodes	Number of levels	Number of children		Number of parents		Number of encoding bits		
			Max	Avg	Max	Avg	Min	Max	Avg
Gene	31882	16	5088	3,5	7	1,6	43	38588	113,8
Pizza	152	7	68	3,7	10	2,1	22	160	35,5
SUMO	630	16	15	2,8	3	1,1	24	44	26,5

5 Conclusions

We have presented a novel approach in encoding multi-hierarchical structures. It is a combination of two well-known methods. The experiments show that it allows to achieve better results in terms of memory and time requirements, comparing to the original solutions.

There are still few ways to improve our algorithm. First, we expect that finding a proper heuristics in deciding at which point the numbering method can be stopped and the coloring one should start. In general, the tree substructures are better for indexing whereas the subgraphs, where nodes have more parents are better for coloring. Moreover, we expect that finding a good heuristic to remove conflicting edges from a conflict graph would make it possible to obtain shorter encodings. We suppose also that using PQ-trees [9,20] would be an alternative way to achieve this goal.

Acknowledgments. The research has been supported by grant No. N N516 375736 received from Polish Ministry of Education and Science.

References

1. Agrawal, R., Borgida, A., Jagadish, H.V.: Efficient management of transitive relationships in large data and knowledge bases. In: Clifford, J., Lindsay, B., Maier, D. (eds.) 1989 ACM SIGMOD, pp. 253–262. ACM, New York (1989)
2. Ait-Kaci, H., Boyer, R., Lincoln, P., Nasr, R.: Efficient implementation of lattice operations. ACM Trans. Program. Lang. Syst. 11(1), 115–146 (1989)

3. Alavi, H.S., Gilbert, S., Guerraoui, R.: Extensible encoding of type hierarchies. In: 35th Ann. ACM SIGPLAN-SIGACT Symp. Principles of Programming Languages, pp. 349–358 (2008)
4. Baehni, S., Barreto, J., Eugster, P., Guerraoui, R.: Efficient distributed subtyping tests. In: 2007 Inaugural Int'l Conf. on Distributed Event-Based Systems, vol. 233, pp. 214–225. ACM, New York (2007)
5. Bender, M.A., Pemmasani, G., Skiena, S., Sumazin, P.: Finding least common ancestors in directed acyclic graphs. In: 12th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 845–854. SIAM, Philadelphia (2001)
6. Caseau, Y.: Efficient handling of multiple inheritance hierarchies. In: 8th Ann. Conf. on Object-Oriented Programming Systems, Languages, and Applications, pp. 271–287. ACM, New York (1993)
7. Dietz, P.F.: Maintaining order in a linked list. In: 14th Ann. ACM Symp. on Theory of Computing, pp. 122–127. ACM, New York (1982)
8. Gene Ontology, <http://www.geneontology.org>
9. Hsu, W.: PC-Trees vs. PQ-Trees. In: Wang, J. (ed.) COCOON 2001. LNCS, vol. 2108, pp. 207–217. Springer, Heidelberg (2001)
10. Krall, A., Vitek, J., Horspool, R.N.: Near optimal hierarchical encoding of types. In: Aksit, M., Matsuoka, S. (eds.) ECOOP 1997. LNCS, vol. 1241, pp. 128–145. Springer, Heidelberg (1997)
11. Li, Q., Moon, B.: Indexing and Querying XML Data for Regular Path Expressions. In: Apers, P.M., Atzeni, P., Ceri, S., Paraboschi, S., Ramamohanarao, K., Snodgrass, R.T. (eds.) 27th Int'l Conf. on VLDB, pp. 361–370 (2001)
12. Lentner, M., Subieta, K.: ODRA: A Next Generation Object-Oriented Environment for Rapid Database Application Development. In: Ioannidis, Y., Novikov, B., Rachev, B. (eds.) ADBIS 2007. LNCS, vol. 4690, pp. 130–140. Springer, Heidelberg (2007)
13. Matono, A., Amagasa, T., Yoshikawa, M., Uemura, S.: A path-based relational RDF database. In: H. E. Williams and G. Dobbie (eds.) 16th Australasian Database Conf., vol. 39; ACM Int'l Conf. Proc. Series, vol. 103, pp. 95–103. Australian Computer Society, Darlinghurst (2005)
14. Maier, D.: An Efficient Method for Storing Ancestor Information in Trees. SIAM Journal on Computing 8(4), 599–618 (1979)
15. OWL Web Ontology Language Overview, <http://www.w3.org/TR/owl-features>
16. Pizza Ontology v1.5, <http://www.co-ode.org/ontologies/pizza/2007/02/12>
17. Preuveneers, D., Berbers, Y.: Prime Numbers Considered Useful: Ontology Encoding for Efficient Subsumption Testing: Technical report CW464, Department of Computer Science, Katholieke Universiteit, Leuven (2006)
18. Sans, V., Laurent, D.: Prefix based numbering schemes for XML: techniques, applications and performances. In: Proc. VLDB, vol. 1(2), pp. 1564–1573 (2008)
19. Suggested Upper Merged Ontology, <http://www.ontologyportal.org>
20. Zibin, Y., Gil, J.Y.: Efficient subtyping tests with PQ-encoding. In: 16th ACM SIGPLAN Conf. on Object-Oriented Programming, Systems, Languages, and Applications, pp. 96–107 (2001)

Theme Extraction from Chinese Web Documents Based on Page Segmentation and Entropy

Deqing Wang, Hui Zhang, and Gang Zhou

State Key Lab of Software Development Environment, Beihang University,
100191 Beijing, P.R. China
{wangdeq, hzhang, gzhou}@nlsde.buaa.edu.cn

Abstract. Web pages often contain “clutters” (defined by us as unnecessary images, navigational menus and extraneous Ad links) around the body of an article that may distract users from the actual content. Therefore, how to extract useful and relevant themes from these web pages becomes a research focus. This paper proposes a new method for web theme extraction. The method firstly uses page segmentation technique to divide a web page into many unrelated blocks, and then calculates entropy of each block and that of the entire web page, then prunes redundant blocks whose entropies are larger than the threshold of the web page, lastly exports the rest blocks as theme of the web page. Moreover, it is verified by experiments that the new method takes better effect on theme extraction from Chinese web pages.

Keywords: Page segmentation, information entropy, theme extraction.

1 Introduction

Web pages may often contain “clutters” (defined by us as unnecessary images, navigational menus and extraneous links) around the body of an article that may distract users from the actual content. These “clutters” bring new challenges to theme extraction from web pages. How to extract informative theme from web pages has become one of the hotspots in information extraction area.

At the same time, a web page often contains more than one theme. How to divide a web page into many blocks using its semi-structural features is another issue needs to be addressed.

For the above analysis, the main technical difficulties of theme extraction from web pages are: firstly, how to use semi-structural features of web pages to achieve divisions of thematic blocks; secondly, how to discover and extract informative blocks.

This paper proposes a method named Theme Extraction from Chinese Web Based on Page Segmentation and Entropy. Our method, first uses page segmentation algorithm to divide a web page into blocks, then calculates entropy value of each block, and last partitions blocks into either informative or redundant by comparing to the

threshold of the entire web page. The redundant blocks are pruned and the remaining blocks will be as web theme. Experiments show that the new method works well on theme extraction from Chinese web pages.

2 Related Work

The themes of web pages can be located by their structures. So information extraction systems often parse web pages to be DOM trees and extract themes from DOM trees by automatic or semi-automatic rules [1-3].

Cardie [4] defines five pipelined processes for an information extraction system: tokenization and tagging, sentence analysis, extraction, merging, and template generation. SoftMealy [2] and Wrapper [3] are well known systems that extract the structural information from web pages based on manually generated templates or examples. In Wrapper induction [3], the author manually defines six wrapper classes, which consist of knowledge to extract data by recognizing delimiters to match one or more of the classes. The richer a wrapper class, the more probable it will work with any new site [5]. SoftMealy [2] provides a graphical user interface that allows users to open a Web site, define attributes and label tuples on the Web page. The common disadvantages of information extraction systems are the cost of templates, domain-dependent NLP knowledge, or annotations of corpora generated by hand. This is why these systems are merely applied to specific Web applications, which extract the structural information from pages of specific Web sites or pages. Crescenzi [6] perfects current templates through dealing with mismatches of pages, and ultimately derives templates covering training set. The templates can be used to extract web contents from similar web pages. Shian-Hua Lin [7] designed an information extraction system, named InfoDiscoverer. The system first partitions a web page into several content blocks according to HTML tag <TABLE> in the Web page. Based on the weights (calculated by $tf*idf$) of features in the set of web pages, it calculates entropy value of each feature. According to the entropy value of each feature in a content block, it can discover whether each block is informative or redundant. The entropy values of features in InfoDiscoverer system, considers all of the training set. But our new method only considers one web page. We will introduce some concepts before explaining our new method.

3 Concepts

3.1 Semantic Text Unit

Web pages are one source of web mining and each page contains one or several text blocks, which often are thematic blocks. Many navigational menus and Ad links are embedded in these blocks. This paper names each block as one Semantic Text Unit (STU), proposed by Buyukkokten [9]. That is to say, each block is entitled a "STU". Fig. 1 shows a Chinese sample page of STUs.



Fig. 1. A Chinese sample page of STUs

In Fig.1, each black rectangular box can be called a STU. Through analyzing the web page, we know that not each STU is relevant to the theme of the web page. Only the middle STU is the theme of the entire web page, and the rest 4 STUs are redundant.

3.2 Entropy

Information theory establishes a common measurable model for various types of information. Since information extraction treats “information” to be targets and objectives, we can use information theory to discuss the measure of “information”, which provides an information measurable model from statistical point.

1948, Shannon [10] introduced the concept of entropy, defined by Boltzmann, into information theory as a measure of uncertainty of random incident. Given a random event A , it has n possible independent outcomes: A_1, A_2, \dots, A_n , and the probability of each outcome is P_1, P_2, \dots, P_n , respectively. In order to measure the uncertainty of event A , Shannon introduced formula 1,

$$H_n(A) = -k \sum_{i=1}^n P_i \ln P_i \quad (1)$$

$H_n(A)$ is called Shannon entropy, stands for the uncertainty of event A . In formula 1, k is a constant and greater than zero. The entropy value $H_n(A)$ is greater than zero. The smaller the entropy value is, the more certain event A is.

4 A New Method

Through analyzing traditional information extraction methods of web pages, this paper introduces entropy into pruning algorithm of DOM trees, and proposes a new method named Chinese Web Theme Extraction Based on Page Segmentation and

Entropy. The new method both considers structural characteristics of web pages and considers semantic information of each node of DOM tree. The new method is based on the following assumptions:

- (1) Each web page has a convergent theme.
- (2) The smaller the entropy value of a STU is, the more convergent the STU node is and the more relevant the STU node is to the theme of the entire web page.

For the first assumption, a web page often presents one or several themes and the theme of them are convergent. It means the web page only presents limited themes, rather than an unlimited number of themes. So the first assumption is reasonable.

The second assumption is in terms of information entropy. According to formula 2,

$$I(A_i) = \log \frac{1}{p(A_i)} \tag{2}$$

$p(A_i)$ is the probability of the keyword A_i . If $p(A_i)$ is smaller, $I(A_i)$ is bigger. In a web page, if the frequency of a keyword is lower, the keyword is less related to themes. That is, the keyword is probable noisy. From the point of block, if probability of all keywords of a STU node is smaller, then the entropy value of the STU node is larger. That means the STU node has less relation with the entire web page. Contrariwise, a STU node, whose entropy value is smaller, has more relation with the entire web page.

The new method of theme extraction from web pages has the following processes:

- 1) Parse HTML Document into DOM tree, and then preprocess the tree, including deleting images, scripts and buttons by sample rules.
- 2) Divide the web page into several STUs, then segment text of each STU into Chinese words using ICTCLAS, a famous Chinese word segmentation tool, and count the frequency of each word.
- 3) Calculate the entropy value of each STU node, the entropy value of parent node can get by summing entropy values of its children nodes.
- 4) Prune the STU nodes which entropy values exceed threshold (the entropy value of the entire web page multiplies by constant K), and the remaining STU nodes are output as the theme of the web page. The system flow chart is shown in Fig.2.

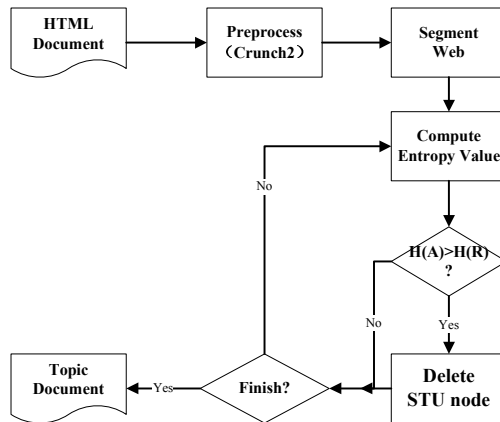


Fig. 2. Flow chart of the System

The merits of the new method are to take web pages' own structural features and semantic information of each STU node into consideration. And information entropy values provide a basis on pruning DOM tree.

In the rest of Section 4, the preprocessing of the web page is presented in Section 4.1. In Section 4.2, web segmentation based on tags is introduced. In Section 4.3, a new calculation formula of entropy is proposed. Last, some pruning rules of DOM tree are presented.

4.1 Preprocess

The preprocessing of web pages plays an important role in theme extraction from web pages. This paper implements preprocess by Crunch system, developed by Suhit Gupta [11] in Columbia University. Crunch is a versatile solution, allowing programmers and administrators to add heuristics to the framework. These heuristics act as filters that can be parameterized and toggled to perform the content extraction. Crunch reduces human involvement in the application of thresholds for the heuristics by automatically detecting and utilizing the content genre (context) of a given website [11].

The filter modules of Crunch system adopt step-by-step detecting strategy. Detection just uses one filter to delete corresponding node. After passing all of filters, the theme of a web page can be obtained. But some noisy nodes are still retained.

Filter modules are divided into two levels: junior filters and senior filters. The junior filters are sample, and they delete some nodes which contain tags such as , <BUTTON>, <INPUT> and <SCRIPT> and so on. Because the roles of these tags are used to show pictures or script language and these nodes containing such tags are unrelated to the theme, we can prune these nodes when traversing DOM tree. But hyperlink tag can't be deleted by sample rules, because web pages contain some Ad hyperlinks unrelated to theme and some navigation hyperlinks which maybe be related to theme. We must adopt senior filters which analyze hyperlinks and just delete some which are unrelated to theme. We use Equation 3 to decide whether a node is deleted.

$$rate = \frac{Number(link)}{Number(text)} \quad (3)$$

In Equ.3, $Number(link)$ stands for the number of hyperlinks of a STU node and $Number(text)$ stands for the number of non-linked words. If the rate of a STU node exceeds the threshold, then the STU node will be deleted. The threshold can be attained by experience. We gather 100 web pages from Sina News (<http://news.sina.com.cn>) and find that hyperlinks can be deleted efficiently if the threshold is between 0.2 and 0.3. In this paper, the threshold is 0.25. The web pages after being preprocessed still retain some redundant nodes, which can be removed by our new method. So we can get more accurate themes.

4.2 Page Segmentation

Page segmentation is used to divide web pages into STUs. And Chinese word segmentation can be done at the same time. A web page Ω is specified by a two-tuple

$\Omega=(O,\Phi)$, $O=(\Omega_1,\Omega_2,\dots,\Omega_n)$ is a finite set of objects or sub-web-pages. All these objects are not overlapped. Each object Ω_i can be recursively viewed as a sub-web-page and has a subsidiary content structure $\Omega_i=(O_i,\Phi_i)$. $\Phi=(\Phi_1,\Phi_2,\dots,\Phi_i)$ is a finite set of virtual separators, including horizontal separators and vertical separators. In fact, the separator is certain when two STUs are defined.

The tags of page segmentation determine the granularity of STUs. The granularity which is too rough will retain some redundant nodes. Contrarily, the granularity which is too smooth will result from incomplete of theme. Through experiences, we find <TABLE> and <DIV> are appropriate for Chinese news web pages.

The purpose of page segmentation is to transform a DOM tree into a semantic DOM tree. The semantic attributes of each STU contain the text and frequency of keywords. Through statistics of keywords of each STU node, we can get a vector of keywords $K=[C_i],(i=0,\dots,n)$, C_i is count of i-th keyword of the STU node.

4.3 Entropy Calculation

The entropy calculation of web pages must meet two conditions: 1) Semantic text of each STU node can be regarded as a random sequence, and we can use keywords vector to stand for the random sequence. 2) The bigger the frequency of a keyword in a STU block is, the greater contribution of the keyword has to the theme.

If the entire web page has m STUs, We can get vector V of all STUs, $V=(K_1,K_2,\dots,K_i,\dots,K_m)$ in which $K_i=[C_j],(j=0,\dots,n)$, C_j is count of j-th keyword in i-th STU node. We normalize vector K_i and obtain $K_i'=[\frac{C_j}{C}]$, in which C is sum of counts of all keywords in i-th STU node.

For a STU node A, $H(A)$ is the entropy value of A, C is sum of counts of all keywords in A, C_i is the count of i-th keyword in A and C_r is sum of counts of all keywords in the entire web page. Then,

$$H(A) = \sum_{C_i \in C} \frac{C_i}{C} \ln \frac{C_r}{C_i} = - \sum_{C_i \in C} \frac{C_i}{C} \ln \frac{C_i}{C_r} \tag{4}$$

From Equ. 4, we can see that when C_i becomes bigger, $\ln \frac{C_r}{C_i}$ becomes smaller

because C_r is constant for a web page.

For the entire web page W, $H(R)$ is the entropy value of W, C_r is the count of all of keywords of W, C_{ki} is i-th keyword in W. Then,

$$H(R) = \sum \frac{C_{ki}}{C_r} \ln \frac{C_r}{C_{ki}} \tag{5}$$

In Equ. 4 and Equ. 5, they are not the same as formula 1. We do some transformation considering the following factors. First, we change \log to \ln , that just change the unit. Second, we define $k = 1$. Third, we use $\frac{C_r}{C_i}$ instead of $\frac{C}{C_i}$ in Equ.4, just because the new method takes the entire web page as a random sequence. That is, the count of a keyword should be computed in whole web page instead of in STU node. Then, for keywords K1 and K2 in the same STU node, if the count of K1 is greater than that of K2, K1 is more related to theme than K2. And in Equ.4, $\ln \frac{C_r}{C_{K1}} < \ln \frac{C_r}{C_{K2}}$, it meets our assumptions.

4.4 Pruning Rules

The entropy value is used to measure the relevance between a STU node of the web page and the theme of the web page. If we take a web page just containing theme as a convergence system and when the entropy value of the web page is less than a certain value, the remaining text of the web page is the theme. The basic principle is pruning the STU node which makes a greater contribution to the entropy value of entire web page. So the pruning algorithm is as follows:

- 1) Input: Semantic DOM tree, the parameters K;
- 2) According to Equ.5, calculate the entropy value ($H(R)$) of semantic DOM tree;
- 3) Traverse the DOM tree according to the depth-first algorithm, then calculate the entropy value ($H(A)$) of each STU node by Equ.4, and then compare it to $K \times H(R)$ (K is constant; by adjusting the value of K will be able to control the pruning). If $H(A) > K \times H(R)$, prune the node and its children nodes. Otherwise, the node is retained.
- 4) If the traverse of semantic DOM tree is finished, the pruning work is over. Output the remaining STU nodes as the theme.

5 Experiments

5.1 Evaluations

To evaluate the performance of IE system, there are three evaluation criteria: precision rate, recall rate and F-Measure, which are from the Message Understanding Conference (MUC) [12].

The following equations define recall, precision, and F-Measure in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Precision (P) and Recall (R) are sensitive to how well the system performs on the true positives, and ignore true negatives altogether.

$$P = \frac{TP}{TP + FP} \quad (6)$$

$$R = \frac{TP}{TP + FN} \tag{7}$$

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{8}$$

In information extraction system, $\beta = 1$, which means that P and R have the same importance.

5.2 Parameter Estimation

Due to the diversity of web pages, the extraction results are also diverse by our new method. The parameter K will directly affect the final results of theme extraction. So we use precision, recall and F-Measure to decide the value of parameter K.

Through Section4.4 we know the greater K is, the greater $K \times H(R)$ is. So some redundant STU nodes can be retained, which will lead to low precision. In our experience, we randomly collect 50 web pages from News of Sina. We increase K by step 0.05 from 1.0 to 1.4, and we obtain the curves between K and the three evaluation criteria as shown in Fig.3.

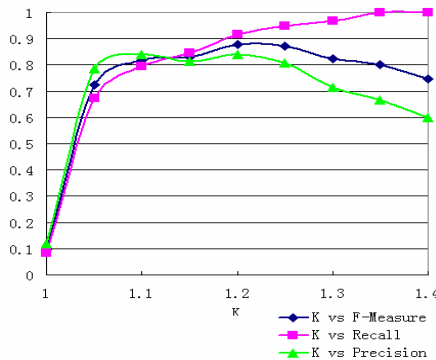


Fig. 3. The curves between K and F-Measure, Precision, Recall

From Fig.3, we can see that the Precision increases up to a vertex and then decreases as shown in the triangle curve. That is because informative blocks increase faster when the value of K increases from 1.0 to 1.20 by step 0.05. And when the value of K is greater than 1.2, redundant blocks increase faster, so the Precision begins decreasing. The Recall rate varies directly with the value of K because when the value of K increases, pruned STU nodes will decrease. That is to say, all of the informative blocks will be retained. Especially in our experience, when the value of K is greater than 1.35, the Recall rate is 100%. The relation between F-Measure and K is similar with that between Precision rate and K. In our IE system, Precision rate and Recall rate have the same importance, so the value of K is the best value when F-Measure gets maximum value. In Figure 3, when K = 1.20, the F-Measure of News of Sina gets maximum value and F-Measure = 0.878.

5.3 Results

The new method is implemented with Java in PC with P4 2.8GHz CPU and 1G memory. We randomly collect 8 datasets from 8 famous news web sites in China, one of which has 50 web pages. Then we manually label a set of tags for identifying STU blocks of web pages. We evaluate our new method through manually analyzing whether the extracted blocks is informative blocks.

Through statistics of true informative blocks, false informative blocks and informative blocks needed to extract, we get the result as shown in Table 1.

As is shown in Table 1, the new method does well in web theme extraction, especially in qq (a famous news web site), the precision rate of qq is up to 88.3% and the recall rate of qq is up to 94.4%. That meets the requirement of web theme extraction. At the same time, the new method has universal applicability to Chinese news web sites.

Table 1. Precision, Recall and F-Measure of 8 datasets

Datasets	number	K	TP+FP	TP	TP+FN	Precision	Recall	F-Measure
sina	50	1.19	107	90	98	0.841	0.918	0.878
163	50	1.04	118	99	105	0.839	0.943	0.888
qq	50	1.03	77	68	72	0.883	0.944	0.913
xinhua	50	1.23	105	91	100	0.867	0.910	0.888
renmin	50	1.2	73	60	67	0.822	0.896	0.857
china	50	1.06	61	53	58	0.869	0.914	0.891
cyol	50	1.06	88	76	82	0.864	0.927	0.894
sohu	50	1.04	82	69	76	0.841	0.908	0.873

6 Conclusion

The paper proposes a new method for extracting web theme using structural characters of web pages and information entropy. The new method is efficient and easy to achieve, especially for web pages whose themes are clear. For web pages having more themes and complex structures, semantic concepts should be applied to web theme extraction. But now semantic analysis and semantic understanding technologies are not perfect to web theme extraction, especially Chinese has much ambiguity and synonyms, which make the understanding of Chinese difficult. So we will apply semantic analysis of Chinese into web theme extraction in our future work and we expect to get better results.

Acknowledgments

This research is supported in part by National Science & Technology Infrastructure Foundation of China (NO. 2005DKA63901). We thank Xiujuan Jiang for checking spelling and grammatical error.

References

1. Freitag, D.: Machine Learning for Information Extraction. PhD Dissertation, Computer Science Dept., Carnegie Mellon University Pittsburgh (1998)
2. Hsu, C.N., Dung, M.T.: Generating Finite-State Transducers for Semi-Structured Data Extraction from the Web. *J. Information Systems* 23(8), 521–538 (1998)
3. Kushmerick, N., Weld, D., Doorenbos, R.: Wrapper Induction for Information Extraction. In: Proc. 15th Int'l Joint Conf. Artificial Intelligence, IJCAI (1997)
4. Cardie, C.: Empirical Methods in Information Extraction. *AI Magazine* 18(4), 5–79 (1997)
5. Chidlovskii, B.: Wrapper Generation by k-Reversible Grammar Induction. In: Workshop on Machine Learning for Information Extraction (August 2000)
6. Crescenzi, V., Mecca, G., et al.: Roadrunner: Towards automatic data extraction from large web sites. In: Proceedings of the 27th International Conference on Very Large Database, Roma, Italy (2001)
7. Lin, S.H., Ho, J.M.: Discovering Informative Content Blocks from Web Documents. In: Proc. Eighth ACM SIGKDD (2002)
8. Sahuguet, A., Azavant, F.: Building intelligent Web applications using lightweight wrappers. In: Data & Knowledge Engineering (2001)
9. Buyukkokten, O., et al.: Accordion Summarization for End-Game Browsing on PDAs and Cellular Phones. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems (2001)
10. Shannon, C.: A Mathematical Theory of Communication. *Bell System Technical Journal* 27, 379–423, 623–656 (1948)
11. Gupta, S., Kaiser, G., Neistadt, D.: DOM-based content extraction of HTML documents. In: Proceedings of the 12th international conference on World Wide Web, pp. 207–214. ACM Press, New York (2003)
12. Message Understanding Conferences,
http://www-nlpir.nist.gov/related_projects/muc

Topic-Based Hard Clustering of Documents Using Generative Models

Giovanni Ponti and Andrea Tagarelli

Dept. of Electronics, Computer and Systems Sciences, University of Calabria, Italy
{gponti, tagarelli}@deis.unical.it

Abstract. In this paper, we describe a framework for clustering documents according to their mixtures of topics. The proposed framework combines the expressiveness of generative models for document representation with a properly chosen information-theoretic distance measure to group the documents via an agglomerative hierarchical clustering scheme. The clustering solution obtained at each level of the dendrogram reflects an organization of the documents into sets of topics, while being produced without the effort needed for a soft/fuzzy clustering method. Experimental results obtained on large, real-world collections of documents evidence the effectiveness of our approach in detecting non-overlapping clusters that contain documents sharing similar mixtures of topics.

1 Introduction

Document clustering is a major topic of research in data management and retrieval due to an ever increasing availability of textual sources in several application domains (e.g., news articles, scientific papers, legal documents, web pages). Document clustering methods traditionally fall into the category of *discriminative* (distance-based) approaches, as they require the computation of distance/similarity between documents to group them into clusters. Moreover, such methods usually exploit classic retrieval models which represent the document contents in a term (word) feature space, such as the popular “bag-of-words” model.

Most of the existing applications of document clustering have traditionally assumed to assign each document to a unique cluster. In this way, any cluster refers to a group of documents that discuss the same main topic (i.e., a concept which is somehow described by the most representative terms of the documents within the cluster). However, this assumption is not effective for a broad variety of text data, such as interdisciplinary documents. To address this issue, there have been various proposals which assign documents to more than one cluster, usually performing a soft/fuzzy scheme [21, 12].

In recent years, a number of approaches to modeling document contents have been developed based on the idea that any document can be represented as a *mixture* of probability distributions over its terms, and each component of the mixture refers to a *topic* [17, 3, 22, 23]. The document representation is hence obtained by a *generative process*, which expresses the document features as mixture models. In this way, generative model-based methods are able to provide a finer-grained modeling of documents, which is particularly useful to fit the case of interdisciplinary or multi-topic documents.

For document clustering purposes, there can be found various ways of inducing an organization of the documents according to the topic-space based representation offered by generative models. For instance, each document can be assigned to the most probable topic, or to a set of topics according to a given probability threshold. Both these strategies however incur evident drawbacks, as the first one is not able to support multi-topic assignment, whereas the second one may lead to overlay overlapping clusters.

In this work, we develop a method for topic-based hard clustering of documents. Documents are assumed to be modeled by a generative process, which provides a mixture of probability mass functions (pmfs) to model the topics that are discussed within any specific document. The proposed clustering method refers to a centroid-linkage-based agglomerative hierarchical scheme [15]. Moreover, it is “hard” in that it produces a clustering solution for a given collection of documents in such a way that each document is assigned to a unique cluster and each cluster contains documents that share the same topic assignment. To compare the documents, we employ an information-theoretic measure which is defined to compute the distance between any two probability distributions over a topic feature space.

We experimentally evaluated the effectiveness of the proposed approach over large collections of documents. A major goal of this evaluation was to assess the advantages of combining the expressiveness of state-of-the-art generative models for document representation with information-theoretic distance; in particular, the combination of LDA model with Bhattacharyya distance revealed to be a compelling solution for topic-based hard clustering of documents.

2 Related Work

We discuss here related work focusing on generative models for document representation. Due to the space limits of this paper, we leave out of consideration the vast literature on document clustering algorithms and applications.

Identifying a topic feature space in a given document collection is traditionally accomplished by mapping the term-document representation to a lower-dimensional latent “semantic” space [6]. Following this line, one of the earliest proposals for topic-based document representation via a generative model is *Probabilistic Latent Semantic Analysis* (PLSA) [17]. As a probabilistic version of LSA [6], originally introduced to better handling problems of term polysemy in document retrieval applications (e.g., [7, 18, 20]), PLSA defines a statistical latent topic model in which the conditional probability between documents and terms is modeled as a latent variable. In this way, it is possible to assign an unobserved class variable to each observation (i.e., the occurrence of a word in a given document), since each document is composed by a mixture of distributions. Each term may belong to one or more classes and a document may discuss more than one topic.

PLSA assumes a bag-of-words document representation, in which the word order is disrupted. The so-called assumption of *exchangeability* for the words within a document can also be extended to the documents in a collection. The *Latent Dirichlet Allocation* (LDA) [3] is able to consider mixture models that express the exchangeability of both words and documents. In LDA, the generative process of a document collection consists

of a three-level scheme that involves the whole corpus, the documents, and the words in each document. More precisely, for each document, a distribution over topics is sampled from a Dirichlet distribution; for each word in a document, a single topic is selected according to this distribution, and each word is sampled from a multinomial distribution over words specific to the sampled topic. In this way, LDA defines a more sophisticated generative model for a document collection, whereas PLSA generates a model for each document separately from the other ones in the collection.

Generative models for document representation like PLSA and LDA put the basis for a relatively recent corpus of study on model-based document classification and clustering. In the context of supervised classification, the *Parametric Mixture Model* (PMM) [19] assumes that a multi-labeled text contains words that support for a mixture of categories. A basis vector is used for each document to express which category the document belongs to, and the model parameters of each single topic are mixed with an equal ratio. Recently, *Parametric Dirichlet Mixture Model* (PDMM) [16] has been proposed as an improved PMM, which aims to address the problem of considering each topic with equal mixture ratio by using a Dirichlet distribution.

Ext-PLSA [9] has very recently been proposed as an extension of PLSA for document clustering tasks. The main idea is to overcome some limits deriving from a straightforward use of PLSA to derive clustering solutions, such as the one-to-one correspondence between clusters and topics. To meet the requirement that the number of desired clusters does not necessarily match the number of topics, Ext-PLSA introduces a new latent variable that allows words and documents to be clustered simultaneously.

3 The Proposed Framework

In this section, we present our framework for topic-based hard clustering of documents. Let $\mathcal{D} = \{d_1, \dots, d_N\}$ be a collection of documents and $\mathcal{W} = \{w_1, \dots, w_M\}$ be a set of terms occurring in the documents in \mathcal{D} , which corresponds to the vocabulary of \mathcal{D} . Moreover, let $\mathcal{T} = \{\tau_1, \dots, \tau_T\}$ denote the set of topics underlying the documents in \mathcal{D} . Using a generative model for document representation, \mathcal{T} is typically associated to a latent variable, since \mathcal{T} is unknown. The topic distribution of any given document is represented as a probability mass function (pmf), since the variable associated to the topic distribution \mathcal{T} can be seen as a discrete random variable, such that $\sum_{t=1}^T \Pr(\tau_t | d_i) = 1, \forall i \in [1..N]$. We denote with \hat{d}_i the document d_i modeled as a probability distribution according to a given generative process and with $\hat{\mathcal{D}}$ the probabilistic representation of the document collection.

Figure 1 depicts the conceptual structure of the framework, which consists of three main steps. First, the input corpus \mathcal{D} of raw texts is structured by using standard pre-processing techniques for text data (e.g., lexical analysis, removal of stopwords, stemming); this step yields a word-occurrence matrix, where each element x_{ij} represents the number of occurrences of the word w_j in the document d_i .

The second step consists in applying a generative model to represent the documents into a feature space over the topics in \mathcal{T} . This step produces a probability matrix which expresses the mixture of topics underlying the input documents; precisely, the value $\Pr(\tau_t | d_i)$ is computed to measure the probability that the topic τ_t is associated to the document $d_i, \forall i \in [1..N]$ and $\forall t \in [1..T]$.

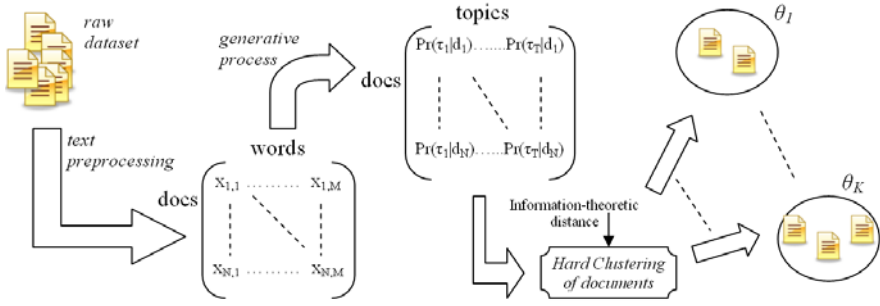


Fig. 1. Conceptual architecture of topic-based hard clustering of documents

The final step is to cluster the documents such that each of the resulting clusters is characterized by a mixture of topics underlying the documents within that cluster. Given a clustering solution $\mathcal{C} = \{C_1, \dots, C_K\}$ for $\hat{\mathcal{D}}$, there exists a set of topic-sets $\Theta = \{\theta_1, \dots, \theta_K\}$, where $\theta_k \subset \mathcal{T}, \forall k \in [1..K]$ and $\bigcup_{k=1}^K \theta_k = \mathcal{T}$, such that each $C_k \in \mathcal{C}$ is described by one topic-set $\theta_k \in \Theta$.

It should be noted that while the assignment of documents to clusters is performed in a disjoint way, clusters correspond to possibly overlapping topic-sets. In a sense, this strategy introduces softness in the clustering solution (to a topic-level), although it is conceptually simpler than, e.g., fuzzy clustering because it does not allow overlaps of cluster in terms of their members (documents).

3.1 Distance for Model-Based Documents

The clustering algorithm in our framework is equipped with a distance measure that is able to compute the proximity between documents according to their mixtures of topics. In this respect, *information theory* represents a fruitful area of research. Two of the most frequently used measures for probability distributions are the Kullback-Leibler divergence [11, 10] and the Chernoff distance [4], which fall into the Ali-Silvey class of information-theoretic distance measures [1]. However, the use of such distances for document retrieval might not be a good idea due to a number of drawbacks. For instance, Kullback-Leibler divergence is not symmetric, while the Chernoff distance has a high computational complexity; in addition, both these functions do not satisfy the triangle inequality.

A well-suited distance measure exploits instead the notion of *Bhattacharyya coefficient* [2, 8]. Given a discrete random variable X over a set of values $S = \{s_1, \dots, s_L\}$, $s_l \in \mathbb{R}, \forall l \in [1..L]$, and any two probability distributions $p_X(x)$ and $q_X(x)$, the Bhattacharyya coefficient is defined as:

$$\rho(p_X(x), q_X(x)) = \sum_{x \in S} \sqrt{p_X(x) q_X(x)} \tag{1}$$

The Bhattacharyya coefficient has an important geometric interpretation. In fact, it represents the cosine between the two vectors for p and q , which are composed by the square root of the probabilities of the mixtures that shape p and q [2]. This property is

Algorithm 1. MuToT-AHC

Input: a corpus of model-based text data $\hat{D} = \{\hat{d}_1, \dots, \hat{d}_N\}$,
 (optionally) a desired number $K = |\Theta|$ of topic-sets

Output: a set of partitions \mathbf{C}

- 1: $\mathcal{C} \leftarrow \{C_1, \dots, C_N\}$ such that $C_i = \{\hat{d}_i\}, \forall i \in [1..N]$
- 2: $\mathcal{P}_{C_i} \leftarrow \hat{d}_i, \forall i \in [1..N]$, as initial cluster prototypes
- 3: $\mathbf{C} \leftarrow \{\mathcal{C}\}$
- 4: **repeat**
- 5: let C_i, C_j be the pair of clusters in \mathcal{C} such that
 $\frac{1}{2}(\text{dist}(\mathcal{P}_{C_i \cup C_j}, \mathcal{P}_{C_i}) + \text{dist}(\mathcal{P}_{C_i \cup C_j}, \mathcal{P}_{C_j}))$ is minimum
- 6: $\hat{\mathcal{C}} \leftarrow \{\mathcal{C} \mid C \in \mathcal{C}, C \neq C_i, C \neq C_j\} \cup \{C_i \cup C_j\}$
- 7: $\mathbf{C} \leftarrow \mathbf{C} \cup \{\hat{\mathcal{C}}\}$
- 8: update prototypes $\mathcal{P}_C, C \in \mathcal{C}$
- 9: **until** $|\mathcal{C}| = 1$ or $|\mathcal{C}| = K$
- 10: **return** \mathbf{C}

important in our setting, since each model-based document is expressed as a pmf which is a mixture of topics.

Among the various distance measures which can be defined based on the Bhattacharyya coefficient (e.g., [8]), in this work we resort to the following definition:

$$B(p_X(x), q_X(x)) = \sqrt{1 - \rho(p_X(x), q_X(x))} \quad (2)$$

Equation 2 has many advantages with respect to other Bhattacharyya distances; for instance, it satisfies the triangle inequality and is easier to compute in practice than its more general case (i.e., the Chernoff distance).

3.2 The Clustering Algorithm

The proposed clustering method is a centroid-based-linkage agglomerative hierarchical algorithm (AHC), called *Multi-Topic Text data* (MuToT-AHC). The main features of MuToT-AHC, which is shown in Algorithm 1, concern (i) the computation of cluster prototypes (i.e., cluster centroids), and (ii) the cluster merging criterion. A cluster centroid is represented as a mixture that summarizes the pmfs of the documents within that cluster. The cluster merging criterion, which decides the pair of clusters to be merged at each step, uses a proper information-theoretic distance to compare the cluster centroids—as we previously discussed, in this work we employ the Bhattacharyya distance (Equation 2) as default.

Given a corpus \hat{D} of text data modeled via a generative algorithm, MuToT-AHC follows the classic AHC scheme to yield a hierarchy \mathbf{C} of clustering solutions. Optionally, the algorithm requires a number of clusters which corresponds to a given number of topic-sets ($K = |\Theta|$). At each iteration of the algorithm, the prototype of each cluster is represented as the mean of the pmfs of the documents within that cluster. The merging score criterion (Line 5) applies to each pair of clusters C_i and C_j , and computes the average distance between the prototype of each of such clusters (\mathcal{P}_{C_i} and \mathcal{P}_{C_j}) and the prototype of the union cluster ($\mathcal{P}_{C_i \cup C_j}$). The pair of clusters which minimizes such a distance computation is then chosen as the pair of clusters to be merged. Intuitively, this

criterion aims to measure the lowest error merging as the one which is closest to both the original clusters. The algorithm stops when the number of clusters is equal to one, or the desired number of clusters is reached.

4 Experimental Evaluation

4.1 Datasets

We used four multi-topic datasets, called RCV1, PubMed, CaseLaw, and 20Newsgroups, whose main characteristics are shown in the first three columns of Table 1. RCV1 is a subset of the Reuters Corpus Volume 1 [14], which contains news headlines discussing topics related to, e.g., markets, politics, wars, crimes. PubMed is a collection of full free texts of biomedical articles available from the PubMed website. Fifteen topics were selected from the Medline’s Medical Subject Headings (MeSH) taxonomy ensuring that no ancestor-descendant relationship holds for any pair of the selected topics (e.g., viruses, medical informatics, mass spectrometry, genetics, etc.). CaseLaw is a corpus of tagged case law documents, which is comprised of very long texts discussing topics such as, e.g., sex, leases and rents, immigration, employment, divorces, etc. Finally, 20Newsgroups is a subset of an original collection of approximately 20,000 newsgroup documents, partitioned over 20 different newsgroups, including politics, religion, culture, and sciences.

Table 1. Datasets used in the experiments

<i>dataset</i>	<i>size</i> (# of docs)	<i># of words</i>	<i># of topic</i> <i>labels</i>	<i># of topic-sets</i> ($ \Theta $)	<i># of docs</i>
RCV1	6,588	37,688	23	49	5,896
PubMed	3,687	85,771	15	50	2,627
CaseLaw	2,550	50,567	20	1,697	2,550
20Newsgroups	2,544	22,386	25	23	2,234

Generating reference partitions. To assess the proposed hard clustering framework, we chose to compare the obtained clustering solutions with a reference partition of the documents for each dataset used in the evaluation. Since topic-labels are available for any specific dataset (cf. Table 1), a reference partition was built up by (i) identifying a number of topic-sets, each of which covers a significant portion of the document collection, and finally (ii) assigning each document to one of the classes (topic-sets) identified.

Figure 2 shows an example in which five topics are distributed over six documents, where any cross denotes the assignment of a document to a topic-label. Three topic-sets can be identified in this example, which correspond to a partitioning of the document collection in three classes. Note that each topic can be included in more classes.

The last two columns of Table 1 show the number of topic-sets identified ($|\Theta|$) and the corresponding number of documents involved for each dataset. We experimentally

¹ <http://www.ncbi.nlm.nih.gov/sites/entrez/>

² <http://caselaw.lawlink.nsw.gov.au/>

³ <http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html>

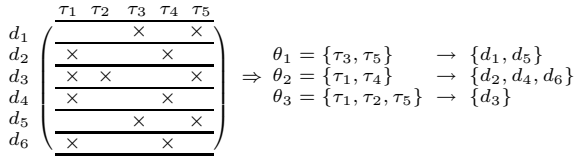


Fig. 2. Example of identification of topic-sets

determined the minimum number of documents to belong to any given topic-set for each dataset, by tuning this parameter and performing different runs of the clustering algorithm. In nearly all cases this minimum number of documents was selected as equal to 20, which revealed to be a reasonable choice to obtain a good trade-off between accuracy results and cluster size. An exception is represented by **CaseLaw**, which has a high number of topic-sets; as a consequence, there are few documents in this dataset that share the same multi-topic labeling, i.e., many topic-sets individually include less than 20 documents.

4.2 Assessing Clustering Solutions

We resort to a well-known external validity criterion called *F-measure* [13], which is defined as the harmonic mean of two standard notions in Information Retrieval, namely precision and recall.

Given a collection \mathcal{D} of objects, let $\Gamma = \{\Gamma_1, \dots, \Gamma_H\}$ denote the reference classification of the objects in \mathcal{D} , and $\mathcal{C} = \{C_1, \dots, C_K\}$ be the output partition yielded by a clustering algorithm. *Precision* of cluster C_j with respect to class Γ_i is the fraction of the objects in C_j that has been correctly classified, whereas *recall* of cluster C_j with respect to class Γ_i is the fraction of the objects in Γ_i that has been correctly classified. Formally, $P_{ij} = |C_j \cap \Gamma_i|/|C_j|$ and $R_{ij} = |C_j \cap \Gamma_i|/|\Gamma_i|$; from these values, the F-measure value is computed as $F_{ij} = 2P_{ij}R_{ij}/(P_{ij} + R_{ij})$. In order to score the quality of \mathcal{C} with respect to Γ by means of a single value, the overall F-measure $F(\mathcal{C}, \Gamma)$ is computed as the weighted sum of the maximum F_{ij} over $j \in [1..K]$ for each class Γ_i :

$$F(\mathcal{C}, \Gamma) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^H |\Gamma_i| \max_{j \in [1..K]} F_{ij}$$

4.3 Results

In this section, we present the main results obtained by our clustering algorithm on the various datasets. We also discuss how the accuracy trends vary as the clustering size decreases.

As far as the setup of the framework, we varied the generative model used for document representation, involving LDA, PLSA, and Ext-PLSA. Each of such models requires the setting of the latent variable used in the generative process; since this variable is related to the document topics, it is reasonable to set the number of its possible values to the number of topics for each dataset. Ext-PLSA has a further latent variable related to the size of the desired clustering. Moreover, MuToT-AHC was equipped with either symmetrized Kullback-Leibler (*KL*) or Bhattacharyya (*B*) distance.

Table 2. Summary of accuracy results: (a) without term selection, (b) filtering out terms with $DF < 3\%$, and (c) with $DF < 5\%$

(a)				(b)				(c)			
<i>dataset</i>	<i>model</i>	F_B	F_{KL}	<i>dataset</i>	<i>model</i>	F_B	F_{KL}	<i>dataset</i>	<i>model</i>	F_B	F_{KL}
RCV1	LDA	.66	.58	RCV1	LDA	.71	.61	RCV1	LDA	.69	.60
	PLSA	.57	.49		PLSA	.62	.51		PLSA	.60	.50
	Ext-PLSA	.60	.50		Ext-PLSA	.64	.53		Ext-PLSA	.63	.51
PubMed	LDA	.54	.42	PubMed	LDA	.59	.45	PubMed	LDA	.58	.43
	PLSA	.49	.36		PLSA	.51	.40		PLSA	.49	.38
	Ext-PLSA	.50	.38		Ext-PLSA	.52	.42		Ext-PLSA	.51	.39
CaseLaw	LDA	.79	.62	CaseLaw	LDA	.81	.65	CaseLaw	LDA	.79	.64
	PLSA	.77	.60		PLSA	.78	.63		PLSA	.77	.62
	Ext-PLSA	.77	.61		Ext-PLSA	.79	.63		Ext-PLSA	.78	.62
20Newsgroups	LDA	.61	.48	20Newsgroups	LDA	.68	.52	20Newsgroups	LDA	.65	.51
	PLSA	.52	.42		PLSA	.59	.45		PLSA	.52	.43
	Ext-PLSA	.54	.44		Ext-PLSA	.61	.47		Ext-PLSA	.53	.44

Accuracy. We tested MuToT-AHC on the evaluation datasets which were preprocessed according to three different settings, each having a different impact on the vocabulary of the collection: the first setting corresponds to using the entire vocabulary, whereas the second and third settings correspond to filtering out terms with document frequency (DF) lower than 3% and 5%, respectively.

A number of observations can be made analyzing Table 2, which summarizes the F-measure scores obtained by our algorithm by varying the distance measure (i.e., F_B and F_{KL}) and the document representation model. First, our clustering algorithm obtains significant improvements when equipped with the Bhattacharyya distance rather than with the Kullback-Leibler distance, regardless of the generative model and term selection used. This result supports what we mentioned in Section 2, that is, Bhattacharyya is more effective than Kullback-Leibler in estimating the best pair of clusters to be merged at each step; indeed, the computation of the symmetrized Kullback-Leibler leads to infinity when a topic is associated to a zero probability for only one of the two (document) pmfs [5].

Second, LDA always performs better than the other generative models, and the improvement is significant on most datasets—for instance, in relation to the setting $DF < 3\%$, the improvement is up to 9% on RCV1 and 20Newsgroups, 8% on PubMed. An exception is represented by CaseLaw, for which the accuracy gap between the generative models is less evident. This mainly depends on the fact that few documents share the same topic-set in CaseLaw, since there are many distinct topic-sets in that dataset. As a consequence, the benefit deriving from considering the whole document collection in the generative process by LDA is lower than other cases. Another remark concerns Ext-PLSA, which behaves as good as or slightly better than PLSA.

A final remark can be made on the impact of term selection on the clustering performance. Filtering out terms that are “rare” to a certain degree may lead to higher clustering quality. In particular, the best results are obtained by setting $DF < 3\%$, whereas the setting $DF < 5\%$ produce results that are quite close to those obtained when no term selection is carried out.

Scalability. We investigated the impact of different clustering sizes on the clustering performance. For this purpose, we varied the size of the clustering solutions for each

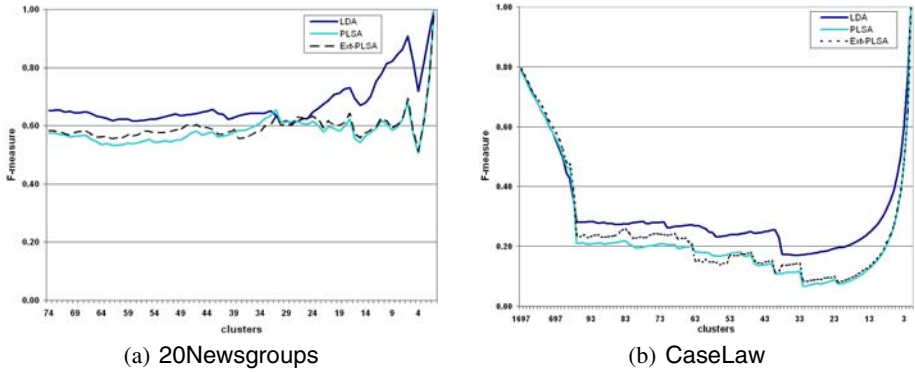


Fig. 3. Clustering performance by varying the clustering size

dataset from the original number of topic-sets (i.e., $|\Theta|$) to a minimum of 2 clusters; this corresponds to evaluating each level of the dendrogram produced by the MuToT-AHC algorithm according to a specific reference partition which was derived from the original set of topic-sets. Such reference partitions were built by merging, at each iteration of the algorithm, the two topic-sets sharing the maximal number of topics and having similar size. We accomplished this by computing the Jaccard distance on each pair of topic-sets θ_i and θ_j , which is formally expressed as $J(\theta_i, \theta_j) = 1 - |\theta_i \cap \theta_j| / |\theta_i \cup \theta_j|$. In this way, the pair θ_i and θ_j that maximizes the distance $J(\theta_i, \theta_j)$ is selected for merging.

Figure 3 shows the F-measure trends of MuToT-AHC equipped with Bhattacharyya, as the clustering size decreases. For the sake of brevity of presentation, we report the graphs for two datasets only, namely **20News groups** and **CaseLaw**. As we can see in the figure, the relative difference of performance between the three generative models by varying the clustering size confirms the observations previously discussed. This also holds for the remaining datasets. However, **CaseLaw** represents a distinguished case, in which the reduction of clustering size impacts on the quality significantly. In particular, the clustering quality drastically decreases during the earlier iterations of the algorithm, due to a very high number of topic-sets for this dataset (cf. Table 1).

5 Conclusion

We presented a framework for hierarchically clustering generative model-based documents using an information-theoretic distance measure. Documents are represented as probability distributions (mixtures of topics) and grouped together into disjoint clusters, each containing documents sharing the same topic-set. The information-theoretic distance is properly chosen in order to fit some natural requirements in a document clustering task. Experimental results have shown the effectiveness of our framework in clustering documents according to their mixtures of topics, and have highlighted the advantages offered by employing state-of-the-art document generative models and their combination with the Bhattacharyya distance.

Since the proposed approach intuitively contrasts with methods for soft clustering of documents on a term space, we are currently investigating such a comparison.

References

1. Ali, S.M., Silvey, S.D.: A General Class of Coefficients of Divergence of One Distribution from Another. *J. Royal Statistical Soc.* 28(1), 131–142 (1966)
2. Bhattacharyya, A.: On a Measure of Divergence Between Two Statistical Populations Defined by their Probability Distributions. *Bull. Calcutta Mathematical Soc.* 35, 99–110 (1943)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. *J. Machine Learning Research* 3, 993–1022 (2003)
4. Chernoff, H.: A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the sum of Observations. *Annals of Mathematical Statistics* 23(4), 493–507 (1952)
5. Cover, T.M., Thomas, J.A.: *Elements of Information Theory*, 2nd edn. Wiley-Interscience, Hoboken (2006)
6. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. *J. American Soc. for Information Science* 41, 391–407 (1990)
7. Bellegarda, J.R.: Exploiting both local and global constraints for multi-span statistical language modeling. *Acoustics, Speech and Signal Processing* 2, 677–680 (1998)
8. Kailath, T.: The Divergence and Bhattacharyya Distance Measures in Signal Selection. *IEEE Trans. on Comm. Tech.* 15(1), 52–60 (1967)
9. Kim, Y.-M., Pessiot, J.-F., Amini, M.-R., Gallinari, P.: An extension of PLSA for document clustering. In: *Proc. of ACM CIKM*, pp. 1345–1346 (2008)
10. Kullback, S.: *Information Theory and Statistics*. Wiley, Chichester (1959)
11. Kullback, S., Leibler, R.A.: On Information and Sufficiency. *Ann. Math. Stat.* 22(1), 79–86 (1951)
12. Kumnamuru, K., Dhawale, A., Krishnapuram, R.: Fuzzy co-clustering of documents and keywords. In: *Proc. of IEEE Int. Conf. on Fuzzy Systems*, vol. 2, pp. 772–777 (2003)
13. Larsen, B., Aone, C.: Fast and effective text mining using linear-time document clustering. In: *Proc. of ACM KDD*, pp. 16–22 (1999)
14. Lewis, D.D., Yang, Y., Rose, T.G., Dietterich, G., Li, F.: RCV1: A new Benchmark Collection for Text Categorization Research. *J. Machine Learning Research* 5, 361–397 (2004)
15. Murtagh, F.: A Survey of Recent Advances in Hierarchical Clustering Algorithm. *The Computer Journal* 26(4), 354–359 (1983)
16. Sato, I., Nakagawa, H.: Knowledge discovery of multiple-topic document using parametric mixture model with dirichlet prior. In: *Proc. of ACM KDD*, pp. 590–598. ACM, New York (2007)
17. Hofmann, T.: Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning* 42(1-2), 177–196 (2001)
18. Landauer, T.K., Dumais, S.T.: A Solution to Plato’s Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological Review* 104(2), 211–240 (1997)
19. Ueda, N., Saito, K.: Parametric Mixture Models for Multi-Labeled Text. In: *Proc. of Neural Information Processing Systems*, pp. 721–728 (2002)
20. Wolfe, M.B.W., Schreiner, M.E., Rehder, B., Laham, D., Foltz, P.W., Kintsch, W., Landauer, T.K.: Learning from text: Matching readers and texts by latent semantic analysis. *Discourse Processes* 25(2/3), 309–336 (1998)
21. Zhao, Y., Karypis, G.: Soft clustering criterion functions for partitioned document clustering: a summary of results. In: *Proc. of ACM CIKM*, pp. 246–247 (2004)
22. Zhong, S., Ghosh, J.: A unified framework for model-based clustering. *J. Machine Learning Research* 4, 1001–1037 (2003)
23. Zhong, S., Ghosh, J.: Generative model-based document clustering: a comparative study. *Knowl. Inf. Syst.* 8(3), 374–384 (2005)

Boosting a Semantic Search Engine by Named Entities

Annalina Caputo, Pierpaolo Basile, and Giovanni Semeraro

Department of Computer Science,
University of Bari,
70125 Bari, Italy
{acaputo,basilepp,semeraro}@di.uniba.it

Abstract. Traditional Information Retrieval (IR) systems are based on bag-of-words representation. This approach retrieves relevant documents by lexical matching between query and document terms. Due to synonymy and polysemy, lexical methods produce imprecise or incomplete results. In this paper we present SENSE (SEmantic N-levels Search Engine), an IR system that tries to overcome the limitations of the ranked keyword approach, by introducing *semantic levels* which integrate (and not simply replace) the lexical level represented by keywords. Semantic levels provide information about word meanings, as described in a reference dictionary, and named entities. This paper focuses on the named entity level. Our aim is to prove that named entities are useful to improve retrieval performance. We exploit a model able to capture entity relationships, although they are not explicit in documents text. Experiments on CLEF dataset prove the effectiveness of our hypothesis.

1 Introduction

In recent years a lot of attention has been invested on Named Entities (NE), and their informative and discriminative power within documents. Due to the importance of research on NE, several sub-areas arose, such as entity detection and extraction, entity disambiguation and entity ranking. The typical information extraction task involving NE is Named Entity Recognition (NER). This task has been defined for the first time during the Message Understanding Conference (MUC) [1], and requires the identification and categorization of NE as entity names (for people and organization), place names, temporal expressions and numerical expressions. Named Entities play also a key role in the Information Retrieval context. Indeed, a very common task in that research area is the entity ranking, whose aim is to retrieve entities (rather than documents) that satisfy the user query. Entity ranking tasks have been proposed by Initiative for the Evaluation of XML retrieval (INEX) since 2007. In 2009, the Text REtrieval Conference (TREC) has proposed an Entity Track whose goal is to perform entity-related search on Web data.

Most documents we deal on everyday contain a lot of references to persons, dates, monetary values and places. Often this information is enough to identify

the context of a document and to understand the main concepts dealt with in it. Moreover, named entity terms are among the most frequently searched terms on the Web. Statistics on Yahoo's top 10 search terms in 2008¹ showed that all the ten search terms consist of named entity terms: six persons, one sport organization, one role-playing game, one fictional character and one TV show.

In this paper we propose a new way of exploiting named entities in Information Retrieval. Although ranked keyword search has been quite successful in the past, this approach has obvious limits basically due to polysemy, the presence of multiple meanings for one word, and synonymy, different words having the same meaning. The result is that, due to synonymy, relevant documents can be missed if they do not contain the exact query keywords, while due to polysemy wrong documents could be deemed as relevant. These problems call for alternative methods that work not only at the lexical level of the documents, but also at the *meaning* level. Named entities mentioned in a document constitute an important part of its semantics. However, when named entities are considered alone they may fail to capture the semantics expressed in a document or in a user query. For that reason we adopt an IR model, called *N-levels*, able to capture semantic information in a text by exploiting *word meanings*, described in a reference dictionary (e.g. WORDNET [2]), and named entities. Thus, we propose an IR system, called *SENSE* (SEmantic N-levels Search Engine), which manages documents indexed at multiple separate levels: keywords, senses (word meanings) and entities (named entities). The system is able to combine keyword search with semantic information provided by the two other indexing levels. In this paper we present the development of the full-fledged entity level based on a novel model called *Semantic Vectors*.

The rest of the paper is structured as follows: Section 2 provides a brief description of SENSE and the N-levels model, along with an overview of keyword and word meaning levels. A detailed description of the entity level is presented in Section 3. Section 4 describes the experiments performed to test system effectiveness improvements, while Section 5 gives a brief discussion about the main work related to our research. Conclusions and future work close the paper.

2 SENSE: SEmantic N-Levels Search Engine

This section presents an IR model, called N-Levels [3], that tries to overcome the limitations of the ranked keyword approach by introducing *semantic levels* which integrate (and not simply replace) the lexical level represented by keywords. Semantic levels provide information about word meanings, as described in a reference dictionary or other semantic resources. The N-Levels model is able to manage documents indexed at separate levels (keywords, word meanings, named entities, and so on) as well as to combine keyword search with semantic information provided by the other indexing levels. The N-Levels model is an open framework to represent different semantic aspects (or levels) pertaining document content. The main idea underlying the work is that there are

¹ <http://buzz.yahoo.com/yearinreview2008/top10/>

several ways to describe the semantics of a document. Each semantic facet needs specific techniques and ad-hoc similarity functions. To address that problem we propose a framework that defines a specific model of Information Retrieval for each semantic level involved in the document representation. Each level corresponds to a *logical view* that aims at describing one of the possible spaces in which documents can be represented. The adoption of different levels is intended to guarantee acceptable system performance even when not all semantic representations are available for a document.

In particular, we suppose that the basic level - keywords - is always available and exploited and, when further levels are available, they are used to offer enhanced retrieval capabilities. Furthermore, our meta-framework allows to use the appropriate representation and similarity measure for each level.

The following semantic levels are currently supported by the framework:

Keyword level - The entry level in which the document is represented by the words in the text.

Word meaning level - The representation at this level is based on *synsets* obtained by WORDNET.

Named entity level - This level consists of entities recognized in the text of the document.

The choice of a specific model for each level should take into account the structure and the meaning of *features* at different levels. Indeed, some features such as word meanings, come with a structure of properties and relationships between them. Hence, term weight measures, such as standard tf-idf, are not suitable for this kind of features.

Analogously, N different levels of representation (one for each level) are needed for representing queries. The N query levels are not necessarily extracted simultaneously from the original keyword query issued by the user: A query level can be obtained when needed.

In addition, the notion of relevance should be extended in order to enhance keyword search with semantic information. Therefore, the degree of similarity $R(q, d)$ between a query q and a document d must be evaluated at each level by defining a proper *local similarity function* that computes document relevance according to the weights defined by the corresponding local scoring function. Since the final goal is to obtain a *single* list of documents ranked in decreasing order of relevance, a *global ranking function* is needed to merge all the result lists that come from each level. This function is independent of both the number of levels and the specific local scoring and similarity functions because it takes as input N ranked lists of documents and produces a unique merged list of the most relevant documents. The aggregation of lists in a single one requires two steps: The first one produces the N normalized lists and the second one merges the N lists in a single one. The two steps are thoroughly described in [3]. We adopt Z-Score normalization and CombSUM [4,5] respectively as score normalization and rank aggregation function.

Both keyword and word meaning levels rely on the classical Vector Space Model [6]. Keyword local scoring function is based on tf-idf, conversely synsets

are scored using an adaptation of tf-idf that takes into account a confidence factor representing the likelihood with which each sense can be associated with a word. More details on both keyword and word meaning levels are provided in [7]. The local similarity functions for both the meaning and the keyword levels are based on cosine similarity. For the meaning level, both query and document vectors contain synsets instead of keywords.

3 Named Entity Level

Named entities are phrases that contain the names of persons, organizations, locations and, more generally, entities that can be identified by proper names. For example:

[ORG CERN] celebrates 20th anniversary of [MISC World Wide Web].
[LOC Geneva], [MISC 13 March 2009]. Web inventor [PERS Tim
Berners-Lee] today returned to the birthplace of his brainchild.

This sentence contains five entities: *Tim Berners-Lee* is a person, *Geneva* is a location, *World Wide Web* and *13 March 2009* are recognized as miscellaneous while *CERN* is an organization. Named entity recognition is an important task of information extraction systems. There has been a lot of work on named entity recognition, especially for English. The Message Understand Conference (MUC) series [1] have offered the opportunity to evaluate systems for English on the same data in a competition. They have also produced a schema for entity annotation. Other system competitions, such as INEX² and CoNLL-2003³, dealt with different languages.

In order to identify named entities in a text, several methods can be applied such as Rule-based, Dictionary-based or Statistical ones. We adopted a statistical method exploiting YamCha [8], a generic open source text chunker useful for a lot of NLP tasks. YamCha adopts a state-of-the-art machine learning algorithm called Support Vector Machines (SVMs), introduced by Vapnik in 1995 [9]. We trained YamCha using the dataset provided by CoNLL-2003 organization during the Shared-Task 2003 [10]. The dataset contains entities extracted from Reuters dataset [11]. In particular three types of entities are extracted: PERSON, LOCATION, ORGANIZATION and MISC, which contains entities that do not belong to the previous three categories.

We used META [12] to extract entities from the CLEF 2008 collection [13]. META is a tool for text analysis that implements several NLP operations, such as entity recognition and indexing. The results of the entity recognition task are exported into a Lucene index. In detail, each document is split in two fields: HEADLINE and TEXT, in compliance with the document structure in CLEF. Each field contains the set of the entities recognized by META and, for each entity, the number of occurrences.

² <http://inex.is.informatik.uni-duisburg.de/>

³ <http://www.cnts.ua.ac.be/conll2003/ner/>

Building the entity level requires four steps:

1. **Pre-processing:** XML files provided by CLEF 2008 organizers are processed in order to extract textual features. The output of this step is a collection of documents compliant to META standard;
2. **Entity extraction:** In this step entities are extracted from documents and are stored in IOB2 format. In IOB2, words outside the Named Entity are tagged with O, while the first word in the entity is tagged with B-k (to begin class k), and further words receive the I-k tag, indicating that these words are inside the entity;
3. **Entity indexing:** Entities extracted in the previous step are stored into an index using Lucene. The entity extraction procedure allows to obtain an entity-based vector space representation, called bag-of-entities (BoE). In this model an entity vector, rather than a word vector, corresponds to a document. Let D be a collection of M documents. Given the j -th document in D :

$$d_j = \langle e_{j1}, e_{j2}, \dots, e_{jn} \rangle, j = 1, \dots, M$$

where e_{jk} is the k -th entity in d_j and n is the total number of entities in d_j . Document d_j is represented in a $|V|$ -dimensional space by an entity-frequency vector f_j , V being the vocabulary for D (the set of distinct entities recognized by META in the collection):

$$f_j = \langle w_{j1}, w_{j2}, \dots, w_{j|V|} \rangle, j = 1, \dots, M$$

where w_{jk} is the weight of the entity e_k in d_j , computed according to a modified version of the TF/IDF score in which entities replace words.

4. **Semantic Vector building:** In this step semantic vectors are built by exploiting the Lucene index. This step is thoroughly described in the next section.

3.1 Semantic Vectors

Semantic Vector models have received considerable attention from researchers in Natural Language processing over the past years, though their invention can be traced back to Salton's introduction of the Vector Space Model for information retrieval [6]. The main idea behind models based on Semantic Vectors is that words and concepts are represented by points in a mathematical space, and this representation is learned from text in such a way that concepts with similar or related meanings are near to one another in that space. This model introduces a range of possible applications, the most immediate one is the "semantic search engine". Semantic Vector models include a family of related models for representing concepts as vectors in a high dimensional vector space, such as Latent Semantic Analysis [14], Hyperspace Analogue to Language [15], and WordSpace [16][17]. The work proposed by Schütze is the most relevant to understand the idea behind WordSpace. The key principle is that vectors representing words found in a particular region of text, called *context vectors*, can be clustered and the centroids of these clusters can be treated as word-senses. Hence,

an occurrence of an ambiguous word can be mapped to one of these word-senses, with a confidence factor computed as the similarity degree between the context vector for that occurrence and the nearest centroid.

Generally speaking, algorithms for creating semantic vectors involve matrix factorization, a computationally expensive task, which introduces scalability bounds. For that reason we decided to adopt the `SemanticVectors` package [18]. This tool is able to build Semantic Vectors using a technique called Random Projection introduced by Kanerva [19]. This method allows to solve efficiently the problem of reducing dimensions, which is one of the key features used to uncover the “latent semantic dimensions” of a word distribution. The main insight of Random Projection is that high dimensional vectors chosen at random are “nearly orthogonal”, in a way that can be formally characterized. Thus, it achieves a result that is for many aspects comparable to orthogonalization methods, such as Singular Value Decomposition, but saving computational resources. Random Projection is able to build semantic vectors with no need for the document-term matrix or the term-term one, because vectors are inferred using an iterative strategy.

The `SemanticVectors` package offers tools for indexing a collection of documents and their retrieval. It relies on Apache Lucene to create a basic term-document matrix. Then the Lucene API is exploited to create a `WordSpace` model from the term-document matrix, by using Random Projection to perform *on-the-fly* dimensionality reduction. This is a relevant point because it allows us to use the same entity index produced by META (see Section 3) to induce semantic vectors. In order to achieve that goal we modified the standard `SemanticVectors` package by adding some ad-hoc features to support CLEF 2008 collection. In particular, documents in CLEF 2008 collection are split in two fields - *headline* and *title* - and are not tokenized using the standard text analyzer in Lucene.

Finally, the idea is to build a semantic space about entities. In that space entities strongly related to each other are represented by similar vectors. In this context, “similar” means that vectors are close in the vector space produced by `SemanticVectors` package.

4 Experimental Session

For the evaluation of the system effectiveness, we used the CLEF Ad Hoc WSD-Robust dataset derived from the English CLEF data, which comprises corpora from “Los Angeles Times” and “Glasgow Herald”, amounting to 166,726 documents and 160 topics in English and Spanish. The relevance judgments were taken from CLEF. The goal of the evaluation was to prove that the combination of three indexing levels outperforms a single level. In particular, that adding the entity level increases the effectiveness of the search with respect to the keyword and meaning levels. To evaluate system effectiveness, different runs were performed by exploiting a single level at a time, or a combination of two or more levels. Each experiment is identified by the names of the used levels.

To measure retrieval performance, we adopted Mean-Average-Precision (MAP)⁴ and Geometric-Mean-Average-Precision (GMAP)⁵ calculated by *trec_eval 0.8.1*, a simple program supplied by the Text REtrieval Conference organizers⁶, on the basis of 1,000 retrieved items per request. Table 1 shows the results for each run, with an overview on the exploited features.

Table 1. Results of the performed experiments

Run	MAP	GMAP
Keyword (K)	0.192	0.041
Meaning (M)	0.188	0.035
K+M	0.220	0.057
Entity (E)	0.134	0.006
K+E	0.220	0.048
M+E	0.228	0.054
K+M+E	0.252	0.076

The results confirm our hypothesis: named entity recognition, in conjunction with an IR model capable of expressing semantics, can greatly improve the retrieval performance. If evaluated individually, the entity level does not yield to satisfactory results. This result is due to the presence of topics in which no entity was recognized. Conversely, when search is performed by making use of multiple levels, the entity level is able to improve performance even on those (difficult) topics for which few relevant documents are returned. This result suggests that named entities play a key role in increasing the number of retrieved relevant results previously ignored. Specifically, considering the experiment *K+M+E* where we used all three levels, an improvement of 14.5% in the MAP and 33.3% in the GMAP was observed. Generally speaking, we noted an overall improvement in all the experiments that used the entity level, compared to the equivalent experiments in which that level was not exploited.

5 Related Work

Several semantic approaches to IR have been proposed in order to improve relevance of results, and many strategies have been used to embody semantic information coming from electronic dictionaries into search paradigms. Mainly, two aspects have been addressed in literature: query expansion with semantically

⁴ Given the Average Precision (AP), that is the mean of the precision scores obtained after retrieving each relevant document, MAP is computed as the arithmetic mean of the AP scores over all topics. Zero precision is assigned to unretrieved relevant documents.

⁵ GMAP computes the geometric mean of the Average Precision over all topics.

⁶ http://trec.nist.gov/trec_eval/

related terms, and semantic similarity measures to compare queries and documents. Query expansion with WORDNET has shown its capability to increase recall [20,21]. On the other hand, semantic similarity measures have the potential to redefine the similarity between a document and a user query [22,23]. Another remarkable attempt to indexing documents according to WORDNET senses, that similar to our approach, is reported in [24]. The authors developed an information retrieval system which performs a combined word-based and sense-based indexing and retrieval. While previous methods tried to *replace* the lexical space with *one* semantic space, in SENSE we defined an adaptation of the vector space model that makes easy the integration of the lexical space with *one or more* semantic spaces. Most IR approaches that exploit named entities are ontological approaches in which named entities are recognized as ontology instances. For instance, the strategy proposed in [25,26] combines keyword search with techniques for navigating and querying ontologies. Specifically, in [25], documents are annotated with concepts in a domain ontology, and indexed using a classical bag-of-words model, while a search tool based on ontology-assisted query rephrasing and keyword search is described in [26]. The main limitation of this approach is that relevance computation is simply performed by using a tf-idf score on concepts, instead of keywords. A gain in retrieval performance using named searching has been shown by the work in [27], where authors proved that proximity-based name searching led to significant improvement over the baseline. More recent attempts try to address the entity ranking problem. In [28], Wikipedia categories and link structure, along with link co-occurrences, are exploited to improve the effectiveness of entity ranking. To leverage the existing development in document-oriented information retrieval, in [29] for each entity a concordance document is created, which consists of all the sentences in the corpus containing that entity. Then, the retrieval is performed over indexes created by these concordance documents.

6 Conclusion and Future Work

We have described and tested SENSE, a semantic N -levels IR system which manages documents indexed at multiple separate levels: keywords, meanings and named entities. We focused on the role of Named Entities in Information Retrieval. Since our aim was to prove that named entities recognized in documents can be useful to improve the effectiveness of the retrieval. Named entities are useful even when no ontology is adopted or relationship recognized, as long as their powerfulness is exploited by a model able to capture their latent semantics. We performed an intensive evaluation using the CLEF Ad Hoc Robust-WSD dataset. This dataset supplies both words and synsets for each document, and it is the ideal framework to evaluate the N -levels architecture. Entities were extracted from the CLEF 2008 collection by using META. The experiments show that the effectiveness of the N -levels model increases when the entity level is involved. As future research, we plan to extend the system capabilities by incorporating mechanisms of relevance feedback and exploiting specialized IR models for each level.

Acknowledgments

This research was partially funded by MIUR (Ministero dell'Università e della Ricerca) under the contract Fondo per le Agevolazioni alla Ricerca, DM19410 "Laboratorio di Bioinformatica per la Biodiversità Molecolare" (2007-2011).

References

1. Grishman, R., Sundheim, B.: Message understanding conference-6: A brief history. In: COLING, pp. 466–471 (1996)
2. Miller, G.A.: Wordnet: a lexical database for english. *Commun. ACM* 38(11), 39–41 (1995)
3. Basile, P., Caputo, A., Gentile, A.L., Degenmis, M., Lops, P., Semeraro, G.: Enhancing Semantic Search using N-Levels Document Representation. In: Bloehdorn, S., Grobelnik, M., Mika, P., Tran, D.T. (eds.) *Proceedings of the Workshop on Semantic Search (SemSearch 2008) at the 5th European Semantic Web Conference (ESWC 2008)*, Tenerife, Spain, June 2, 2008. CEUR Workshop Proceedings, CEUR-WS.org, vol. 334, pp. 29–43 (2008)
4. Fox, E.A., Shaw, J.A.: Combination of Multiple Searches. In: TREC, pp. 243–252 (1993)
5. Lee, J.H.: Analyses of Multiple Evidence Combination. In: SIGIR, pp. 267–276. ACM, New York (1997)
6. Salton, G.: *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River (1971)
7. Basile, P., Caputo, A., Semeraro, G.: UNIBA-SENSE at CLEF 2008: SEMantic N-levels Search Engine. In: *CLEF 2008: Ad Hoc Track Overview (2008)* (CLEF 2008 Working Notes)
8. Kudo, T., Matsumoto, Y.: Fast methods for kernel-based text analysis. In: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pp. 24–31 (2003)
9. Vapnik, V.: *Statistical Learning Theory*. John Wiley, New York (1998)
10. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: Daelemans, W., Osborne, M. (eds.) *Proceedings of CoNLL-2003*, Edmonton, Canada, pp. 142–147 (2003)
11. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research* 5, 361–397 (2004)
12. Basile, P., de Gemmis, M., Gentile, A., Iaquinta, L., Lops, P., Semeraro, G.: META-Multilanguage Text Analyzer. In: *Proc. of the Language and Speech Technology Conference-LangTech.*, pp. 137–140 (2008)
13. Agirre, E., Di Nunzio, G.M., Ferro, N., Mandl, T., Peters, C.: CLEF 2008: Ad Hoc Track Overview. In: *Working notes for the CLEF 2008 Workshop (2008)*, http://www.clef-campaign.org/2008/working_notes/adhoc-final.pdf
14. Landauer, T.K., Dumais, S.T.: A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological Review* 104, 211–240 (1997)
15. Lund, K., Burgess, C.: Producing High-Dimensional Semantic Spaces From Lexical Co-Occurrence. *Behavior Research Methods Instruments and Computers* 28, 203–208 (1996)

16. Sahlgren, M.: The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces. PhD thesis, Stockholm: Stockholm University, Faculty of Humanities, Department of Linguistics (2006)
17. Schütze, H.: Automatic word sense discrimination. *Computational Linguistics* 24(1), 97–123 (1998)
18. Widdows, D., Ferraro, K.: Semantic Vectors: A Scalable Open Source Package and Online Technology Management Application. In: Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008 (2008)
19. Kanerva, P.: Sparse Distributed Memory. MIT Press, Cambridge (1988)
20. Smeaton, A., Kelledy, F., O'Donnell, R.: TREC-4 Experiments at Dublin City University: Thresholding Posting Lists, Query Expansion with WordNet, and POS Tagging of Spanish. In: TREC (1995)
21. Voorhees, E.M.: Query expansion using lexical-semantic relations. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, July 3-6, pp. 61–69 (1994) (Special Issue of the SIGIR Forum)
22. Corley, C., Mihalcea, R.: Measuring the Semantic Similarity of Texts. In: Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, Ann Arbor, Michigan, June 2005, pp. 13–18. Association for Computational Linguistics (2005)
23. Resnik, P.: Semantic Similarity in a Taxonomy: An Information-based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research* 11, 95–130 (1999)
24. Moldovan, D.I., Mihalcea, R.: Using WordNet and Lexical Operators to Improve Internet Searches. *IEEE Internet Computing* 4(1), 34–43 (2000)
25. Davies, J., Weeks, R.: QuizRDF: Search Technology for the Semantic Web. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS 2004)-Track 4, vol. 4, p. 8 (2004)
26. Ducatel, G., Cui, Z., Azvine, B.: Hybrid Ontology and Keyword Matching Indexing System. In: Proceedings of IntraWeb Workshop at WWW2006, Edimburgh (2006)
27. Thompson, P., Dozier, C.: Name searching and information retrieval. In: Proceedings of Second Conference on Empirical Methods in Natural Language Processing, pp. 134–140 (1997)
28. Pehcevski, J., Vercoustre, A.M., Thom, J.A.: Exploiting Locality of Wikipedia Links in Entity Ranking. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 258–269. Springer, Heidelberg (2008)
29. Bautin, M., Skiena, S.: Concordance-Based Entity-Oriented Search. In: Web Intelligence, pp. 586–592. IEEE Computer Society, Los Alamitos (2007)

Detecting Temporal Trends of Technical Phrases by Using Importance Indices and Linear Regression

Hidenao Abe and Shusaku Tsumoto

Department of Medical Informatics, Shimane University, School of Medicine
89-1 Enya-cho, Izumo, Shimane 693-8501, Japan
abe@med.shimane-u.ac.jp, tsumoto@computer.org

Abstract. In this paper, we propose a method for detecting temporal trends of technical terms based on importance indices and linear regression methods. In text mining, importance indices of terms such as simple frequency, document frequency including the terms, and tf-idf of the terms, play a key role for finding valuable patterns in documents. As for the documents, they are often published daily, monthly, annually, and irregularly for each purpose. Although the purposes of each set of documents are not changed, roles of terms and the relationship among them in the documents change temporally. In order to detect such temporal changes, we combined a method to extract terms, importance indices of terms, and trend identification based on linear regression analysis. Empirical results show that our method detected emergent and subsiding trends of extracted terms in a corpus of a research domain. By comparing this method with the existing burst detection method, we investigated the trend of phrases consisting of several burst words in the titles of AAAI and IJCAI.

Keywords: Text Mining, Trend Detection, TF-IDF, Jaccard Coefficient, Linear Regression.

1 Introduction

In recent years, developing information systems in every field such as enterprise, academic and medical organizations, and stored data have increased year by year. Accumulation is advanced to document data by not the exception but various fields. Especially, the document data gives valuable findings not only for domain experts in headquarter sections but also for novice users about particular domains such as day traders, news readers and so forth. In this situation, detecting a new phrases and terms has been very important. In order to realize the detection, emergent term detection (ETD) methods have been developed [12].

However, because the frequency of the word was used in earlier methods, detection was difficult as long as the word that became an object did not appear. In addition, almost of the conventional methods are not consider nature of terms

and importance indices separately. This causes difficulties of text mining applications such as limitations for extensionality of time direction, time consuming post-processing, and other generality expansions. By considering the problem, we focus on temporal changes of importance indices of phrases and changes of them. The temporal changes of the importance indices of extracted phrases paid to attention so that the specialist may recognize an emergent terms or/and such fields in this research.

In this paper, we propose a method to detect trends of phrases by combining term extraction methods, importance indices of the terms and trend analysis methods in Section 3. Then, taking as an example of the titles and abstracts of IEEE International Conference of Data Mining (ICDM)¹, we show the comparison of changes in two importance indices in Section 4. In Section 5, we compared the trend of phrases consisting of several burst words in the titles of AAAI and IJCAI. Finally, in Section 6, we summarize this paper, and describe our future work.

2 Related Work

In order to detect emergent terms/themes/topics in textual data, there are some conventional studies. As the first step, [1] proposed a method to find temporal trends of words. Then, applying various metrics such as frequency [3], n-gram [4] and tf-idf [5], some kind of emergent term detection (ETD) methods were developed [2]. As one of the advantaged method of ETD, [6,7] suggested a method to find emergent theme patterns based on finite state machine by using Hidden Markov Model(HMM). Topic modeling [8] is one of the related methods from the view point of the temporal text analysis. In these methods, they consider the changes of each particular index of the terms rather than considering the nature of the terms on each language model.

Besides, in the natural language processing field, there are studies to find out meaningful terms in the document [9,10]. One method is based on χ^2 statistics of co-occurrence of nouns. [10] proposes a method to determine meaningful terms based on adjacent frequency of compound nouns. By focusing on the methods to find out meaningful terms consisting of two or more words based on co-occurrence, [11] suggested a method to extract technical terms consisting of co-existing nouns by calculating χ^2 statistics on the contingency matrix of occurrences of each pair of nouns in a given corpus.

In the conventional studies to detect emergent words or/and phrases in documents such as Web pages and particular electric message boards, they did not explicitly treat the trends of the calculated indices of words or/and phrases. However, base on the two different techniques, we consider a method to detect trends of phrases, which consist of from two to nine words. The reason why we focused on short phrases is that too long phrase may be a pattern including grammatical structure and anonymous words as shown in [7].

¹ <http://www.cs.uvm.edu/~icdm/>

3 A Method to Detect Trends of Importance Indices of Automatically Extracted Terms

In this section, we describe a method to detect some temporal trends of technical terms by using multiple important indices consisting of the following three sub-processes:

1. Technical term extraction in a corpus
2. Importance indices calculation
3. Trend detection

There are some conventional studies to extract technical terms in a corpus based on each particular importance index [2]. Although these methods calculate each index to extract technical terms, the information of the importance of each term are lost by cutting off the information with a threshold value. We suggest separating term determination and trends detection based on importance indices. By separating these phases, we can calculate multiple kinds of importance indices. Subsequently, to the dataset, we can apply many kinds of temporal analysis methods based on statistical analysis, clustering and machine learning algorithms. The overview of this method is illustrated in Figure 1.

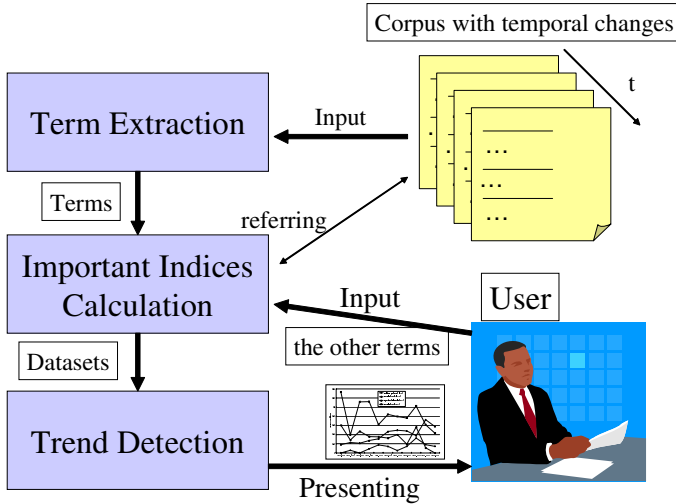


Fig. 1. An overview of the proposed remarkable temporal trend detection method

Firstly, the system determines terms in a given corpus. There are two reasons why we introduced term extraction methods before calculating importance indices. One is that the costs to build up a dictionary for each particular domain are very expensive task. The other is the need to detect new concepts in a given temporal corpus. Especially, a new concept is often described in the document for which the character is needed at the right time in using the combination

of existing words. By considering above reasons, we applied a term extraction method based on adjacent frequency of compound nouns. The method extracts the technical terms by using the following values for each candidates CN :

$$FLR(CN) = f(CN) \times \left(\prod_{i=1}^L (FL(N_i) + 1)(FR(N_i) + 1) \right)^{\frac{1}{L}} > 1.0$$

Where $f(CN)$ means frequency of the term CN consisting of L nouns. Similarly, $FL(N_i)$, and $FR(N_i)$ mean frequencies of the right and left of the object noun N_i . In order to determine terms in this part, we can also use the other term extraction methods and terms/keywords from users.

After determining terms in the given corpus, the system calculates multiple importance indices of the terms for the documents in each period. As for importance indices of words and phrases in a corpus, there are some well known indices. Term frequency divided by inversed document frequency (tf-idf) is one of the popular indices to measure the importance of the terms. The definition of tf-idf is shown in the following:

$$tfidf(t) = tf(t) \times \left(\log_e \frac{|D_{period}|}{df(t)} \right)$$

Where $tf(t)$ means the frequency of each term t in the corpus with $|D_{period}|$ documents included in each period. And $df(t)$ means the frequency of documents containing w_i , which are the words included in the term t . As another importance index, we used Jaccard’s matching coefficient [12] [2] calculated as the following:

$$Jaccard(t) = \frac{h(w_1, w_2, \dots, w_L)}{h(w_1)h(w_2)\dots h(w_L)}$$

Where $h(w_i)$ means the number of hit documents in the corpus to the word w_i . Each Jaccard coefficient value shows strength of co-occurrence of multiple words as an importance of the terms in the given corpus. We can also assume the degrees of co-occurrence such as the χ^2 statistics to the terms consisting of multiple words as the importance indices in our method.

In our method, we propose treating these indices explicitly as a temporal dataset. Figure 2 shows an example of the dataset consisting of an importance index for the years.

Then, the method provides the choice of some adequate trend extraction method such as linear regression analysis, clustering and so forth to the datasets. In the following case study, we applied the linear regression analysis technique in order to detect the degree of existing trends based on the two importance indices. The degree of each term t calculated as the following:

$$Deg(t) = \frac{\sum_{i=1}^M (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^M (x_i - \bar{x})^2},$$

² Here after, we call just “Jaccard coefficient”.

Term	Jacc 1996	Jacc 1997	Jacc 1998	Jacc 1999	Jacc 2000	Jacc 2001	Jacc 2002	Jacc 2003	Jacc 2004	Jacc 2005
output feedback	0	0	0	0	0	0	0	0	0	0
H/sub infinity	0	0	0.012876	0	0.00885	0	0	0	0.005405	0.003623
resource allocation	0.006060606	0	0	0	0	0	0	0	0	0
image sequences	0	0	0	0	0	0	0	0.004785	0	0
multiagent systems	0	0	0	0	0	0	0.004975	0	0	0
feature extraction	0	0.005649718	0	0.004484	0	0	0	0	0	0
images using	0	0	0	0	0	0.004673	0	0	0	0
human-robot interaction	0	0	0	0	0.004425	0	0	0	0	0
evolutionary algorithm	0	0.005649718	0	0.004484	0	0	0	0	0.002703	0.003623
deadlock avoidance	0	0	0	0	0.004425	0	0	0	0	0
ambient intelligence	0	0	0	0	0	0	0	0	0	0.003623
feature selection	0	0	0	0	0	0	0	0	0.002703	0
data mining	0	0	0	0	0.004425	0	0	0	0.002703	0

Fig. 2. Example of the dataset consisting of an importance index

where \bar{x} means the average of the period, and \bar{y} means the average of each importance index for the period $1 \leq i \leq M$. At the same time, we also calculated the intercept $Int(t)$ of each term t as the following:

$$Int(t) = \bar{y} - Deg(t)\bar{x}$$

4 Experiment: Detecting Remarkable Trends of Technical Phrases in a Temporal Corpus

In this experiment, we show the results of detecting two trends by using the method described in Section 3. As the input of temporal documents, annual sets of titles and abstracts of ICDM conference from 2002 to 2008 are taken.

In these corpus, we determined technical terms by using the term extraction method [10] for each entire corpus.

Subsequently, tf-idf and Jaccard coefficient values are calculated for each term to the annual documents on each corpus. To the datasets consisting of temporal values of the important indices, we applied linear regression to detect the following two trends of the phrases: Emergent and Subsiding.

4.1 Extracting Technical Terms

As for the documents, we assumed each title and abstract of the article as one document. Then, we did not use any stemming technique, because we want to consider detailed difference of the terms.

Table 1 shows the description of the abstracts and titles of ICDM conferences from 2002 to 2008.

We applied the term extraction method to all of the abstracts and the titles of the seven years' ICDM conferences. From of the entire abstracts for the seven years, the method extracted 21,599 terms.

As same as to the abstracts, 1,912 terms are extracted in the entire titles for the seven years.

³ The implementation is called Gensen, distributed in <http://gensen.dl.itc.u-tokyo.ac.jp/termextract.html> (in Japanese).

Table 1. Description of the ICDM corpus

Year	Abstract		Title	
	#documents	#words	#documents	#words
2002	112	18,916	112	960
2003	125	19,068	125	1,040
2004	106	15,985	106	840
2005	141	20,831	141	1,153
2006	152	24,217	152	1,307
2007	101	16,143	101	782
2008	144	22,971	144	1,136
TOTAL	881	138,131	881	7,218

4.2 Results of the Automatically Extracted Terms

By using the degree and the intercept of each term, we tried to determine the following two trends:

- Emergent
 - sorting the degree with ascending order
 - sorting the intercept with descending order
- Subsiding
 - sorting the degree with descending order
 - sorting the intercept with ascending order

Table 2 shows the top ten emergent phrases extracted in the abstracts of ICDM having the two trends based on the two importance indices. Table 3 shows the top ten emergent terms extracted in the titles of ICDM.

Table 2. Top 10 terms with the emergent trend based on tf-idf and Jaccard coefficient values in the abstracts of ICDM

Rank	tf-idf			Jaccard coefficient		
	t	Deg(t)	Int(t)	t	Deg(t)	Int(t)
1	text mining	3.84	-0.06	collaborative filtering	0.08	0.28
2	feature extraction	3.58	-2.76	upper bound	0.06	0.09
3	SVM classifier	3.40	-4.82	chemical compounds	0.06	0.13
4	subspace clustering	3.28	-1.59	false alarm	0.06	0.20
5	social network	2.88	-0.18	social networks	0.05	-0.03
6	random walk	2.66	-0.13	social network	0.05	0.03
7	matrix factorization	2.65	-0.53	Latent Dirichlet Allocation	0.05	-0.06
8	Experimental results	2.57	17.84	pairwise constraints	0.05	-0.02
9	social networks	2.56	-1.81	gene expression	0.05	0.48
10	labeled data	2.52	1.29	matrix factorization	0.05	0.04

As similar to the emergent phrases, Table 4 and Table 5 show the phrases with the subsiding trends respectively.

On each set of documents, our method identified similar phrases both of the two importance indices for each temporal trend. “Social Network” shows the most typical emergent trend with both of the importance indices. This indicates

Table 3. Top 10 terms with the emergent trend based on tf-idf and Jaccard coefficient values in the titles of ICDM

Rank	tf-idf			Jaccard coefficient		
	t	Deg(t)	Int(t)	t	Deg(t)	Int(t)
1	Data Streams	1.93	2.78	Collaborative Filtering	0.13	0.10
2	Nonnegative Matrix Factorization	1.43	-1.68	Nonnegative Matrix Factorization	0.10	-0.05
3	Active Learning	1.15	0.28	Random Walk	0.10	0.05
4	Social Networks	1.09	-0.74	Dimension Reduction	0.09	0.02
5	Collaborative Filtering	1.06	1.19	Social Networks	0.05	0.01
6	Sequential Pattern Mining	1.04	-0.33	Data Streams	0.03	0.03
7	Random Walk	1.04	0.18	Active Learning	0.02	0.04
8	Dimension Reduction	0.74	0.33	Anomaly Detection	0.01	0.13
9	Clustering Algorithm	0.68	1.29	Sequential Pattern Mining	0.01	-0.01
10	Evolving Data	0.63	0.46	Document Clustering	0.01	0.02

Table 4. Top 10 terms with the subsiding trend based on tf-idf and Jaccard coefficient values in the abstracts of ICDM

Rank	tf-idf			Jaccard coefficient		
	t	Deg(t)	Int(t)	t	Deg(t)	Int(t)
1	association rules	-6.01	39.48	association rules	-0.06	0.46
2	association rule	-4.81	29.42	association rule	-0.05	0.42
3	frequent itemsets	-4.06	38.44	pruning strategy	-0.04	0.23
4	web pages	-2.75	18.19	easily extended	-0.04	0.23
5	Bayesian network	-2.75	13.26	web pages	-0.04	0.48
6	mining algorithm	-2.64	18.08	nearest neighbor	-0.04	0.50
7	association rule mining	-2.25	14.21	Bayesian network	-0.04	0.20
8	web site	-2.25	12.64	nearest neighbors	-0.04	0.36
9	sequential patterns	-2.14	22.02	frequent itemsets	-0.04	0.37
10	mining frequent itemsets	-1.98	10.16	neural network	-0.03	0.16

Table 5. Top 10 terms with the subsiding trend based on tf-idf and Jaccard coefficient values in the titles of ICDM

Rank	tf-idf			Jaccard coefficient		
	t	Deg(t)	Int(t)	t	Deg(t)	Int(t)
1	Association Rules	-3.39	18.22	Association Rules	-0.09	0.58
2	Data Sets	-1.77	9.80	Decision Trees	-0.08	0.64
3	Latent Semantic Indexing	-0.79	4.88	Latent Semantic Indexing	-0.06	0.39
4	Decision Trees	-0.77	7.11	Feature Selection	-0.04	0.53
5	Pattern Discovery	-0.70	5.75	Pattern Discovery	-0.03	0.23
6	Data Mining	-0.59	14.52	Data Sets	-0.02	0.11
7	Frequent Itemset Mining	-0.51	3.84	Dimensional Data	-0.01	0.05
8	Dimensional Data	-0.50	4.02	Model-based Clustering	-0.01	0.05
9	Model-based Clustering	-0.41	3.84	Data Mining	0.00	0.10
10	Noisy Data	-0.10	3.53	Frequent Itemset Mining	0.00	0.03

that both of the standardized frequency of the phrases and the characteristic combinations of the words included in the phrases has been focused as an important topic in these years by both of the reviewers and the authors.

As shown in Table 2 and Table 3, our method can detected similar phrases as emergent phrases. Although these phrases are different on their representations, human experts can understand that their meanings are similar.

5 Discussion

In this section, we discuss about the difference between our method and the existing burst detection method by comparing the degrees and the intercepts of

several phrases including the burst words. In addition, we also discuss about the other indices to score phrases in a set of documents.

5.1 Comparison about Burst Words and the Degrees of the Importance Indices

In [6], it proposed a method to detect bursty words that occur with high intensity over a limited period of time. The analysis uses a probabilistic automaton whose states correspond to the frequencies of individual words. To the titles of several famous international conferences related to computer science, the bursty words are detected [4].

From AAAI and IJCAI titles, the method detected the following words as currently bursting: auctions, combinational, reinforcement. Table 6 shows the phrases including the three words with the degrees and intercepts of tf-idf and Jaccard coefficient. In order to eliminate specific paper, we selected the terms which appeared more than two times.

As shown in these degrees, they show positive degree with small intercepts. This means that our method also determined these terms as emergent based on the two importance indices in Text Mining and Information Retrieval. Since our method can detect the degree of emergent as continuously, our method has a feasibility to detect various emergent patterns of the terms.

Table 6. Degrees and intercepts of tf-idf and Jaccard coefficient of the phrases including auctions, combinational, and reinforcement

	Term	TFIDF Deg	TFIDF Int	Jacc Deg	Jacc Int
AAAI	Combinatorial Auctions	0.2171	-0.9177	0.0118	-0.0580
	Combinatorial Auction	0.1294	-0.4290	0.0056	-0.0197
	Auctions	0.4283	-2.0073	0.0559	-0.2688
	Reinforcement Learning	0.0049	-0.0099	0.0072	-0.0102
	Reinforcement Learning Algorithm	0.0006	-0.0011	0.0003	-0.0006
IJCAI	Reinforcement Learning	0.5760	-1.6870	0.0049	-0.0099
	Reinforcement Learning Approach	0.0786	-0.1127	0.0006	-0.0011

Table 7. Other indices to score terms in a set of documents

Index name	Definition
Term Frequency	$tf(t)$
Document Frequency	$df(t)$
Support/Coverage	$tf(t)/ D_{period} , df(t)/ D_{period} $
Odds	$df(t)/(D_{period} - df(t))$
Cosine Similarity	$tf(t) / \sqrt{tf(w_1)tf(w_2)..tf(w_L)}$

⁴ They can be found in <http://www.cs.cornell.edu/home/kleinber/kdd02.html>

5.2 Other Importance Indices

In the above mentioned experiments, we did not use the frequencies of terms $tf(t)$ and the frequencies of documents including the terms $df(t)$. Similar to the evaluation indices of association rules [13], we introduced the indices to sequential item patterns. Table 7 shows a part of indices and their definitions to score each term t on a set of documents D_{period} . In order to introduce more indices, we have to consider the sequential constraints of terms. This should be a basis to integrate frequently item sequence mining and text mining.

6 Conclusion

In this paper, we proposed the method to detect trends of technical terms by focusing on the temporal changes of the importance indices. We implemented the method by combining the technical term extraction method, the two important indices, and linear regression analysis.

The case study shows that the temporal changes of the importance indices can detect the trend of each term, according to the degree of the values for each annual document. The emergent terms, which detected by a domain expert, are ranked as the terms with increasing degrees of the importance indices. Regarding to the result, our method can support to find out trends of terms in documents based on the temporal changes of the importance indices.

In the future, we will apply other term extraction methods, importance indices, and trend detection method. As for importance indices, we are planning to apply evaluation metrics of information retrieval studies, probability of occurrence of the terms, and statistics values of the terms. To extract the trends, we will introduce temporal pattern recognition methods, such as temporal clustering. Then, we will apply this framework to other documents from various domains.

References

1. Lent, B., Agrawal, R., Srikant, R.: Discovering trends in text databases, pp. 227–230. AAAI Press, Menlo Park (1997)
2. Kontostathis, A., Galitsky, L., Pottenger, W.M., Roy, S., Phelps, D.J.: A survey of emerging trend detection in textual data mining. *A Comprehensive Survey of Text Mining* (2003)
3. Swan, R., Allan, J.: Automatic generation of overview timelines. In: *SIGIR 2000: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 49–56. ACM, New York (2000)
4. Shannon, C.E.: A mathematical theory of communication. *The Bell System Technical Journal* 27, 379–423 (1948)
5. Sparck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. *Document retrieval systems*, 132–142 (1988)
6. Kleinberg, J.M.: Bursty and hierarchical structure in streams. *Data Min. Knowl. Discov.* 7(4), 373–397 (2003)

7. Mei, Q., Zhai, C.: Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In: KDD 2005: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 198–207. ACM, New York (2005)
8. Wallach, H.M.: Topic modeling: beyond bag-of-words. In: ICML 2006: Proceedings of the 23rd international conference on Machine learning, pp. 977–984. ACM, New York (2006)
9. Frantzi, K.T., Ananiadou, S.: Extracting nested collocations. In: Proceedings of the 16th conference on Computational linguistics, Morristown, NJ, USA, pp. 41–46. Association for Computational Linguistics (1996)
10. Nakagawa, H.: Automatic term recognition based on statistics of compound nouns. *Terminology* 6(2), 195–210 (2000)
11. Yutaka Matsuo, M.I.: Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools* 13(1), 157–169 (2004)
12. Anderberg, M.R.: *Cluster Analysis for Applications*. Monographs and Textbooks on Probability and Mathematical Statistics. Academic Press, Inc., New York (1973)
13. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right interestingness measure for association patterns. In: Proceedings of International Conference on Knowledge Discovery and Data Mining KDD 2002, pp. 32–41 (2002)

Detecting Emotions in Classical Music from MIDI Files

Jacek Grekow¹ and Zbigniew W. Ras²

¹ Bialystok Technical University, Faculty of Computer Science, Wiejska 45A,
Bialystok 15-351, Poland
grekowj@wi.pb.edu.pl

² University of North Carolina, Computer Science Dept., 9201 University City Blvd.,
Charlotte, NC 28223, USA
ras@uncc.edu

Abstract. At a time when the quantity of sounds surrounding us is rapidly increasing and the access to different recordings as well as the amount of music files available on the Internet is constantly growing, the problem of building music recommendation systems including systems which can automatically detect emotions contained in music files is of great importance. In this article, a new strategy for emotion detection in classical music pieces which are in MIDI format is presented. A hierarchical model of emotions consisting of two levels, L1 and L2, is used. A collection of harmonic and rhythmic attributes extracted from music files allowed for emotion detection with an average of 83% accuracy at level L1.

Keywords: Music Information Retrieval, Emotion Detection.

1 Introduction

Music has accompanied man for ages in various situations. We hear it in advertisements, in films, at parties, at the philharmonic, in clubs, etc. One of the most important functions of music is its effect on man. Certain pieces of music have a relaxing effect, while others stimulate us to act, and some cause a change in or emphasize our mood. Music is not only a great number of sounds arranged by a composer, it is also the emotion contained within these sounds. At a time when the quantity of sounds surrounding us is rapidly increasing and the access to different recordings as well as the amount of music files available on the Internet is constantly growing, the problem of building music recommendation systems including systems which can automatically detect emotions contained in music files is of great importance.

1.1 Input Data

Many research papers deal with the problem of emotion detection. Some of them rely on audio files [5], [6], [7], [11], [12], [14] and others on MIDI files [1], [8]. In our research, we concentrated on emotion detection in MIDI files containing symbolic

representation of music (key, structure, chords, instrument). The means of representation of music content in MIDI files is much closer to the description which is used by musicians, composers, and musicologists. To describe music, they use key, tempo, scale, sounds, etc. This way, we avoid the difficult stage of extraction of separate notes, tracks, instruments from audio files, and we can concentrate on the deciding element which is the music content.

1.2 Mood Model

There are several models describing emotions contained in music. One of them is the model proposed by Hevner [4]. This model is made up of a list of adjectives grouped in 8 main categories. After modification it was used by Li et al. [5] and Wieczorkowska et al. [12]. This model is quite developed and complex, too complicated to use in our experiment, however, it illustrates the intricacy of describing emotions.

Another model is the two-dimensional Thayer model [9] in which the main elements are Stress and Energy laid out on 2 perpendicular axes. Stress can change from happy to anxious, and Energy varies from calm to energetic. This way, 4 main categories form on the plain: Exuberance, Anxious, Depression and Contentment. This model was used by Liu et al. [7], DiPaola et al. [1], Wang et al. [11], Yang et al. [14].

The model we chose is based on Thayer's model (Fig. 1). Following the example of this model, we created a hierarchical model of emotions consisting of two levels, L1 and L2.

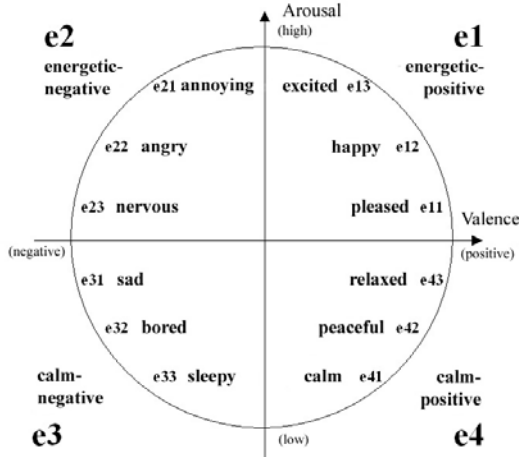


Fig. 1. Thayer's arousal-valence emotion plane

The first level L1 contains 4 emotions. To ease the indexing of files, group names were replaced with compound adjectives referencing Arousal and Valence. Our mood model contains the following groups (Table 1):

Table 1. Description of mood groups in L1, the first level

Abbreviation	Description
e1	energetic-positive
e2	energetic-negative
e3	calm-negative
e4	calm-positive

In the first group (e1), pieces of music can be found which convey positive emotions and have a quite rapid tempo, are happy and arousing (Excited, Happy, Pleased). In the second group (e2), the tempo of the pieces is fast, but the emotions are more negative, expressing Annoying, Angry, Nervous. In the third group (e3), are pieces that have a negative energy and are slow, expressing Sad, Bored, Sleepy. In the last group (e4), are pieces that are calm and positive and express Calm, Peaceful, Relaxed.

The second level is related to the first, and is made up of 12 sub-emotions, 3 emotions for each emotion contained in the first level (Table 2).

Table 2. Description of mood groups in L2, the second level

Abbreviation	Description	Abbreviation	Description
e11	pleased	e31	sad
e12	happy	e32	bored
e13	excited	e33	sleepy
e21	annoying	e41	calm
e22	angry	e42	peaceful
e23	nervous	e43	relaxed

2 Method

2.1 Database

A database with 83 MIDI files of classical music (F. Chopin, R. Shuman, F. Schubert, E. Grieg, F. Mendelssohn-Bartholdy, etc.) was created specifically for the needs of the experiment. Starting from the 5th bar, 16 second segments were isolated from each piece. The shift forward was chosen with the aim of avoiding various, unstable introductions at the beginning of many pieces. Each 16 second segment was divided into 6 subsegments of 6 seconds each with a mutual overlap (overlapping 2/3). There were 498 resulting 6-second subsegments. Overlapping allows for precise tracking of emotion contained within musical segments.

2.2 Indexing

The 498 subsegments were annotated with an emotion by a listener-tester, a person with a formal music education/background, who has professional experience in listening to music.

2.3 Feature Extraction

The next stage was to obtain features describing the files in the database. Specially written software “AKWET simulator - Features Explorer” was used in connection with the program MATLAB, every record in the database was described with 63 features.

Harmony Features




Harmony, along with rhythm and dynamics, is one of the main elements of music upon which emotion in music is dependent. Harmony Features reflect dissonance and consonance of harmony of sounds. They are based on previous work by the author [2], [3]. To calculate the harmony parameters, we used the frequency ratio of simultaneously occurring sounds (Table 3).

A given consonance (interval, chord, polyphone) comprises of simultaneously resonant sounds, the frequency ratio of which can be noted as following:

$$N_{R1}: N_{R2}: \dots : N_{Rk} \tag{1}$$

where k is the number of sounds comprising the consonance.

Table 3. Example consonance sound frequency ratios

k	Musical notation	Consonance sound frequency ratios $N_{R1}: N_{R2}: \dots : N_{Rk}$
2		2:3
3		4:5:6
4		25:30:36:45

From the frequency ratios, we calculated the AkD parameter, which mirrors the degree of dissonance in a single chord. The higher its value, the more dissonant is the consonance; when the AkD value is lower, the consonance is more consonant – more pleasant for the ear.

$$AkD = LCM (N_{R1}, N_{R2}, \dots, N_{Rk}) \tag{2}$$

where k is the number of sounds in a given sample. In the case when $k = 1$, then $AkD = 1$. LCM means Least Common Multiple.

From the sequence of consonance samples collected from a musical segment (Fig. 2), the table can be defined as:

$$AkD_s = (AkD_1, AkD_2, \dots, AkD_p) \tag{3}$$

where p is the number of samples collected from a given segment.

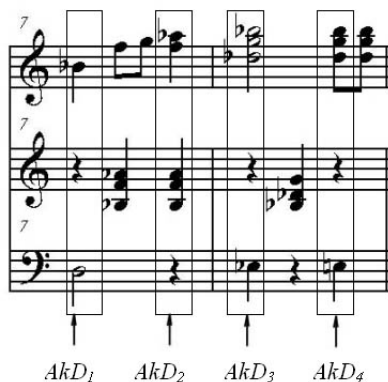


Fig. 2. Process of sample collection from a segment

The moments of sample collection from a segment have been defined according to two criteria. The first is the collection of samples at every eighth, and the second is the collection of samples at every new chord in a segment.

Table 4. Main harmony features

Feature group	Main features
Basic statistical functions	Average AkD_s Standard deviation of AkD_s Number of samples in AkD_s First max in AkD_s Second max in AkD_s Third max in AkD_s
Common values	First common value in AkD_s Second common value in AkD_s Third common value in AkD_s
Chord location	Average amplitude of sound in a chord Standard deviation of sound in a chord

Harmony features describe what kind of harmony occurs in a given segment, which ones dominate, how many of them occur, etc. (Table 4). Below is a presentation of AkD samples (chords) collected at every eighth in a segment from Étude Op.10 No 5 by F. Chopin (Fig. 3).

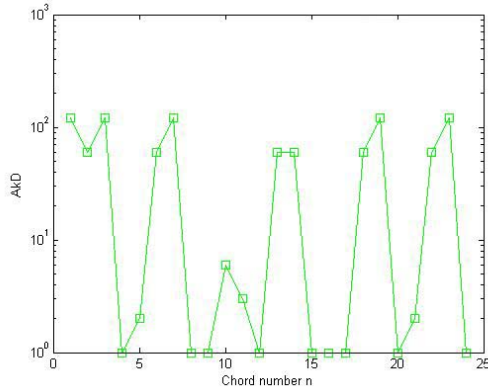


Fig. 3. *AkD* for fragment of F. Chopin Etude Op.10 No 5

Rhythmic Features

Rhythmic features represent rhythmic regularity in a given segment of music. These features were obtained from the beat histogram, which was acquired from the calculation of autocorrelation [10].

$$autocorrelation[lag] = \frac{1}{N} \sum_{n=0}^{N-1} x[n]x[n-lag] \tag{4}$$

where *n* is the input sample index (in MIDI ticks), *N* is total number of MIDI ticks in segment and *lag* is delay in MIDI ticks ($0 < lag < N$). The value of *x[n]* is proportional to the velocity of Note On events.

The histogram was transformed so that each bin corresponded to a periodicity unit of beats per minute (Fig. 4).

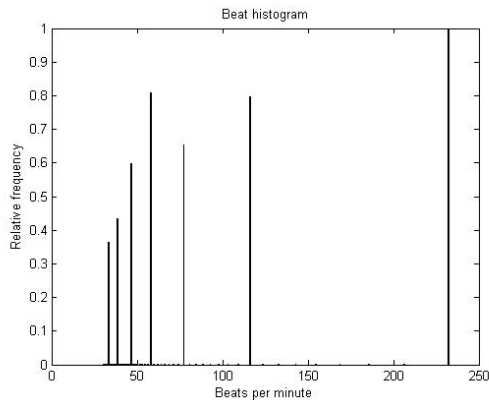


Fig. 4. Beat histogram for fragment of F. Chopin Etude Op.10 No 5

Rhythmic features describe the strongest pulses in the piece, relations between them, their quantity, etc. (Table 5).

Table 5. Main rhythmic features

Feature group	Main features
Strongest Rhythmic Pulses	First Strongest Rhythmic Pulse Second Strongest Rhythmic Pulse Third Strongest Rhythmic Pulse
Pulse Ratios	Ratios of Strongest Pulses
Relatively Strong Pulses	Number of Relatively Strong Pulses Number of beat peaks with a value of 30-50% of relative frequency
Rhythmic Note values	Average Note Duration - Average duration of notes in seconds Note Density - Average number of notes per second

Correlations between Features

Individual features, such as harmony or dynamics, are related to rhythm. They are often correlated. The moment of appearance of a given accent, chord, etc. in the bar is of great significance. The most important and significant parameters were obtained through the correlation of parameters with rhythm.

We created an AkD_B data table. It comprises of AkD samples collected from musical segments at moments of the strongest pulses.

$$AkD_B = (AkD_1, AkD_2, \dots, AkD_b) \tag{5}$$

where b is the number of collected samples at moments of the strongest pulses. All values from the beat histogram which are more than 50% of the strongest (Strongest Rhythmic Pulse) in a beat histogram were accepted as the strongest pulses. Next, statistical features were calculated, similarly as with AkD_s (Table 4).

Dynamic Features

Dynamic features are based on the intensity of sound, the length of sounds, and their development in a segment (Table 6).

Table 6. Main dynamic features

Feature group	Main features
Basic statistical functions	Average of loudness levels of all notes Standard deviation of loudness levels of all notes

The last stage consisted of exporting of the obtained data to Arff format, allowing for data analysis in the WEKA program.

2.4 Mood Detection

Describing emotions contained within a given segment is not always clear-cut. Some segments contain a single emotion, while others can contain several emotions

simultaneously. In order to allow the tester to mark his opinion, the choice of many emotions for a segment was permitted. This allowed the tester to assign not only a single emotion but several to the consecutive examples. Marking an emotion from the lower level, L2, automatically caused the marking of the appropriate emotion from the higher level, L1.

3 The Experiment Results

The program WEKA was used to carry out the experiments, which allowed for testing data utilizing many methods [13].

Because many musical segments were labeled by many labels simultaneously, multi-label classification in emotion detection was used (multi-label decision attribute was replaced by a set of binary decision attributes representing emotions). The same, we transformed data into several two-class types of data and tested one against the rest of the data. For each class, a data set was generated containing a copy of each instance in the original data, but with a modified class value. If the instance had the class associated with the corresponding dataset it was tagged YES, otherwise, it was tagged NO. The classifiers were built for each of these binary data sets. The proposed strategy greatly simplified the process of building classifiers for a decision system with a multi-label decision attribute.

The classification results were calculated using a cross validation evaluation CV-10. We used attribute selection to find the best subset of attributes. The best result was achieved by using Wrapper Subset Evaluator. After testing the data utilizing many methods, one of the best results was achieved with the use of the k-NN classifier (k-nearest neighbors). The use of attribute selection improved the accuracy of classifiers by an average of about 10% (Fig. 5).

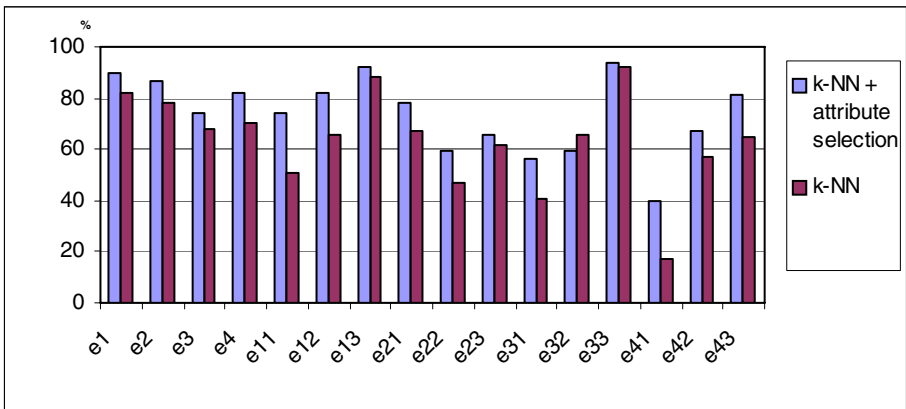


Fig. 5. Comparison of results attained using the k-NN algorithm with and without the use of attribute selection

A classifier was created for each emotion separately. L1 first level classifiers (Table 7) are more accurate than L2 second level classifiers. This is connected with the fact that the groups of examples from the first level are larger as well as the

Table 7. Coverage factor of L1 first level classifiers

Classifier	Emotion	No. of objects	Coverage factor
e1	energetic-positive	151	90%
e2	energetic-negative	172	87%
e3	calm-negative	111	74%
e4	calm-positive	103	82%

Table 8. Coverage factor of L2 second level classifiers

Classifier	Emotion	No. of objects	Coverage factor
e11	pleased	66	74%
e12	happy	69	82%
e13	excited	19	92%
e21	annoying	37	78%
e22	angry	52	59%
e23	nervous	82	66%
e31	sad	47	56%
e32	bored	52	59%
e33	sleepy	12	94%
e41	calm	17	40%
e42	peaceful	30	67%
e43	relaxed	56	81%

emotions are much easier to recognize for the listener. The most accuracy was attained for emotion classifier e1 – energetic-positive (90%), and the least accuracy was attained for emotion classifier e3 – calm-negative (74%).

The accuracy of L2 second level classifiers (Table 8) is somewhat less accurate than L1 first level classifiers, and fluctuates from 40-92%. This is connected with the fact that the example groups for specific emotions are smaller as well as that the recognition of these emotions – on this more precise level – is more difficult for the listener. The least accuracy was attained for emotion classifier e41 – calm. Also, e42 is not high, which is tied to the fact that the division of emotions into groups e41 – calm and e42 – peaceful is not the most apt. These are rather difficult for the listener to distinguish. In the future, for further research, these two groups should be combined into one. The best results (80-90%) were obtained for emotions e12 (happy), e13 (excited), e33 (sleepy), and e43 (relaxed). These are the most easily recognized emotions by the listener, and it is rather difficult to confuse them with other emotions.

4 Conclusion

In this article, we presented emotion detection in pieces of classical music in the form of MIDI files. A hierarchical model of emotions consisting of two levels, L1 and L2, is used. A collection of harmonic and rhythmic attributes extracted from music files allowed for emotion detection with an average of 83% accuracy at level L1.

We plan to find emotional profiles of different users using the music file search system, which searches for files according to emotion. This will be achieved through testing on a larger group and through grouping them according to their responses. This should resolve the problem of subjective emotion assessment by different users. We also plan to expand the collection of attributes as well as enhance the file database by adding other genres of music.

Acknowledgments. This paper is supported by the S/WI/5/08.

References

1. DiPaola, S., Arya, A.: Emotional Remapping of Music to Facial Animation. In: ACM Siggraph 2006 Video Game Symposium Proceedings, Boston (2006)
2. Grekow, J.: An analysis of the harmonic content – main parameters in the AKWET method. In: II Konferencja Technologii Eksploracji i Reprezentacji Wiedzy, TERW 2007, Hołny Mejera (2007)
3. Grekow, J.: Broadening musical perception by AKWEDs technique visualization. In: Proceedings of the 9th International Conference on Music Perception and Cognition, ICMPC9 (2006)
4. Hevner, K.: Experimental studies of the elements of expression in music. *American Journal of Psychology* 48, 246–268 (1936)
5. Li, T., Ogihara, M.: Detecting emotion in music. In: Proceedings of the Fifth International Symposium on Music Information Retrieval (ISMIR 2003), pp. 239–240 (2003)
6. Liu, C., Yang, Y., Wu, P., Chen, H.: Detecting and Classifying Emotion in Popular Music. In: Proceedings of the 9th Joint Conference on Information Sciences (JCIS)/CVPRIP (2006)
7. Liu, D., Lu, L., Zhang, N.: Automatic mood detection from acoustic music data. In: ISMIR (2003)
8. McKay, C., Fujinaga, I.: Automatic genre classification using large high-level musical feature sets. In: Proceedings of the International Conference on Music Information Retrieval, pp. 525–530 (2004)
9. Thayer, R.E.: *The biopsychology arousal*. Oxford University Press, Oxford (1989)
10. Tzanetakis, G., Cook, P.: Musical Genre Classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing* 10(5) (2002)
11. Wang, M., Zhang, N., Zhu, H.: User-adaptive music emotion recognition. In: 7th International Conference on Signal Processing, ICSP (2004)
12. Wiczorkowska, A., Synak, P., Ras, Z.: Multi-label classification of emotions in music. In: *Intelligent Information Processing and Web Mining, Advances in Soft Computing, Proceedings of IIS 2006 Symposium*, Ustron, Poland, vol. 35, pp. 307–315. Springer, Heidelberg (2006)
13. Witten, I.H., Frank, E.: *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco (2005)
14. Yang, Y., Su, Y., Lin, Y., Chen, H.: Music Emotion Recognition: The Role of Individuality. In: Proceedings of the international workshop on Human-centered multimedia ACM MM/HCM 2007 (2007)

Mining Musical Patterns: Identification of Transposed Motives

Fernando Berzal, Waldo Fajardo, Aída Jiménez, and Miguel Molina-Solana

Dept. Computer Science and Artificial Intelligence,
ETSIIT - University of Granada, 18071, Granada, Spain
fberzal@decsai.ugr.es, aragorn@ugr.es, aidajm@decsai.ugr.es,
miguelmolina@ugr.es

Abstract. Automatic extraction of frequent repeated patterns in music material is an interesting problem. This paper presents an effective approach of unsupervised frequent pattern discovery method from symbolic music sources. Patterns are discovered even if they are transposed. Experiments on some songs suggest that our approach is promising, specially when dealing with songs that include non-exact repetitions.

1 Introduction

When listening to music, many people realize that music has some repeating patterns. This is true for many types of music. In fact, a musical surface can be seen as a string of musical entities such as notes or chords on which pattern recognition techniques can be applied.

We can define a music motive as *the smallest meaningful melody element*. As a rule, motives are a group of notes no longer than one measure. Comparing with human speech, a motive is a word. In the same way that sentences consist of words, motives form musical phrases. Melody is formed by several main motives, which are repeated, developed and opposed one against another within the melody evolution.

Motive extraction in a piece of music is a basic task when analysing a music work. In those situations, musicians carry out a deep analysis of the musical material. Their studies include contextual information (such as the author, the aim or the period) but also morphological data from the music itself. One of the very first things a musician does when faced with a music sheet is looking for the motives that build the whole work.

Audio-thumbnailing (i.e., summarizing or abstracting) is another interesting application of information retrieval in the musical domain that is related with motive extraction. It provides the user with a brief excerpt of a song that, ideally, contains the main characteristics of the work. Before hearing or purchasing a whole song, it would be useful to hear a representative thumbnail of the whole work. This technique is also important for indexing large datasets of songs, which

can be browsed more quickly and searched more efficiently if indexed by those small patterns instead of by the whole song.

There have been some applications of pattern processing algorithms on musical strings. A complete overview can be found in the paper by Cambouropoulos *et al.* [1]. There are many different meaningful ways of representing a piece of music as a string, but all of them use either event strings (each symbol represents an event) or interval strings (each symbol represents the transformation between events).

Most of these applications work from a symbolic transcription of music. For example, Hsu *et al.* [2] used a dynamic programming technique to find repeating factors in strings representing monophonic melodies; whereas Rolland [3] recursively computed the distances between large patterns from the distances between smaller patterns.

Recently, some researchers have addressed the problem of pattern induction in an acoustic signal. For instance, Aucouturier and Sandler [4] propose an algorithm to find repeated patterns in an acoustic signal by focusing on timbre; whereas Chu and Logan [5] propose a method to find the most representative pattern in a song.

Meredith *et al.* [6] also proposed a geometric approach to repetition discovery in which the music is represented as a multidimensional dataset.

The paper by Grachten *et al.* [7] is of particular relevance because it represents melodies at a higher level than the notes but lower enough to capture the essence of the melody. This level is the ‘Narmour patterns’ level (based on Narmour’s I/R model), which is well-known in musicology.

In the approach we present in this paper, we are able to find frequent melodic and rhythmical patterns in the music starting from a MusicXML representation of the song. We first transform this symbolic representation into a sequence of notes. These notes are defined at their lowest level (i.e., pitch and duration) and in an absolute way, not relative. We have decided to work with a symbolic representation versus an audio one because it is closer to the original sheet of music. We do not take into account any differences in the relative importance between notes.

We have developed an Apriori-based algorithm to discover frequent subsequences in music files. Our algorithm is able to identify sequences even if they are transposed. Finally we post-process the output to return a set of representative subsequences (i.e., the song motives).

Our paper is organized as follows. We introduce some standard terms and review some sequence mining algorithms proposed in the literature in Section 2. Section 3 formally defines our sequence pattern mining problem and describes the algorithm we have devised to solve it. In Section 4, we explain the way our algorithm works by means of a particular example. We present some experimental results in Section 5 and we end our paper with some conclusions in Section 6.

2 Background

In our approach for musical motive extraction, we firstly transform a song into a sequence of notes. There is a rich variety of sequence types, ranging from simple sequences of letters to complex sequences of relations.

A *sequence* over an element type τ is an ordered list $S = s_1 \dots s_m$, where:

- each s_i (which can also be written as $S[i]$) is a member of τ , and is called an element of S ;
- m is referred to as the length of S and is denoted by $|S|$;
- each number between 1 and $|S|$ is a position in S .

$T = t_1 \dots t_n$ is called a *subsequence* of the sequence $S = s_1 \dots s_m$ if there exist integers $1 < j_1 < j_2 < \dots < j_n < m$ such that $t_1 = s_{j_1}$, $t_2 = s_{j_2}$, and in general, $t_n = s_{j_n}$.

Sequences have been used to solve different problems in the literature [8] [9]:

- *Sequential pattern mining*: Sequential patterns have been used for predicting the behaviour of individual customers. Each customer is typically modelled by a sequence of transactions containing the set of items he has bought. Several algorithms address this kind of problems, the most common being GSP [10] and PrefixSpan [11].
- *Sequence classification and clustering*: SVMs and Artificial Neuron techniques have been employed to classify sequences. Hierarchical and graph-based clustering algorithms have also been developed.
- *Periodic Patterns*: A traditional periodic pattern consists of a tuple of k components, each of which is either a literal or “*”, where k is the period of the pattern and “*” can be substituted for any literal and is used to enable the representation of partial periodicity. [12] [13]
- *Sequence Motifs*: A motif is essentially a short distinctive sequence pattern shared by a number of related sequences. There are four problems in this area: motif representation (i.e., designing the proper representation of the motif for the different applications), motif finding (i.e., finding a motif shared by several sequences), sequence scoring (i.e., computing the probability of a sequence to be generated by a motif), and sequence explanation (i.e., given a sequence and a motif with hidden states, provide the most likely state path that produced that sequence) [8].

Our problem of motive extraction in a piece of music can be included within the sequence motif applications. In particular, it can be expressed as a motif finding problem. This problem has been addressed by using Position Weight Matrix [14]. In a Position Weight Matrix each row corresponds to a letter in the alphabet and each column corresponds to a position in the window of the sequence. Using this matrix, we can represent the similarity between the sequences when they are aligned. Markov Chain Models have also been used as an extension of this technique [9].

In this paper, we propose an Apriori-based algorithm of finding motives in one (or several) sequences in order to solve the problem of motive extraction in a piece of music.

3 Our Sequence Pattern Mining Algorithm

The goal of frequent sequence pattern mining is the discovery of all the frequent subsequences in a large database of sequences D , or in a unique large sequence.

Let $\delta_T(S)$ be the occurrence count of a subsequence S in a sequence T and d_T a variable such that $d_T(S)=0$ if $\delta_T(S) = 0$ and $d_T(S)=1$ if $\delta_T(S) > 0$. We define the *support* of a subsequence as $\sigma(S) = \sum_{T \in D} d_T(S)$, i.e., the number of sequences in D that include at least one occurrence of the subsequence S . Analogously, the *weighted support* of a subsequence is defined as $\sigma_w(S) = \sum_{T \in D} \delta_T(S)$, i.e., the total number of occurrences of S within all the sequences in D . We say that a subsequence S is *frequent* if its support is greater than or equal to a predefined minimum support threshold. We define L_k as the set of all frequent k -subsequences (i.e., subsequences of size k).

If we consider the occurrences of a pattern that match approximately (i.e., those occurrences that are very similar but are not exactly the same), we define the *exact support* of a subsequence as the number of occurrences that are exactly equal to the pattern (while the support or weighted support will count both equal and similar occurrences).

Our algorithm is based on the POTMiner [15] frequent tree pattern mining algorithm, an Apriori-like algorithm for discovering frequent patterns in trees [16]. Therefore, it follows the Apriori iterative pattern mining strategy, where each iteration is broken up into two distinct phases:

- *Candidate Generation*: A candidate is a potentially frequent subsequence. In Apriori-like algorithms, candidates are generated from the frequent patterns discovered in the previous iteration. Most Apriori-like algorithms, including ours, generate candidates of size $k + 1$ by merging two patterns of size k having $k - 1$ elements in common.
- *Support Counting*: Given the set of potentially frequent candidates, this phase consists of determining their actual support and keeping only those candidates whose support is above the predefined minimum support threshold (i.e., those candidates that are actually frequent).

The pseudocode of our algorithm is shown in Figure 1. The details of the candidate generation and support counting phases will be described in the following sections.

algorithm

```

Obtain frequent elements (frequent patterns of size 1)
Build candidate classes  $C_1$  from the frequent elements
for  $k=2$  to MaxSize
  for each class  $P \in C_{k-1}$ 
    for each element  $p \in P$ .
      Compute the frequency of  $p$ 
      if  $p$  is frequent
        then
          Create a new class  $P'$  from  $p$ .
          Add  $P'$  to  $C_k$ 

```

Fig. 1. Our sequence mining algorithm

3.1 Candidate Generation

We use an equivalence class-based extension method to generate candidates [17]. This method generates $(k + 1)$ -subsequence candidates by joining two frequent k -subsequences with $k - 1$ elements in common.

Two k -subsequences are in the same equivalence class $[P]$ if they share the same prefix string until their $(k - 1)$ th element. Each element of the class can then be represented x , where x is the k -th element label.

Elements in the same equivalence class are joined to generate new candidates. This join procedure, also called extension, is described in the following paragraphs.

Let (x) and (y) denote two elements in the same class $[P]$, and $[P_x]$ be the set of candidate sequences derived from the sequence that is obtained by adding the element (x) to P . The join procedure results in attaching the element (y) to the sequence generated by adding the element (x) to P , i.e., $(y) \in [P_x]$. Likewise, $(x) \in [P_y]$.

3.2 Occurrence Lists

Once we have generated the potentially frequent candidates, it is necessary to determine which ones are actually frequent.

The support counting phase in our algorithm follows the strategy of AprioriTID [16]. Instead of checking the presence of each candidate in the sequence, $O(|S|)$, special lists are used to preserve the occurrences of each pattern in the database, thus facilitating the support counting phase.

Each occurrence list contains tuples (t, m, p, d, Θ) where t is the sequence identifier, m stores the elements of the sequence which match those of the $(k-1)$ prefix of the pattern X , p is the position of the last element in the pattern X , d is a position-based parameter used for guaranteeing that elements in the pattern are contiguous within the sequence and Θ indicates the similarity between the occurrence and the original pattern.

When building the scope lists for patterns of size 1, m is empty and the element d is initialized with the position of the pattern only element in the original database sequence.

We obtain the occurrence list for a new candidate of size k by joining the lists of the two subsequences of size $k - 1$ that were involved in the generation of the candidate. Let $(t_x, m_x, p_x, d_x, \Theta_x)$ and $(t_y, m_y, p_y, d_y, \Theta_y)$ be those lists, the join operation proceeds as follows:

if

1. $t_x = t_y = t$ **and**
2. $m_x = m_y = m$ **and**
3. $d_x = 1$ (only if $k \neq 2$) **and**
4. $p_x < p_y$ **and**
5. $\Theta_x = \Theta_y$

then add $[t, m \cup \{p_x\}, p_y, d_y - d_x, \Theta_y]$ to the occurrence list of the generated candidate.

3.3 Support Counting

Checking if a pattern is frequent consists of counting the elements in its occurrence list. The counting procedure is different depending on whether the weighted support σ_w is considered or not.

- If we count occurrences using the weighted support, all the tuples in the lists must be taken into account.
- If we are not using the weighted support, the support of a pattern is the number of different sequence identifiers within the tuples in the list of the pattern.

It should be noted that d represents the distance between the last node in the pattern and its prefix m . Therefore, we also have to consider only the elements in the scope lists whose d parameter equals 1 for guaranteeing that elements in the pattern are contiguous within the sequence. It should be also noted that remaining elements in the lists can not be eliminated because they may be useful when building the occurrence lists of larger patterns.

3.4 Closed Subsequences

The amount of patterns returned by our algorithm is probably too large to be easily understood by the user, so we have included a post-processing stage in order to reduce the number of patterns. This stage deletes all the patterns that are included within larger ones. So that, our algorithm only returns frequent closed subsequences. A subsequence c is closed if there is no subsequence s such as $c \subset s$, having c the same support as s .

4 Example

In this section, we present an example to help the reader understand the way our algorithm identifies frequent subsequences in a sequence. In order to facilitate the understanding of the procedure, we only take into account those transpositions of fifth.

We will use in this paper the scientific pitch notation which combines a letter-name, accidentals (if any) and a number identifying the pitch's octave. This notation is the most common in English written texts.

Let suppose we have the following piece of a song: G4 A4 G4 E4 D5 E5 D5 B4 G4 A4 G4 E4 A4 G4 B4 G4 A4 G4 E4, and we want to extract those subsequences that appear at least four times in it.

The first step of our algorithm is to scanning the sequence to obtain all the occurrences of each note. Then, the occurrence lists of each note are built as shown in Figure 2.

The first element is 1 in all the tuples because we only have one sequence (i.e., only one song) in our example. The second one is the prefix of the substring (empty in patterns of size 1). The third element indicates the position of the last

G4	A4	E4	D5	E5	B4
{1,_,1,1,=}	{1,_,2,2,=}	{1,_,4,4,=}	{1,_,5,5,=}	{1,_,6,6,=}	{1,_,8,8,=}
{1,_,3,3,=}	{1,_,10,10,=}	{1,_,12,12,=}	{1,_,7,7,=}		
{1,_,9,9,=}	{1,_,13,13,=}	{1,_,10,19,=}			
{1,_,11,11,=}	{1,_,17,17,=}	{1,_,8,8,+5}			
{1,_,14,14,=}	{1,_,6,6,+5}				
{1,_,16,16,=}					
{1,_,18,18,=}					
{1,_,5,5,+5}					
{1,_,7,7,+5}					

Fig. 2. Occurrence lists of the elements of the following sequence: G4 A4 G4 E4 D5 E5 D5 B4 G4 A4 G4 E4 A4 G4 B4 G4 A4 G4 E4

element of the pattern in the sequence, while the fourth is the distance between the last element of the pattern and its prefix. Finally, the last element indicates if the occurrence is exactly equal to the pattern (“=”) or if it is transposed up one fifth (“+5”). It should be noted that the relation between two notes is commutative, so that, we only look for related patterns in one direction. We will always find at least a version of the pattern that summarizes all its transpositions. For example, the note G4 is transposed up one fifth as D5, therefore there are 9 tuples in the list of occurrences of G4: 7 as itself, 2 as D5.

The next step is checking if all the elements are frequent. In this case, only G4 A4 and E4 have at least four occurrences. Therefore only these patterns will be kept.

Figure 3 shows the extension of the element G4. This element is extended with all the frequent patterns of size 1 including itself, and the occurrence lists of the candidate patterns of size 2 are obtained by joining the lists of the elements that generated it, as explained in section 3.2.

Prefix: G4					
G4 G4		G4 A4		G4 E4	
{1,1,3,2,=}	{1,1,9,8,=}	{1,1,2,1,=}	{1,1,10,9,=}	{1,1,4,3,=}	{1,1,12,11,=}
{1,1,11,10,=}	{1,1,14,13,=}	{1,1,13,12,=}	{1,1,17,15,=}	{1,1,19,18,=}	{1,3,4,1,=}
{1,1,16,15,=}	{1,1,18,17,=}	{1,3,10,7,=}	{1,3,13,10,=}	{1,3,12,9,=}	{1,3,19,9,=}
{1,3,9,6,=}	{1,3,11,8,=}	{1,3,17,14,=}	{1,9,10,1,=}	{1,9,12,3,=}	{1,9,19,10,=}
{1,3,14,11,=}	{1,3,16,13,=}	{1,9,13,4,=}	{1,9,17,8,=}	{1,11,12,1,=}	{1,11,19,8,=}
{1,3,18,15,=}	{1,9,11,2,=}	{1,11,13,2,=}	{1,11,17,6,=}	{1,14,19,5,=}	{1,16,19,3,=}
{1,9,14,5,=}	{1,9,16,7,=}	{1,14,17,3,=}	{1,16,17,1,=}	{1,18,19,1,=}	{1,5,8,3,+5}
{1,9,18,9,=}	{1,11,14,3,=}	{1,5,6,1,+5}		{1,7,8,1,+5}	
{1,11,16,5,=}	{1,11,18,7,=}				
{1,14,16,2,=}	{1,14,18,4,=}				
{1,16,18,2,=}	{1,5,7,2,+5}				

Fig. 3. Extension of the element G4 in Figure 2

Figure 3 shows with bold letters the tuples where $d = 1$. That means that these are contiguous occurrences of the pattern. In our example, only the patterns G4 A4 and G4 E4 appears as contiguous subsequences in our song. Furthermore,

they have at least four occurrences (our minimum support threshold) and they will be extended to generate candidates of size 3. The pattern G4 G4 although is not contiguous and will not be extended, is preserved to perform the extension of G4 A4 and G4 E4.

After two more extensions we will obtain the pattern: G4 A4 G4 E4 with $support = 4$ and $exactsupport = 3$.

5 Experiments

We have tested our algorithm using extracts of three well-known songs: *Fur Elise* (268 notes) by L.v. Beethoven, *Marcha alla turca* (485 notes) by W.A. Mozart, and *Ballade pour Adeline* (453) by R. Clayderman. Unlike the former example, in these experiments all the possible transpositions are taken into account.

Fig. 4 summarizes the set-up and the results of our experiments. Each row correspond to one of the songs.

First column indicates the number of notes in that song. Support column indicates the minimum number of repetitions a pattern should have to be considered as frequent. It should be noted that these repetitions do not need to be exact, as they include all possible transpositions. Regarding the length of the excerpts been tested, a minimum support of 6 seems adequate.

The third column indicates the maximum length of patterns. As said before, musical motives are generally no longer than a measure. So that, we have included this restriction in our algorithm. The concrete values for this input have been set manually but they could have been chosen dynamically by some kind of optimization procedure.

Regarding the second part of the table, firstly we can find the number of patterns (no matter the size) that the algorithm returns for each song. As can be seen, the amount of patterns is pretty high (more than two times the number of notes). In fact, many of these patterns are not musically relevant since they are part of bigger ones. Even more, this set also includes those patterns of size 1 and 2 that can hardly be considered as motives.

Fifth column (filter patterns) shows the number of representative motives, those that we have formerly called ‘closed subsequences’ (see Section 3.4). The minimum length of a closed subsequence is also set to three.

Finally, we show the number of closed patterns that would not have been obtained without considering transpositions. As can be seen from the Figure, in

	Input			Output		
	Notes	Support	MaxSize	All Patterns	Filter Patterns	non-exact patterns
Fur Elise	268	6	15	498	10	4
Marcha Alla Turca	485	6	11	1635	24	17
Ballade pour Adeline	453	6	10	1467	42	14

Fig. 4. Results from experiments with *Fur Elise*, *Marcha alla turca* and *Ballade pour Adeline*

Marcha alla turca only four patterns would reach the minimum support just by exact repetitions, whereas there are thirteen patterns that have the same interval structure but different pitch (i.e., they are transposed patterns).

6 Conclusions

We have presented an application of frequent pattern mining to the discovery of musical motives in a piece of music. We obtain the sequence of notes of a song from its Music XML file, which can be easily collected.

We have developed an Apriori-based algorithm which is able to identify frequent subsequences in a sequence where the matching between the patterns might not be exact but similar in the way the user defines. Using this algorithm we can identify motives even if they are transported. We have shown that our approach is able to perform well in a small set of manually annotated examples.

In this paper we have used an absolute approach to represent notes. In the future, we plan to employ interval strings rather than absolute pitches. We would also consider more abstract levels of representing melodies, as the one proposed by Narmour. Furthermore, we plan to use some datasets already labelled in order to compare our algorithm to other existing approaches.

Acknowledgements

F. Berzal and A. Jiménez are supported by the Spanish Ministry of Education and Science under the project TIN2006-07262, whereas W. Fajardo and M. Molina-Solana are supported by the research project TIN2006-15041-C04-01.

References

1. Cambouropoulos, E., Crawford, T., Iliopoulos, C.S.: Pattern processing in melodic sequences: Challenges, caveats and prospects. *Computers and the Humanities* 35(1), 9–21 (2001)
2. Hsu, J.-L., Liu, C.-C., Chen, A.L.: Efficient repeating pattern finding in music databases. In: *Proc. ACM 7th Int. Conf. on Information and Knowledge Management*, pp. 281–288 (1998)
3. Rolland, P.-Y.: Discovering patterns in musical sequences. *Journal of New Music Research* 28(4), 334–350 (1998)
4. Aucouturier, J.-J., Sandler, M.: Finding repeating patterns in acoustic musical signals: Applications for audio thumbnailing. In: *Audio Engineering 22nd Int. Conf. on Virtual, Synthetic and Entertainment Audio (AES22)*, pp. 412–421 (2002)
5. Chu, S., Logan, B.: Music summary using key phrases. In: *IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP 2000* (2002)
6. Meredith, D., Lemström, K., Wiggins, G.A.: Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research* 31(4), 321–345 (2002)
7. Grachten, M., Arcos, J.L., de Mantaras, R.L.: Melodic similarity: Looking for a good abstraction level, pp. 210–215 (2004)

8. Dong, G., Pei, J.: *Sequence Data Mining (Advances in Database Systems)*. Springer, New York (2007)
9. Han, J., Kamber, M.: *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco (2005)
10. Srikant, R., Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) *EDBT 1996. LNCS*, vol. 1057, pp. 3–17. Springer, Heidelberg (1996)
11. Pei, J., Han, J., Mortazavi-asl, B., Pinto, H., Chen, Q., Dayal, U., chun Hsu, M.: Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: *International Conference on Data Engineering*, pp. 215–224 (2001)
12. Wang, W., Yang, J., Yu, P.S.: Meta-patterns: revealing hidden periodic patterns. *IBM Research Report*, pp. 550–557 (2001)
13. Yang, J., Wang, W., Yu, P.S.: Infominer: mining surprising periodic patterns. In: *Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 395–400. ACM, New York (2001)
14. Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.F., Wootton, J.C.: Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *Science* 262(5131), 208–214 (1993)
15. Jimenez, A., Berzal, F., Cubero, J.C.: Mining induced and embedded subtrees in ordered, unordered, and partially-ordered trees. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) *Foundations of Intelligent Systems. LNCS (LNAI)*, vol. 4994, pp. 111–120. Springer, Heidelberg (2008)
16. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: *20th International Conference on Very Large Data Bases*, pp. 487–499 (1994)
17. Zaki, M.J.: Efficiently mining frequent embedded unordered trees. *Fundamenta Informaticae* 66(1-2), 33–52 (2005)

Musical Instruments in Random Forest

Miron Kursa¹, Witold Rudnicki¹, Alicja Wieczorkowska²,
Elżbieta Kubera³, and Agnieszka Kubik-Komar³

¹ Interdisciplinary Centre for Mathematical and Computational Modelling (ICM),
University of Warsaw, Pawinskiego 5A, 02-106 Warsaw, Poland

² Polish-Japanese Institute of Information Technology,
Koszykowa 86, 02-008 Warsaw, Poland
alicja@pjwstk.edu.pl

³ University of Life Sciences in Lublin, Akademicka 13, 20-950 Lublin, Poland

Abstract. This paper describes automatic classification of predominant musical instrument in sound mixes, using random forests as classifiers. The description of sound parameterization applied and methodology of random forest classification are given in the paper. Additionally, the significance of sound parameters used as conditional attributes is investigated. The results show that almost all sound attributes are informative, and random forest technique yields much higher classification results than support vector machines, used in previous research on these data.

1 Introduction

In the era of vast and continuously growing audiovisual data available in private repositories and in the Internet, it becomes desirable to be able to automatically browse these data in order to find the specified contents. The user may be interested in finding pieces of music in the desired mood or style, finding tunes, or timbres. This research addresses the problem of automatic identification of timbre, i.e. musical instrument, in audio data. This can be performed using various classifiers, run on digital data. In this paper, we use exemplary feature vector to parameterize sound for timbre classification purposes, and we present random forest technique applied as a classifier. Since the feature vector is arbitrarily chosen, we decided to check importance of these parameters, even though they were already used and reported fairly successful in similar research [23].

1.1 Recognition of Musical Instruments

Musical instrument sound recognition has been investigated last years by various research centers, starting from recognition of isolated sounds, and last years also sounds in mixes. There is no standard set of parameters used, although MPEG-7 features reflect parameterization used in audio research and offer numerous features [11]. The classifiers applied in research on musical instruments include k-nearest neighbors, artificial neural networks, rough set based algorithms, support vector machines (SVM), etc. [5,8,12,16,24]. Parameterization used includes

features describing properties of DFT spectrum, wavelet analysis coefficients, MFCC (Mel-Frequency Cepstral Coefficients), multidimensional scaling analysis trajectories, etc. [10]. The results are difficult to compare because of various numbers of classes used, no. of objects per class, and audio data. Generally, recognition of instruments for isolated sounds can reach 100% for small number of classes, more than 90% if instrument or articulation family is identified, and the accuracy goes down to about 70% or less for recognition of instrument when there are more classes to recognize. Research on identification of instruments in mixes was also performed (same-pitch multi-timbre mixes being the most difficult) [7,9,13,23,26], and the results vary depending on the instruments and sounds chosen. The outcomes can be used for aiding automatic music transcription, but this usually aims at multiple-pitch tracking, and instrument information is only supplementary (for separation of particular voices).

Here we decided to use random forest technique, as it is promising for high-dimensional feature set data, which is often the case in audio signal classification.

Random Forest. Random Forest (RF) is a classifier which comprises of a set of weak, weakly correlated and non-biased classifiers, namely the decision trees. It has been shown that in many cases RF performs equally well or better than other methods on a diverse set of problems [3]. It has been widely used in classification problems as diverse as bioinformatics [2,6,15], medicine [22] or, more recently, material science [4], transportation safety [1], or customer behavior [25].

In addition, RF offers a useful feature that improves our understanding of a classification problem under scrutiny, namely it gives estimate of the importance of attributes for the final prediction. It is often used for analysis when both classifier and identification of important variables are goals of the study [2,15].

2 Material and Methods

Our data originate from MUMS [17], widely used in similar research. They represent instrument sounds, in many cases played with various methods (articulation). We chose 12 sounds - octave no.4 in MIDI – for the following 14 instruments: clarinet; flute; oboe; English horn; trumpet; French horn; tenor trombone; piano; marimba; vibraphone; tubular bells; violin, viola, and cello *vibrato*.

Our goal was to recognize the instrument dominating in same-pitch mix, as this is the most challenging task, since in this case partials in spectra overlap and separation of sounds is more difficult. The training data describe isolated monophonic instrumental sounds, and the same sounds mixed with the second, artificial sound: triangular wave of the same pitch, saw-tooth wave, white noise, or pink noise. To make sure we recognize the predominant sound, the level of added sound was only a percentage of the main sound level, at 7 versions in equal step in log scale: 50%, $50/\sqrt{2}\%$, 25%, $25/\sqrt{2}\%$, 12.5%, $12.5/\sqrt{2}\%$, 6.25%. The test data represent mixes of the predominant instrument with the remaining instrument sounds of the same pitch from our data set, at the same 7 levels. We also performed experiments on combined training set and combined test set [23].

2.1 Construction of Attributes

The parametrization used here was already applied in similar research [23,26] and the results were promising, so we decided to follow this scheme. The feature set consists of 219 parameters, based mainly on MPEG-7 audio descriptors, and other features used in similar research. Most of the parameters in this feature vector represent average value of frame-based attributes, calculated for consecutive frames of an investigated sound (or mix) using sliding analysis window, moved through the entire file. The calculations were performed for the left channel of digital data for 44.1 kHz sampling rate and 16-bit resolution, using 120 ms analyzing frame with Hamming window (hop size 40 ms), which allows analysis of the low-pitched sounds even for the lowest audible fundamental frequencies, if one would like to investigate full music scale. The feature vector consists of the following parameters, describing sound features in time domain, time-frequency domain, and frequency domain (averaged over all frames) [23,26]:

- MPEG-7 based descriptors [11]: *AudioSpectrumSpread*; *AudioSpectrumFlatness* for 25 out of 32 frequency bands; *AudioSpectrumCentroid*; *AudioSpectrumBasis* features: 165 features for 33 sub-spaces - min, max, mean, distance (summation of dissimilarity, i.e. absolute difference of values, of every pair of coordinates in the vector), and standard deviation of *AudioSpectrumBasis*; *HarmonicSpectralCentroid*, *HarmonicSpectralSpread*, *HarmonicSpectralVariation*, *HarmonicSpectralDeviation*, *LogAttackTime*, *TemporalCentroid*.
- other: *Energy*; *MFCC* (min, max, mean, distance, standard deviation); *ZeroCrossingRate*; *RollOff*; *Flux*; *FundamentalFrequency*; r_1, \dots, r_{11} - various ratios of harmonic partials: r_1 - energy of the fundamental to the total energy of all harmonics, r_2 : amplitude difference [dB] between 1st and 2nd partial, r_3 : ratio of the sum of partials 3-4 to all harmonics, r_4 : partials 5-7 to all, r_5 : partials 8-10 to all, r_6 : remaining partials to all, r_7 : brightness - gravity center of spectrum, r_8, r_9 : contents of even/odd harmonics in spectrum.

2.2 Random Forest Method

RF is an ensemble of classification trees, constructed using procedure which minimizes bias and correlations between individual trees. Each tree is built using different bootstrap sample of the training set. The elements of the sample are drawn with replacement from the original set; roughly 1/3 of the training data are not used in the bootstrap sample for any given tree. For each tree in RF these elements are called out-of-bag (OOB) elements for the tree (they are different for each tree). Let us assume that objects are described by a vector of P attributes. At each stage of tree building, i.e. for each node of any particular tree in RF, p attributes out of all P attributes are randomly selected, where $p \ll P$ (often $p = \sqrt{P}$). The best split on these p attributes is used to split the data in the node. Each tree is grown to the largest extent possible (no pruning).

By repeating this randomized procedure M times one obtains a collection of M trees, hence a random forest. Classification of each object is made by simple

voting of all trees. The number of trees depends on the problem, usually the number of steps is selected to assure that the classification error is not changed after adding more trees. The classification error on the training set is estimated by counting only votes put by trees on their OOB objects.

This estimate of attributes' importance is performed in the following way. For each attribute, one takes all trees, which were using this attribute. Then each tree classifies all its OOB objects. One counts number of correct decisions. Then one permutes values of the attribute between all objects and repeats the procedure. The average difference between number of correct classifications in these two cases is a raw classification score. One can also compute the variance and standard deviation and use it to compute Z-score of the raw score.

2.3 Feature Selection

The validity of the estimate of the variable importance is based on the assumption that the individual trees building the random forest are uncorrelated, which, in most cases is fulfilled only approximately [3]. Also, when the number of variables is large, it is difficult to discern truly important variables from these which gain importance due to random correlations in data. It has been shown by simulations [19,20,21] and observation of experimental data [18] that importance measure for most attributes can be highly variable.

To solve this problem, we developed an algorithm comparing the apparent importance of the original variables with that of the randomized ones [14,18]. This algorithm is a wrapper based on the importance score obtained from the RF method. In this method one creates many times an extended system, where each attribute has a mirror copy. Values of the mirror attributes are randomly permuted between objects, and importance of the original attributes compared with these of the mirror ones. Only attributes of importance consistently higher than the highest importance for the randomized attribute are considered important.

This algorithm finds all attributes, which, for given data set, are correlated more strongly with the decision attribute than random attributes. There is no guarantee that all attributes which are truly related with the decision are found or that the attributes which are found are truly, and not by chance, correlated with the decision attribute. Still, our algorithm is an attempt in this direction and it gives reasonable estimate of the importance of the selected attributes.

3 Results and Discussion

Machine learning methods have been already used for identification of musical instruments; in the previous paper [23] we have shown that SVM yielded results ranging from 55.9% (for learning on single instrument sounds and testing on mixes with added instrumental sounds of 50% level of the main sound level) up to 89.88% correctness (for learning on single instrument sounds with added artificial sounds of 6.25% level and testing on mixes for the same level).

Results of classification for RF trained on pure instrument sounds are shown in Figure 1 (left panel). Horizontal axis represents quantity of added sound [%],

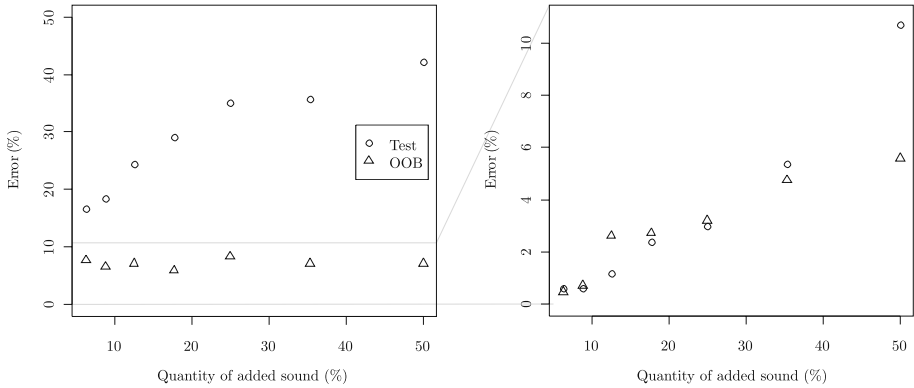


Fig. 1. Classification results using RF for learning on sounds representing only a single instrument (left panel) and learning on sounds containing varying level of additional sounds (right panel); the level is shown as percentage of the main sound level. OOB results are marked with circles, and the results for test sets (mixes of instrument sounds) are marked with triangles. Remark: different scales are used in the left and right panels.

and vertical axis represents recognition error for dominating sound. Learning in the presence of multiple sounds is shown in Figure 1 (right panel). In this case the OOB error and test error of the RF classifier are both small (few percent) and agree very well with each other. As long as the added sound level is small, not much increase of error is observed in tests, so for the low added levels the OOB error estimate agrees well with the test set level. Only in the case where the added sound level is 50% the OOB error estimate and test set estimate diverge. OOB remains relatively small (around 5%), while test set error increases to 10%.

RF OOB error of the classifier for pure sounds shows significant improvement over SVM, but no improvement if this classifier is used to recognize mixes (Fig 2).

For all but one levels of added sound, error for RF classifier is much lower than that of SVM, almost by an order of magnitude. This is still about three times better than in the case of SVM classifier, but this increase of error shows the limits of validity for the applied training and testing procedure. It shows that training on noisy data which is not related to the noise added in the test set works well until the level of added noise is smaller than level of pure sound.

Comparing results of the training performed on the samples of pure sound with those obtained for sound mixes, the OOB error in the latter case is sometimes more than an order of magnitude smaller than in the former one. It is clear that adding noise to the training set dramatically improves results of training.

Following the procedure from earlier work, we have also tested the case when training set consisted of all training samples and similarly, the testing set consisted of all testing samples as well. In this case RF classifier worked very well, the OOB error was marginal (0.05%) and test set error was around 1%. Again, these results were an order of magnitude better than that for SVM classifier.

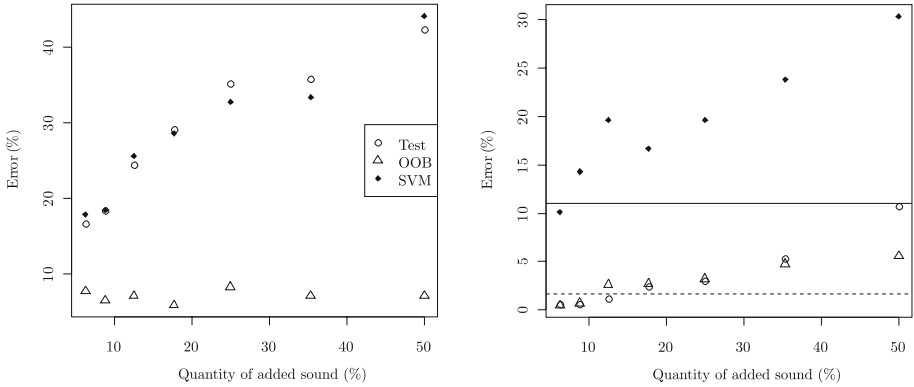


Fig. 2. Classification results using RF and SVM, for learning on single sounds (left) and on sounds containing varying level of additional sounds (right)

Results of the classification show that RF method is very well suited to classification of musical instruments, significantly better than SVM. The difference is of qualitative nature, since in most cases the RF error is one order of magnitude better than SVM. This is mainly due to characteristics of features used for the description of musical instrument sounds: feature values for each instrument are discrete points distributed over wide intervals creating characteristic patterns; intervals corresponding to various instruments overlap significantly (Fig. 3).

As an example all values of two attributes (*TemporalCentroid* and *basis101*) are displayed for the samples taken from clear sound of instruments and for all samples. One can see that adding different sound to the sample of clear instruments does not change the characteristic patterns but moves it slightly, so when one overlays all mixed samples the wider blurred bands are created.

Data of that type fits very well to the tree-based methods, where each leaf can represent separate small interval in 1-dimensional space. On the other hand it poses a challenge for the classifier based on finding continuous intervals. Such method can only succeed using very highly dimensional property space, where one can map all necessary splits on complex multidimensional figures.

Indeed, one can create relatively good RF classifiers using only a single attribute. We have constructed single feature RF classifiers using all features, at 6.25% added sound level. For most features identified as important by our feature selection procedure one can obtain RF classifier with OOB error close to 80%, which is noticeably better than random choice (the reference level of random classification is 92%). If one chooses features which are highly ranked by RF importance ranking algorithm, one can construct significantly better classifiers. For example the RF classifier built using only *TemporalCentroid*, ranked #1 in the importance ranking, has OOB error 14.5%, whereas the error on the test set is 25%. One can also use less important features and still get classifier which is noticeably better than random choice. For example for *basis5* (ranked #30), the errors of the RF classifier built using this feature are respectively 76.8% and

82.9%. In most cases these classifiers are too weak for any useful classification, but one should remember that in our case the test set consists of samples constructed by mixing different sounds than the original training set, nevertheless the classification error is still noticeably smaller than for random classifier.

Important attributes. Importance of the attributes for prediction was estimated using Boruta Algorithm, aiming at finding all truly informative features. This algorithm compares importance of all features with a reference importance, i.e. maximal apparent importance of the randomly permuted mirror features.

The results yielded by Boruta are different for systems with different levels of added sound. Generally, the number of important attributes grows with the level of added sound; for the combined data set (all levels), all attributes were informative. Still, for the 6.25% level, 146 out of 219 features were informative. There was no clear cut-off value of the average importance between important and non important features, and the importance of a given feature in different iterations

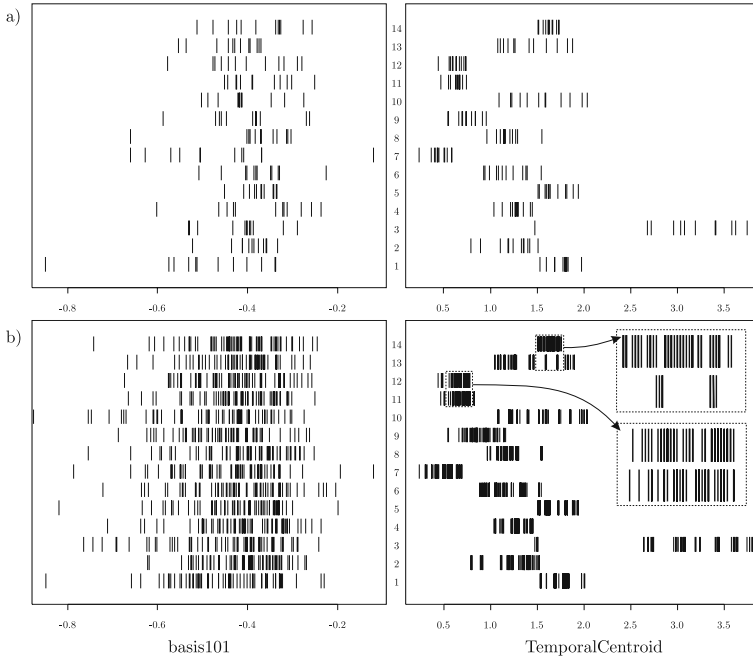


Fig. 3. Attribute values for one of unimportant descriptors (*basis101*) and the most important descriptor (*TemporalCentroid*) for sounds containing only a single instrument a), and for sound containing 50% of added (mixed) sounds b) for all instruments. The instruments are marked on vertical axis as follows: 1.clarinet, 2.cello, 3.trumpet, 4.English horn, 5.flute, 6.French horn, 7.marimba, 8.oboe, 9.piano, 10.trombone, 11.tubular bells, 12.vibraphone, 13.viola, 14.violin. The details for two pairs of instruments which are well discerned (viola and violin) and poorly discerned (bells and vibraphone) by classifier using single attribute are shown in inset in right panel for the case of mixes.

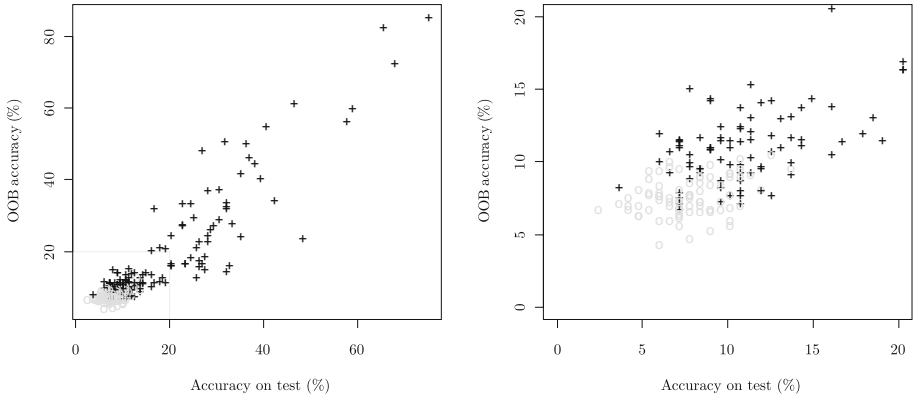


Fig. 4. Importance of attributes vs. the results of the classifiers built using a single attribute. Selection of important attributes was performed for the classifier trained for recognition of musical instrument sounds with mixes (at 6.25% level), using this attribute as the only conditional attribute. Important attributes are marked as “+”, non-important – as “o”. Position of the attribute on the plot corresponds to the accuracy of the classifier for the test set and OOB accuracy on the training set. The results for all attributes are shown in the left panel, the close-up of the is shown on the right.

of Boruta may vary very significantly; e.g., the maximal value registered for the feature which is unimportant by design was higher than average importance of 118 out of 146 important features. Still, the minimal value registered for the feature confirmed to be important was lower than average importance of 68 out of 73 unimportant features. It means that in any single RF run, an important feature can attain value lower than average value for most unimportant features. Still, the unimportant feature can reach the value that is higher than average importance measure for most of the important features. Therefore the result of the importance measure for a single RF run should not be considered reliable.

It is interesting to compare importance of the attributes with the error achieved by RF built on a single attribute (Fig 4). Feature selection algorithm finds all attributes that are alone sufficient for construction of a good classifier. Also, the algorithm can find in the cloud of weak attributes (of low predictive power) a set of features that together with other features yield a good classifier.

4 Summary and Conclusions

Results for classifiers trained on pure instrumental sounds are quite low both in case of RF and SVM. Adding mixed sounds to the training set significantly improves classification accuracy in both cases, but the improvement is much higher for RF. The classification results show spectacular superiority of RF over SVM, even though SVM is commonly considered to be a very good classifier – RF is an order of magnitude better than SVM in most cases. The advantage of

using RF in comparison with SVM is caused by sparse distribution of attribute values. They cannot be mapped on large continuous intervals – a large number of small intervals must be used for representation of the attributes. This structure fits very well trees, whereas the SVM may construct inconvenient representation. When additional sounds are mixed with the main one, single attribute values split into several distinct values occupying an interval. The intervals pertaining to different instruments often overlap, and discernment becomes more difficult. This increased difficulty is reflected in the size of the trees. The number of nodes depends on the level of added sounds; for single sounds, trees had 39-83 nodes, for the highest level of added sounds: 227-381 nodes, and for combined data 545-1089 nodes. The trees trained on pure sounds are too simple and cannot properly classify mixes, but the trees trained on mixes perform well also for simpler cases.

Our analysis of the importance shows that most of MPEG-7 based features may be used for the classification, and reasonably good results can be obtained with RF, using merely a single descriptive attribute for all trees in a given RF.

Acknowledgements. This project was partially supported by ICM grants 501-64-13-BST1345 and G34-5, and the Research Center of PJIIT, supported by the Polish National Committee for Scientific Research (KBN).

References

1. Abdel-Aty, M., Pande, A., Das, A., Knibbe, W.: Assessing Safety on Dutch Free-ways with Data from Infrastructure-Based Intelligent Transportation Systems. *Transp. Res. Rec.* 2083, 153–161 (2008)
2. Bureau, A., Dupuis, J., Falls, K., Lunetta, K., Hayward, B., Keith, T., Eerdewegh, P.: Identifying SNPs Predictive of Phenotype Using Random Forests. *Gen. Epi-dem.* 28 (2005)
3. Breiman, L.: Random Forests. *Machine Learning* 45, 5–32 (2001), http://www.stat.berkeley.edu/~breiman/RandomForests/cc_papers.htm
4. Carr, D.A., Lach-Hab, M., Yang, S.J., Vaisman, I.I., Blaisten-Barojas, E.: Machine learning approach for structure-based zeolite classification. *Micropor. Macropor. Mat.* 117, 339–349 (2009)
5. Cosi, P., De Poli, G., Lauzzana, G.: Auditory Modelling and Self-Organizing Neural Networks for Timbre Classification. *J. New Music Research* 23, 71–98 (1994)
6. Diaz-Uriarte, R., Alvarez de Andres, S.: Gene selection and classification of microarray data using random forest. *BMC Bioinformatics* 7, 3 (2006)
7. Dziubinski, M., Dalka, P., Kostek, B.: Estimation of musical sound separation algorithm effectiveness employing neural networks. *J. Intel. Inf. Syst.* 24(2-3), 133–157 (2005)
8. Fujinaga, I., McMillan, K.: Realtime recognition of orchestral instruments. In: *Proceedings of the International Computer Music Conference*, pp. 141–143 (2000)
9. Goto, M.: A real-time music-scene-description system: predominant-f0 estimation for detecting melody and bass lines in real-world audio signals. *ISCA* 43(4), 311–329 (2004)
10. Herrera, P., Amatriain, X., Batlle, E., Serra, X.: Towards instrument segmentation for music content description: a critical review of instrument classification techniques. In: *International Symposium on Music Information Retrieval ISMIR* (2000)

11. ISO: MPEG-7 Overview, <http://www.chiariglione.org/mpeg/>
12. Kaminskyj, I.: Multi-feature Musical Instrument Classifier. *MikroPolyphonie* 6 (2000)
13. Klapuri, A.: Signal processing methods for the automatic transcription of music. Ph.D. thesis, Tampere University of Technology, Finland (2004)
14. Kursa, M., Jankowski, A., Rudnicki, W.: Boruta – a system for feature selection. In: Nguyen, H.S., Huynh, V.N. (eds.) SCKT-08 Hanoi Vietnam (PRICAI 2008), pp. 122–133 (2009)
15. Lunetta, K.L., Hayward, L.B., Segal, J., Eerdewegh, P.V.: Screening Large-Scale Association Study Data: Exploiting Interactions Using Random Forests. *BMC Genetics* 5, 32 (2004)
16. Martin, K.D., Kim, Y.E.: 2pMU9. Musical instrument identification: A pattern-recognition approach. 136 meeting Acoustical Soc. America, Norfolk, VA (1998)
17. Opolko, F., Wapnick, J.: MUMS – McGill University Master Samples. CD's (1987)
18. Rudnicki, W., Kierczak, M., Koronacki, J., Komorowski, J.: A Statistical Method for Determining Importance of Variables in an Information System. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 557–566. Springer, Heidelberg (2006)
19. Strobl, C., Boulesteix, A., Zeileis, A., Hothorn, T.: Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics* 8, 25 (2007)
20. Strobl, C., Zeileis, A.: Danger: High Power! – Exploring the Statistical Properties of a Test for Random Forest Variable Importance. Tech. Rep.17. Univ. Munich (2008)
21. Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., Zeileis, A.: Conditional Variable Importance for Random Forests. Tech. Rep. 23. Dept. Stat., Univ. of Munich (2008)
22. Ward, M.M., Pajevic, S., Dreyfuss, J., Malley, J.D.: Short-Term Prediction of Mortality in Patients with Systemic Lupus Erythematosus: Classification of Outcomes Using Random Forests. *Arthritis and Rheumatism* 55, 74–80 (2006)
23. Wiczorkowska, A., Kubera, E., Kubik-Komar, A.: Analysis of Recognition of a Musical Instrument in Sound Mixes Using Support Vector Machines. In: Nguyen, H.S., Huynh, V.N. (eds.) SCKT 2008 Hanoi, Vietnam (PRICAI 2008), pp. 110–121 (2008)
24. Wiczorkowska, A.: Rough Sets as a Tool for Audio Signal Classification. In: Raś, Z.W., Skowron, A. (eds.) ISMIS 1999. LNCS (LNAI), vol. 1609. Springer, Heidelberg (1999)
25. Xie, Y.Y., Li, X., Ngai, E.W.T., Ying, W.Y.: Customer churn prediction using improved balanced random forests. *Expert Syst. Appl.* 36, 5445–5449 (2009)
26. Zhang, X.: Cooperative Music Retrieval Based on Automatic Indexing of Music by Instruments and Their Types. Ph.D thesis, Univ. North Carolina, Charlotte (2007)

Application of Analysis of Variance to Assessment of Influence of Sound Feature Groups on Discrimination between Musical Instruments

Alicja Wieczorkowska¹ and Agnieszka Kubik-Komar²

¹ Polish-Japanese Institute of Information Technology,
Koszykowa 86, 02-008 Warsaw, Poland
alicja@pjwstk.edu.pl

² University of Life Sciences in Lublin,
Akademicka 13, 20-950 Lublin, Poland
agnieszka.kubik@up.lublin.pl

Abstract. In this paper, the influence of the selected sound features on distinguishing between musical instruments is presented. The features were chosen basing on our previous research. Coherent groups of features were created on the basis of significant features, adding complementary ones according to the parameterization method applied, to constitute small, homogenous groups. Next, we investigate (for each feature group separately) if there exist significant differences between means of these features for the studied instruments. We apply multivariate analysis of variance along with post hoc analysis in the form of homogeneous groups, defined by mean values of the investigated features for our instruments. If a statistically significant difference is found, then the homogenous group is established. Such a group may consist of one instrument (distinguished by this feature), or more (instruments similar wrt. this feature). The results show which instruments can be best discerned by which features.

1 Introduction

Automatic classification of musical instruments in audio recording is an example of research on music information retrieval. Huge repositories of audio data available for the users are challenging from the point of view of content-based retrieval. Users can be interested in finding melodies sung to the microphone (queries by humming), identify the title and the performer of a piece of music submitted to as an audio input containing a short excerpt from the piece (query by example), or finding pieces played by their favorite instrument. Browsing audio files by users is a tedious tasks, therefore any automation comes very handy. If the audio data are labeled, searching is easy, but usually the text information added to the audio file is limited to the title, performer etc. In order to perform automatic content annotation, sound analysis is usually performed, sound features extracted, and then the contents can be classified into various categories, in order to fulfill the user's query to find the contents specified.

In our previous research [13], we performed automatic recognition of predominant instrument in sound mixes using SVM (Support Vector Machines). The feature vector applied was used before in research on automatic classification of instruments [16], and it contains sound attributes commonly used for timbre identification purposes. Most of the attributes describe low-level sound properties, based on MPEG-7 audio descriptors [4], and since many of them are multi-dimensional, derivative features were used instead (min/max value etc.). Still, the feature vector is quite long, and it contains groups of attributes that can constitute a descriptive feature set themselves. In this research, we decided to compare descriptive power of these groups. In-depth statistical analysis of the investigated sets of features for the selected instruments is presented here.

2 Automatic Identification of Musical Instruments Based on Audio Descriptors

Since audio data basically represent sequences of samples encoding the shape of the sound wave, these data usually are processed in order to extract feature vectors, and automatic classification of audio data is then performed. Sound features used for musical instrument identification purposes include time domain descriptors of sound, spectral descriptors, time-frequency descriptors and can be based on Fourier or wavelet analysis etc. Feature sets applied in research on instrument recognition include MFCC (Mel-Frequency Cepstral Coefficients), Multidimensional Analysis Scaling trajectories of various sound features, statistical properties of spectrum and so on; more details can be found in [3]. Many sound descriptors were incorporated into MPEG-7 standard for multimedia (including audio) content description [4], as they are commonly used in audio research.

Various classifiers can be applied to the recognition of musical instruments. Research performed on isolated monophonic (monotimbral) sounds so far showed successful application of k-nearest neighbors, artificial neural networks, rough-set based classifiers [12], SVM, and so on [3]. Research was also performed on polyphonic (polytimbral) data, when more than one instrument sound is present in the same time. In this case, researchers may also try to separate these sounds from the audio source. Outcome of research on polytimbral instrumental data can be found in [2], [5], [7], [11], [13], [16]. The results of research in this area are rather difficult for comparison, since various scientists utilize different data sets: of different number of classes (instruments and/or articulation), different number of objects/sounds in each class, and basically different feature sets.

Recognition for monotimbral data is relatively easy, in particular for isolated sounds, and more challenging for polytimbral data. This research is focused on identification of predominant instrument in mixes of sounds of the same pitch, as this is the most difficult case (harmonic partials in spectra overlap).

2.1 Feature Groups

In the previous research, we have been investigating automatic identification of predominant instrument in same-pitch mixes [13], [14]. The feature vector used

in this research consisted of 219 features, based on MPEG-7 audio descriptors and other parameters used in automatic sound classification [4], [16]. Although these features were used before in various configurations in similar research, this feature vector was arbitrary chosen. Therefore, we decided to check if it could be limited. Actually, the feature vector contains (among others) a few groups of descriptors that alone can be applied to sound recognition:

- *MFCC* - min, max, mean, distance, and standard deviation of the MFCC vector, averaged through the entire sound. To extract MFCC, Fourier transform is calculated for the analyzed sound frames, then logarithm of the amplitude spectrum is taken. Spectral coefficients are grouped into 40 groups according to mel scale (perceptually uniform frequency scale). For the obtained 40 coefficients, Discrete Cosine Transform is applied, yielding 13 cepstral features per frame [8]. Distance is calculated as the sum of dissimilarity (absolute difference of values) of every pair of coordinates in the vector;
- *tris*: $tris_1, \dots, tris_9$ - various ratios of harmonic partials in the spectrum; $tris_1$: energy of the fundamental to the total energy of all harmonics, $tris_2$: amplitude difference [dB] between 1st and 2nd partial, $tris_3$: ratio of the sum of 3rd and 4th partial to the total energy of harmonics, $tris_4$: ratio of partials 5-7 to all harmonics, $tris_5$: ratio of partials 8-10 to all harmonics, $tris_6$: ratio of the remaining partials to all harmonics, $tris_7$: brightness - gravity center of spectrum, $tris_8, tris_9$: contents of even/odd harmonics in spectrum;
- *Energy* - average energy of spectrum in the parameterized sound;
- *AudioSpectrumFlatness*, $flat_1, \dots, flat_{25}$ - vector describing the flatness property of the power spectrum within a frequency bin for selected bins, averaged for the entire sound; 25 out of 32 frequency bands were used;
- *AudioSpectrumBasis*: $basis_1, \dots, basis_{165}$; parameters of the spectrum basis functions, used to reduce the dimensionality by projecting the spectrum (for each frame) from high dimensional space to low dimensional space with compact salient statistical information. The spectral basis descriptor is a series of basis functions derived from the Singular Value Decomposition of a normalized power spectrum. The total number of sub-spaces in basis function in our case was 33, and for each sub-space, min/max/mean/standard deviation were extracted, yielding 33 subgroups, 5 elements in each group. The obtained values were averaged over all analyzed frames of the sound.

When investigating the significance of the 219 sound parameters used in our previous research, the attributes representing the above groups were often pointed out as significant, i.e. of high discriminant power [14]. Therefore, it seems promising to perform investigations for the groups mentioned above.

Since *AudioSpectrumBasis* group presents a high-dimensional vector itself, and the first subgroup $basis_1, \dots, basis_5$ turned out to have high discriminant power, we decided to limit the *AudioSpectrumBasis* group to $basis_1, \dots, basis_5$. In *AudioSpectrumFlatness* group, $flat_{10}, \dots, flat_{25}$ had high discriminant power, whereas $flat_1, \dots, flat_9$ had not (because their frequency ranges were below the fundamental frequencies of the analyzed sounds). Also, we decided to add *Energy* to the *tris* group, rather than investigating *Energy* as a single

conditional attribute. Therefore, the following groups (feature sets) were investigated:

- *MFCC*: $MFCC_{min}$, $MFCC_{max}$, $MFCC_{mean}$, $MFCC_{dist}$, $MFCC_{sd}$;
- *tris*: $tris_1, \dots, tris_9$, and *Energy*;
- *AudioSpectrumFlatness*: $flat_{10}, \dots, flat_{25}$;
- *AudioSpectrumBasis*: $basis_1, \dots, basis_5$;

One could discuss if such parameters as min or max of MFCC are meaningful, but since they yielded high discriminant power, we decided to investigate them.

3 Experiments and Results

In order to check how particular groups of features can discriminate instruments, we performed multivariate analysis of variance (MANOVA). For each group, our aim was to find features of the highest discriminative power, assuming that classification is performed using only features from this group. We also wanted to find out how to discriminate particular instruments, i.e. which attributes are best suited to recognize a given instrument (discriminate it from other instruments).

Audio Data. Our data represented sounds of 14 instruments from MUMS CDs [10]: B-flat clarinet, flute, oboe, English horn, trumpet, French horn, tenor trombone, violin (bowed vibrato), viola (bowed), cello (bowed), piano, marimba, vibraphone, and tubular bells. Twelve sounds, representing octave no. 4 (in MIDI notation) were used for each instrument, as a target sound to be identified in classification. Additional sounds were mixed with the main sounds, both for training and testing of the classifiers. The level of added sounds was adjusted to 6.25%, $12.5/\sqrt{2}\%$, 12.5%, $25/\sqrt{2}\%$, 25%, $50/\sqrt{2}\%$, and 50% of the level of the main sound, since our goal was to identify the predominant instrument. For each main instrumental sound, additional 4 mixes were prepared for training for each level: with white noise, pink noise, with triangular and saw-tooth wave of the same pitch as the main sound. For testing, another set of mixes was used. Namely, each sound to identify was mixed with 13 sounds of the same pitch representing the remaining 13 instruments from our data set. Again, the sounds added in mixes were adjusted in level, at the same levels as in training. Finally, all single instrumental sounds and all training mixes (for all levels) were used as the training set, and all test mixes were used as the test set.

3.1 Materials and Methods

In the described experiments, our target was to recognize instrument as a class. For each instruments, sound samples represented various pitch (12 notes from one octave) and various levels of added sounds. Since our goal was to identify instrument, we omitted distinguishing for particular levels or pitch values.

MANOVA was used in our research to verify the hypothesis about the lack of differences (between the instruments) for vectors of mean values of the selected features. The test statistic based on Wilks Λ [9] was applied, which can be transformed to a statistic having approximately an F distribution. This form of MANOVA results make it easier to obtain p -value and is definitely preferred [1]. In case of rejecting this hypothesis, i.e. in case of finding out that there existed significant differences of means between instruments, we applied the post hoc comparisons between average values of the studied features, based on HSD Tukey test [15]. These comparisons are presented in the form of homogeneous groups defined by mean values of a given feature, and consisting of the instruments which are not significantly different wrt. this feature for $\alpha = 0.05$. Therefore, mean values of each feature defined homogenous groups of instruments. If differences between means for some instruments were not statistically significant, they constituted a group. The less instruments (sometimes even only one) in such a homogenous group, the higher the discerning power of a given feature.

3.2 Analysis of *AudioSpectrumBasis* Parameters

The results of analysis can be shown as tables presenting groups of instruments, homogenous wrt. particular feature from a given feature set (Figure 10). Our *AudioSpectrumBasis* set consisted of $basis_1, \dots, basis_5$. Mean values for these features significantly differed between the instruments ($F = 188.0, p < 0.01$).

The results of post hoc analysis revealed that $basis_4, basis_5$ and $basis_1$ distinguish instruments to a large extent. The influence of $basis_2$ and $basis_3$ on differentiation between instruments is rather small. Marimba, piano, vibraphone, and the pair of bells and French horn, often determine separate groups. Piano, vibraphone, marimba, cello and trombone are very well separated by $basis_5$, since each of these instruments constitute a 1-element group. Piano, vibraphone, marimba, and cello are separated by $basis_4$ too. Also, $basis_1$ separates marimba and piano; $basis_3$ only discerns marimba from other instruments (only 2 groups produced); $basis_2$ does not separate any single instrument.

3.3 Analysis of MFCC-Based Parameters

The results of MANOVA indicate that the mean values of MFCC features differ significantly between the studied instruments ($F = 262.84, p < 0.01$). The analysis of homogeneous groups shows that $MFCC_{sd}$ and $MFCC_{max}$ yielded the highest difference of means, while $MFCC_{min}$ - the lowest one. The piano determined the separate group for every parameter from our MFCC feature set, so this instrument is very well distinguished by MFCC. However, there were no parameters here that would be capable to distinguish between marimba and flute. These two instruments were always situated in the same group, since the average values of studied parameters for these instruments do not differ too much. Vibraphone and bells were in different groups only for $MFCC_{mean}$.

For MFCC-based features, each feature defined 6-9 groups, homogenous wrt. the mean value of a given feature (tables not shown because of space limitations):

Instrument	mean basis5	1	2	3	4	5	6	7	8	9	10
piano	0.017810	x									
vibraphone	0.022358		x								
marimba	0.024512			x							
tubular bells	0.027763				x						
French horn	0.027912					x					
cello	0.029870						x				
trombone	0.032427							x			
flute	0.033932								x		
clarinet	0.034693								x	x	
English horn	0.035160								x	x	x
trumpet	0.035591								x	x	
viola	0.036379									x	
oboe	0.038118										x
violin	0.038379										x

Instrument	mean basis4	1	2	3	4	5	6	7	8	9
piano	10.33462	x								
vibraphone	11.79222		x							
marimba	13.07193			x						
French horn	15.82109				x					
tubular bells	16.05066					x				
cello	17.58667						x			
trombone	18.72267							x		
flute	19.64378								x	
English horn	20.70552									x
clarinet	20.81804									x
trumpet	21.46201									x
viola	22.19190									x
oboe	22.73954									x
violin	23.05738									x

Instrument	mean basis3	1	2
marimba	0.160810	x	
violin	0.169641		x
oboe	0.169678		x
viola	0.170077		x
trumpet	0.170234		x
English horn	0.170362		x
clarinet	0.170439		x
flute	0.170605		x
trombone	0.170889		x
cello	0.171417		x
French horn	0.171706		x
tubular bells	0.171743		x
vibraphone	0.172575		x
piano	0.173102		x

Instrument	mean basis2	1	2	3	4	5	6
marimba	0.192175	x					
piano	0.195695		x	x			
vibraphone	0.198370			x			
tubular bells	0.204552				x		
French horn	0.205662					x	
trombone	0.211415						x
clarinet	0.212789						x
English horn	0.213999						x
trumpet	0.214331						x
flute	0.214437						x
cello	0.215237						x
oboe	0.217359						x
violin	0.219894						x
viola	0.220354						x

Instrument	mean basis1	1	2	3	4	5	6	7	8	9
marimba	0.078112	x								
flute	0.084683		x							
English horn	0.086258			x	x					
oboe	0.088256				x	x				
violin	0.090201					x	x			
trombone	0.092303						x	x		
viola	0.093202							x	x	x
vibraphone	0.093699								x	x
cello	0.094367								x	x
clarinet	0.096370								x	x
French horn	0.097365									x
trumpet	0.103000									x
tubular bells	0.104220									x
piano	0.120136									x

Fig. 1. Homogenous groups of instruments for $basis_1, \dots, basis_5$

– standard deviation of MFCC $MFCC_{sd}$ - 9 groups:

1. piano (mean=2.040694)
2. French horn (4.977832), tubular bells (5.093240), and vibraphone (5.246537)
3. vibraphone (5.246537) and trumpet (5.397999)
4. trombone (5.719065)
5. clarinet (6.052808) and flute (6.124170)
6. flute (6.124170) and marimba (6.412859)
7. English horn (6.807668) and cello (6.921288)
8. oboe (7.215706)
9. viola (7.509446) and violin (7.747165)

- distance of MFCC $MFCC_{dist}$ - 8 groups:
 1. piano (mean=173.8251)
 2. French horn (433.2952)
 3. tubular bells (461.3751), vibraphone (471.0583), and trumpet (479.7590)
 4. trumpet (479.7590) and trombone (497.5668)
 5. flute (532.8474), clarinet (553.1953) and marimba (555.1694)
 6. cello (600.2499)
 7. English horn (623.1473)
 8. violin (647.3444), oboe (648.5515), and viola (659.6647)
- mean of MFCC $MFCC_{mean}$ - 8 groups:
 1. violin (mean=-2.48152)
 2. flute (-2.27792), viola (-2.22700), marimba (-2.17361), and cello (-2.13859)
 3. French horn (-1.74975), oboe (-1.67394), and trombone (-1.62327)
 4. trombone (-1.62327) and English horn (-1.51288)
 5. English horn (-1.51288), and vibraphone (-1.36259)
 6. clarinet (-0.79084) and trumpet (-0.70124)
 7. tubular bells (-0.48742)
 8. piano (-0.24414)
- maximum of MFCC $MFCC_{max}$ - 9 groups:
 1. piano (mean=3.55248)
 2. vibraphone (7.83704), and tubular bells (8.39608)
 3. tubular bells (8.39608), and trumpet (9.01000)
 4. trumpet (9.01000) and French horn (9.47068)
 5. clarinet (10.61529), and trombone (10.70729)
 6. marimba (11.92844), flute (12.21103) and English horn (12.86478)
 7. oboe (14.11557) and cello (14.74821)
 8. viola (15.78921)
 9. violin (18.68331)
- minimum of MFCC $MFCC_{min}$ - 6 groups:
 1. viola (mean=-15.1502)
 2. oboe (-14.1205), violin (14.1015) and marimba (-13.9336)
 3. English horn (-12.6130), clarinet (-12.5465), cello (-12.5415), trumpet (-12.5147), trombone (-12.4508), flute (-12.4024), and vibraphone (-11.9982)
 4. vibraphone (-11.9982) and tubular bells (-11.2825)
 5. tubular bells (-11.2825), and French horn (-10.7254)
 6. piano (-4.9307)

Piano, cello, viola, violin, bells, English horn, oboe, French horn, and trombone constitute separate groups when MFCC set is used for parameterization, so these instruments can be easily recognized on the basis of MFCC. On the other hand, some groups overlap, i.e. the same instrument may belong to 2 groups.

3.4 Analysis of *tris* Parameters

The results of MANOVA show that mean values of *tris* parameters were significantly different for the studied set of instruments ($F = 248, p < 0.01$). For the *tris* feature set consisting of $tris_1, \dots, tris_9$ parameters and *Energy*, each parameter defined from 3 to 9 groups, as follows (only selected groups are shown):

- *tris*₉ - 9 groups:
 1. vibraphone (0.055816), piano (0.064567), flute (0.069978), marimba (0.07765)
 2. piano (0.064567), flute (0.069978), marimba (0.07765), French horn (0.08705)
 3. French horn (0.087050), and tubular bells (0.111881)
 4. cello (0.142859) and viola (0.165404)
 5. viola (0.165404) and trombone (0.174318)
 6. English horn (0.226685) and violin (0.242791)
 7. violin (0.242791) and oboe (0.262232)
 8. trumpet (0.386531)
 9. clarinet (0.524384)
- *tris*₈ - 9 groups:
 1. marimba (0.094389) and piano (0.121932)
 2. piano (0.121932), vibraphone (0.129555) and clarinet (0.144759)
 3. cello (0.240345), viola (0.258904) and French horn (0.273323)
 4. viola (0.258904), French horn (0.273323), and violin (0.281908)
 5. tubular bells (0.370951) and flute (0.392088)
 6. flute (0.392088) and trumpet (0.408881)
 7. trombone (0.449997)
 8. English horn (0.510827)
 9. oboe (0.549077)
- *tris*₅ - 4 groups:
 1. French horn (0.002759), piano (0.005540), flute (0.005567), English horn (0.006739), trombone (0.006820), and marimba (0.007010)
 2. vibraphone (0.012816) and oboe (0.014900)
 3. violin (0.02379), cello (0.02399), clarinet (0.024949), and viola (0.029246)
 4. viola (0.029246), tubular bells (0.031247), and trumpet (0.033919)
- *tris*₃ - 9 groups:
 1. marimba (0.032259), piano (0.045117) and flute (0.048801)
 2. flute (0.048801) and viola (0.076112)
 3. viola, vibraphone (0.080386), French horn (0.084755), cello (0.096423)
 4. cello (0.096423) and tubular bells (0.116387)
 5. tubular bells (0.116387) and violin (0.137848)
 6. trombone (0.171827)
 7. English horn (0.325869) and oboe (0.343625)
 8. oboe (0.343625) and clarinet (0.368974)
 9. clarinet (0.368974) and trumpet (0.382305)
- *tris*₁ - 8 groups:
 1. trumpet (0.166245) and oboe (0.188429)
 2. English horn (0.259002)
 3. trombone (0.307373) and clarinet (0.327424)

4. piano (0.404295) and tubular bells (0.448198)
5. tubular bells (0.448198), violin (0.468752) and French horn (0.477348)
6. flute (0.537660) and viola (0.575315)
7. viola (0.575315) and cello (0.616396)
8. marimba (0.688202) and vibraphone (0.727886)

$Tris_3$, $tris_8$ and $tris_9$ produced the highest number of homogeneous groups. Some wind instruments (trumpet, trombone, English horn, oboe), or their pairs, were distinguished most easily - determined separate groups for the features forming 8-9 homogeneous groups. Mean values of *Energy* were especially different for piano and violin. $Tris_2$ and $tris_5$ yielded the lowest number of groups.

3.5 Analysis of *AudioSpectrumFlatness* Parameters

AudioSpectrumFlatness feature set consisted of $flat_{10}, \dots, flat_{25}$ parameters; strict details of this analysis are not shown here because of space limitations.

The vector of means for these parameters, similarly to other ones, significantly differed between studied instruments ($F = 94.00$, $p < 0.01$). The post hoc comparisons show the high discriminating power of $flat_{10}, \dots, flat_{14}$, distinguishing marimba, vibraphone and piano. For increasing i in $flat_i$, the group consisting of marimba, vibraphone, and French horn was growing - other instruments were added. At the same time, homogeneous group determined by oboe, clarinet, trumpet, violin, and English horn was differentiating into separate groups.

4 Summary and Conclusions

In this paper, we compared feature sets used for musical instrument sound classification. Mean values for data representing given instruments and statistical tests for these data were presented and discussed. Also, for each feature, homogeneous groups were found, representing instruments which are similar wrt. this feature. Instruments, for which the mean values of a given feature were significantly different, were assigned to different groups, and instruments, for which the mean values were not statistically different, were assigned to the same group.

Sound features were grouped according to the parameterization method, including MFCC, proportions of harmonics in the sound spectrum, and MPEG-7 based parameters (*AudioSpectrumFlatness*, *AudioSpectrumBasis*). These groups were chosen as a conclusion of our previous research, indicating high discriminant power of particular features for instrument discrimination purposes.

Piano, vibraphone, marimba, cello, English horn, French horn, and trombone turned out to be the most discernible instruments. It is very encouraging, because marimba and vibraphone represent idiophones (a part of percussion group), so sound is played by striking and is not sustained (similarly for piano), so there is no steady state, thus making parameterization more challenging. Also, since the investigations were performed for small groups of features (5-16), we conclude that these groups constitute a good basis for instrument discernment.

The results enabled us to indicate, for each instrument, which parameters within a given group represent the highest distinguishing power, and indicate

which features are most suitable to distinguish this instrument. Experiments on musical instrument identification with the feature vector described in this paper were also performed, using random forests as classifiers [6]. The obtained results confirmed significance of particular features, and yielded very good accuracy.

Acknowledgements. The presented work was partially supported by the Research Center of PJIIT, supported by the Polish National Committee for Scientific Research (KBN). This material is also based in part upon work supported by the National Science Foundation (NSF) under Grant Number IIS-0414815.

The authors would like to thank Elżbieta Kubera from the University of Life Sciences in Lublin for help with preparing the initial data for experiments.

References

1. Bartlett, H., Simonite, V., Westcott, E., Taylor, H.: A comparison of the nursing competence of graduates and diplomates from UK nursing programmes. *Journal of Clinical Nursing* 9, 369–381 (2000)
2. Dziubinski, M., Dalka, P., Kostek, B.: Estimation of musical sound separation algorithm effectiveness employing neural networks. *J. Int. Inf. Systems* 24(2-3), 133–157 (2005)
3. Herrera, P., Amatriain, X., Batlle, E., Serra, X.: Towards instrument segmentation for music content description: a critical review of instrument classification techniques. In: *International Symposium on Music Information Retrieval ISMIR* (2000)
4. ISO/IEC JTC1/SC29/WG11: MPEG-7 Overview, <http://www.chiariglione.org/>
5. Itoyama, K., Goto, M., Komatani, K., Ogata, T., Okuno, H.G.: Instrument Equalizer for Query-By-Example Retrieval: Improving Sound Source Separation Based on Integrated Harmonic and Inharmonic Models. In: *9th Int. Conf. ISMIR* (2008)
6. Kursa, M., Rudnicki, W., Wieczorkowska, A., Kubera, E., Kubik-Komar, A.: Musical Instruments in Random Forest. In: *18th Int. Symp. ISMIS* (2009)
7. Little, D., Pardo, B.: Learning Musical Instruments from Mixtures of Audio with Weak Labels. In: *9th Int. Conf. on Music Information Retrieval ISMIR* (2008)
8. Logan, B.: Mel Frequency Cepstral Coefficients for Music Modeling. In: *International Symposium on Music Information Retrieval MUSIC IR* (2000)
9. Morrison, D.F.: *Multivariate statistical methods*, 3rd edn. McGraw-Hill, NY (1990)
10. Opolko, F., Wapnick, J.: MUMS - McGill University Master Samples. CD's (1987)
11. Viste, H., Evangelista, G.: Separation of Harmonic Instruments with Overlapping Partials in Multi-Channel Mixtures. In: *IEEE Workshop WASPAA 2003* (2003)
12. Wieczorkowska, A., Czyzewski, A.: Rough Set Based Automatic Classification of Musical Instrument Sounds. In: *International Workshop RSKD*. Elsevier, Amsterdam (2003)
13. Wieczorkowska, A., Kubera, E., Kubik-Komar, A.: Analysis of Recognition of a Musical Instrument in Sound Mixes Using Support Vector Machines. In: Nguyen, H.S., Huynh, V.-N. (eds.) *SCKT 2008 Hanoi, Vietnam (PRICAI 2008)*, pp. 110–121 (2008)
14. Wieczorkowska, A., Kubik-Komar, A.: Application of discriminant analysis to distinction of musical instruments on the basis of selected sound parameters. In: *International Conference on Man-Machine Interactions ICMMI* (to appear, 2009)
15. Winer, B.J., Brown, D.R., Michels, K.M.: *Statistical principals in experimental design*, 3rd edn. McGraw-Hill, New York (1991)
16. Zhang, X.: *Cooperative Music Retrieval Based on Automatic Indexing of Music by Instruments and Their Types*. Ph.D thesis, Univ. North Carolina, Charlotte (2007)

Alternative Formulas for Rating Prediction Using Collaborative Filtering

Amar Saric, Mirsad Hadzikadic, and David Wilson

College of Computing and Informatics
The University of North Carolina at Charlotte, 9201 University City Blvd, Charlotte,
NC 28223, USA
{asaric,mirsad,davils}@uncc.edu

Abstract. This paper proposes and evaluates several alternate design choices for common prediction metrics employed by neighborhood-based collaborative filtering approach. It first explores the role of different baseline user averages as the foundation of similarity weighting and rating normalization in prediction, evaluating the results in comparison to traditional neighborhood-based metrics using the MovieLens data set. The approach is further evaluated on the Netflix movie data set, using a baseline correlation formula between movies, without meta-knowledge. For the Netflix domain, the approach is augmented with a significance weighting variant that results in an improvement over the original metric. The resulting approach is shown to improve accuracy for neighborhood-based collaborative filtering, and it is general and applicable to establishing relationships among agents with a common list of items which establish their preferences.

Keywords: Collaborative filtering, Personalized Recommendation, Rating Prediction, Similarity measure.

1 Introduction

Collaborative filtering recommender systems employ ratings-based user profiles in order to make item recommendations or predictions about user ratings for items. Suitable items for recommendation, such as suggested movies to watch, are identified not because their description matches them with a target user, but rather because these items have been liked by users who are similar to the target user in terms of how they have rated other items. Collaborative filtering can employ a variety of foundational algorithms, but the most prevalent are the so-called neighborhood-based methods. Neighborhood-based methods first locate a subset of the user population, based on their similarity to the current or active user. Typically, then a weighted combination of the neighbors' ratings are employed as the basis for rating-prediction or recommendation for the active user. Herlocker et al. [4] identified tested several main aspects of the design space for neighborhood-based collaborative filtering, including: similarity weighting, significance weighting, variance weighting, neighborhood selection, rating normalization, and neighbor contribution weighting. In this paper, we focus on improving the accuracy in neighborhood-based collaborative filtering along

the dimensions of computing the neighborhood and computing the prediction (see for instance [3], [6] or [9]). This paper proposes and evaluates several alternate design choices for common prediction metrics employed by neighborhood-based collaborative filtering approaches. It first explores the role of different baseline user averages as the foundation of similarity weighting and rating normalization in prediction, evaluating the results in comparison to traditional neighborhood-based metrics using the MovieLens data set. The approach is further evaluated on the Netflix movie data set, using a baseline correlation formula between movies, without meta-knowledge. For the Netflix prize [8] domain, the approach is augmented with a significance weighting variant that results in an improvement over the original Netflix metric. Evaluation results show that our approach improves accuracy for neighborhood-based collaborative filtering. The approach is general and applicable to establishing relationships among agents with a common list of items that establish their preferences.

2 Collaborative Filtering Formulas

The baseline algorithm in the literature for neighborhood-based collaborative filtering uses the Pearson correlation [4]. For raters M and N it is calculated using

$$r_{MN} = \frac{\sum_k (M_k - \bar{M})(N_k - \bar{N})}{\sqrt{\sum_k (M_k - \bar{M})^2} \sqrt{\sum_k (N_k - \bar{N})^2}},$$

where M_k and N_k respectively are ratings for the item k , $\bar{M} = \frac{1}{\text{dim}_{MN}} \sum_k M_k$ and

$\bar{N} = \frac{1}{\text{dim}_{MN}} \sum_k N_k$ the mean values, and dim_{MN} denotes the total number of items

rated by both users. If $\text{dim}(M, N) = 0$, then r_{MN} is simply set to zero. All sums in the above formula are computed over the ratings which both users have in common. \bar{M} is strictly speaking a function of N , and vice versa, but writing it down explicitly would unnecessarily complicate the formulas. The predictions are then computed using the following formula

$$M_j = \bar{M} + \frac{\sum_{N \in \text{Raters} \setminus \{M\}} (N_j - \bar{N}) r_{MN}}{\sum_{N \in \text{Raters} \setminus \{M\}} |r_{MN}|},$$

where \bar{M} is the mean of all ratings for a given rater. This is the standard GroupLens approach for the so-called user-to-user ratings prediction. We considered the raters/users to be ‘agents’ and compare them among each other to establish links between them (at least conceptually). The viewpoint can be changed so that the items become ‘agents’, the mechanics and, more importantly, also the underlying logic, stay unchanged – it all depends on what is considered the ‘agent’ the user or the rated item. Here we compare the items to each other, with M and N in the above formulas

denoting two items and summation performed over the common raters for these items. Also the summation in the prediction formulas is performed over similar items not raters. This is commonly referred to as item-to-item collaborative filtering. In order to disambiguate the notation in what follows, we define two different correlation coefficients as

$$\overline{r_{MN}} = \frac{\sum_k (M_k - \overline{M})(N_k - \overline{N})}{\sqrt{\sum_k (M_k - \overline{M})^2} \sqrt{\sum_k (N_k - \overline{N})^2}}, \quad \overline{\overline{r_{MN}}} = \frac{\sum_k (M_k - \overline{\overline{M}})(N_k - \overline{\overline{N}})}{\sqrt{\sum_k (M_k - \overline{\overline{M}})^2} \sqrt{\sum_k (N_k - \overline{\overline{N}})^2}}.$$

Here $\overline{\overline{M}}$ denotes the mean values over all ratings of the user M , and $\overline{\overline{M}}$ the mean value only over the ratings that the user has in common with the user N . The summation in both formulas goes over all common ratings. In what follows we will discuss collaborative filtering and show how additional formulas can be obtained. Our main goal will be to see what changes the basic ideas of collaborative filtering allow. We start with

$$M_j = \overline{\overline{M}} + \frac{\sum_{N \in \text{Raters} \setminus \{M\}} (N_j - \overline{\overline{N}}) \overline{\overline{r_{MN}}}}{\sum_{N \in \text{Raters} \setminus \{M\}} |\overline{\overline{r_{MN}}}|}, \quad (1)$$

which is the original GroupLens formula as presented in the paper by Resnick, Iacovou, Suchak, Bergstrom and Riedl [9], restated in our notation. Towards the end of the paper we will also move away from using only linear elements in the prediction formula. There are several different possibilities to alter this formula, for example using normalization. Our main criticism of the formula, however, is that it utilizes the mean over all ratings $\overline{\overline{M}}$ to offset the predictions, while computing the Pearson correlation over only the property values (i.e. ratings) that both agents, say users, have in common. Therefore we test several formulas using the mean calculated over all the values with other agents $\overline{\overline{M}}$ as well as only over the common ratings $\overline{\overline{M}}$ for a given user. Using the mean averages over all the ratings the equivalent of the above

$$M_j = \overline{\overline{M}} + \frac{\sum_{N \in \text{Raters} \setminus \{M\}} (N_j - \overline{\overline{N}}) \overline{\overline{\overline{r_{MN}}}}}{\sum_{N \in \text{Raters} \setminus \{M\}} \overline{\overline{\overline{r_{MN}}}}}. \quad (2)$$

However, some raters might be reluctant to give the best or worst possible ratings on the Likert scale. Therefore, at least for user-to-user comparisons, a possible change from which we might expect some improvement is to adjust the offset from the mean in the prediction formula, which is in fact based on the ratings given by other users. If we try to scale these contributions using the same norm as in the Pearson correlation (see for instance [5]), the GroupLens formula (1) becomes

$$M_j = \overline{\overline{M}} + \frac{\sum_{N \in \text{Raters} \setminus \{M\}} \frac{\sqrt{\sum_k (M_k - \overline{\overline{M}})^2}}{\sqrt{\sum_k (N_k - \overline{\overline{N}})^2}} (N_j - \overline{\overline{N}}) \overline{r_{MN}}}{\sum_{N \in \text{Raters} \setminus \{M\}} |\overline{r_{MN}}|} . \tag{3}$$

Normalization makes little sense if we are comparing items, since in this case we cannot talk of rating ‘tendencies’, although the variance itself might of course be useful. By using the averages over all the ratings instead, the above formula is transformed to

$$M_j = \overline{\overline{M}} + \frac{\sqrt{\sum_k (M_k - \overline{\overline{M}})^2}}{\sum_{N \in \text{Raters} \setminus \{M\}} |\overline{r_{MN}}|} \sum_{N \in \text{Raters} \setminus \{M\}} \frac{\overline{r_{MN}}}{\sqrt{\sum_k (N_k - \overline{\overline{N}})^2}} (N_j - \overline{\overline{N}}) . \tag{4}$$

Nevertheless, the behavior of agents outside the range of values common to both users cannot be implied by their behavior within this range. This is also the most likely reason why the GroupLens formula uses $\overline{\overline{M}}$ instead of \overline{M} . But this itself is not consistent, since the mean value calculated over all the ratings is used as the starting point to which the contributions from the other users are added. At first glance, it might seem that there is no other way to do this but to use \overline{M} . However, the following formula overcomes this “imperfection”

$$M_j = \frac{\sum_{N \in \text{Raters} \setminus \{M\}} |\overline{r_{MN}}| \overline{M}}{\sum_{N \in \text{Raters} \setminus \{M\}} |\overline{r_{MN}}|} + \frac{\sum_{N \in \text{Raters} \setminus \{M\}} (N_j - \overline{\overline{N}}) \overline{r_{MN}}}{\sum_{N \in \text{Raters} \setminus \{M\}} |\overline{r_{MN}}|} . \tag{5}$$

by using a weighted average also in the first term of the formula. The contribution of the other users is added to the average of user M for the same range and the total result scaled by the value of their Pearson correlation. Thus, every user contributes a value to the total estimate, which is proportional to the absolute value of the correlation coefficient. Alternatively, we can also rewrite this formula as

$$M_j = \frac{\sum_{N \in \text{Movies} \setminus \{M\}} |\overline{r_{MN}}| (\overline{M} + \text{sgn}(\overline{r_{MN}})(N_j - \overline{\overline{N}}))}{\sum_{N \in \text{Movies} \setminus \{M\}} |\overline{r_{MN}}|} .$$

The prediction formula is therefore a weighted average of single predictions based on other agents. This is also valid for item-to item collaborative filtering. After normalizing the ratings in the formula (6) we obtain

$$M_j = \frac{\sum_{N \in Raters \setminus \{M\}} \overline{r_{MN}} \overline{M}}{\sum_{N \in Raters \setminus \{M\}} \overline{r_{MN}}} + \frac{\sum_{N \in Raters \setminus \{M\}} \frac{\sqrt{\sum_k (M_k - \overline{M})^2}}{\sqrt{\sum_k (N_k - \overline{N})^2}} (N_j - \overline{N}) \overline{r_{MN}}}{\sum_{N \in Raters \setminus \{M\}} \overline{r_{MN}}}. \tag{6}$$

All of the above formulas are reasonable alternatives to the GroupLens formula (1). One could think that the alternative formulas are computationally more expensive, but the use of \overline{M} the means in Formula (1), which is computed only over the common values, actually requires the same number of passes through data as the Formula (5) – only the additional means have to be stored along with the correlation values. The Formulas (2) and (4) are in fact easier to compute since the values for the means can be pre-computed (for all user ratings), stored, and used in all subsequent calculations. Possibly one formula will calculate a prediction, while the other one will not, in which case we use the average for prediction. This does not happen frequently.

2.1 Experimental Results

The evaluation was performed using the MovieLens data containing 100,000 ratings, which provides data splits suitable for use as training and test sets. The data can be obtained from the GroupLens webpage [1]. The data was randomly split into 5 base and test sets, of 80,000 and 20,000 ratings respectively, in order to be able to empirically evaluate formulas. We used mean absolute error (MEA) as performance measure for comparisons.

The graph in figure 1 indicates that best predictions are obtained for Formulas (5) and (6). The evaluation of these formulas was performed using the mean absolute error as a measure. All users for which the absolute value of correlation with the

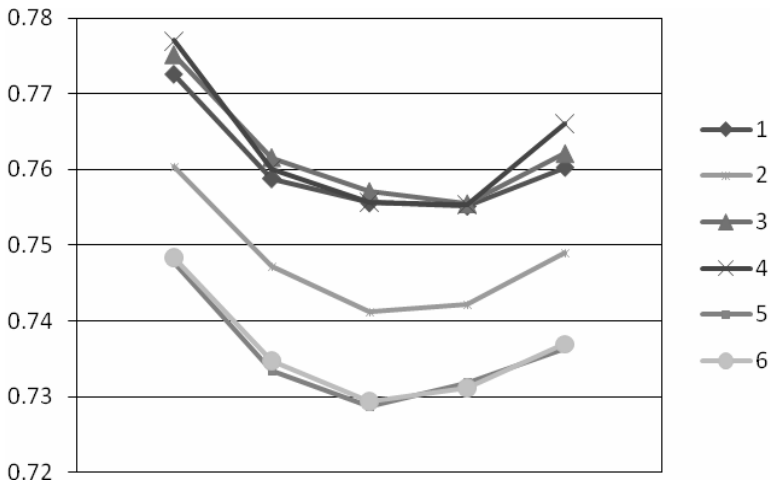


Fig. 1. Mean absolute error using cross-validation on 5 different splits for formulas (1) - (6)

current user was below 0.1 were ignored in the computation, as well as all the users having only one rating in common. After checking whether the predicted value is out of range, i.e. less than 1 or greater than 5 in the case of MovieLens data, and correcting it to the closest possible value, we conclude that the normalized prediction formulas are favored again over other alternatives. This is shown in Figure 2, which brings only minimal gains, possibly because it does not occur very often. Also, there is the possibility of varying the “cut-off” value for the correlation with a small absolute value, as well as discarding those users who have less than a fixed number of ratings in common with the user for whom we are trying to make the prediction in order to remove noise. Consequently, the interesting question is how the different formulas would behave in these cases. In the rest of the text we look into these two issues in more detail. Figure 3, for example, shows the values obtained after discarding all the users with a correlation of less than 0.5, and clipping the predicted value if it falls outside the boundaries. Obviously, increasing this value does not necessarily decrease the overall mean error.

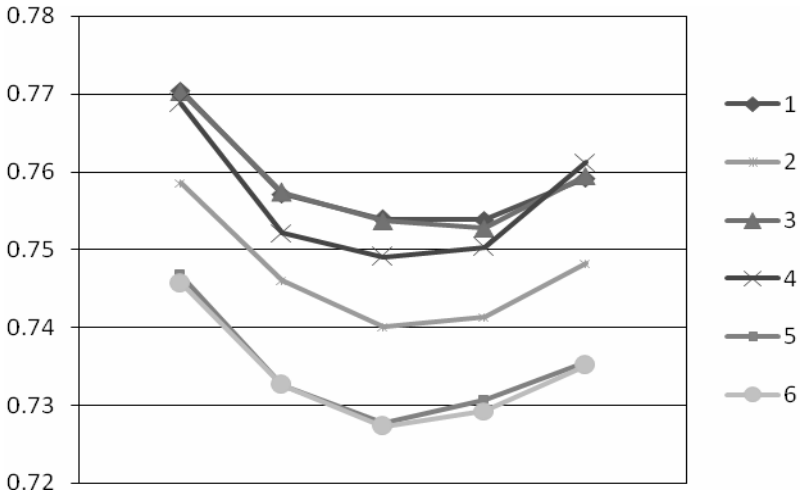


Fig. 2. Clipping values in case out of range predictions for formulas (1) - (6)

Finally, let us examine what happens when we exclude all the correlations that are based on 10 or fewer common ratings, and disallow any contribution from these users. The important thing to note is that this has to do with the trust in the similarity measure and that it does improve results (Figure 4). This will also, obviously, decrease the number of cases in which we are able to make a prediction. Again, the predicted values in this example were corrected if they were out of range, and correlation values were discarded if they were below 0.1. In the rest of the text we look in more detail into these two issues.

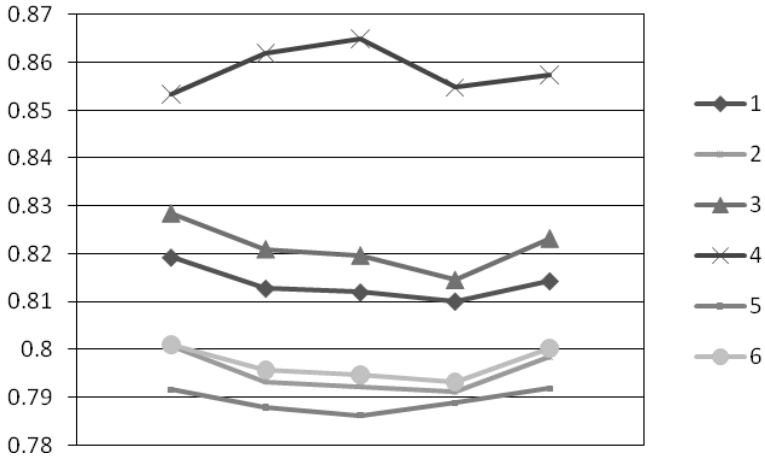


Fig. 3. Excluding small correlation values from contributing to the prediction, does not necessarily improve the MAE

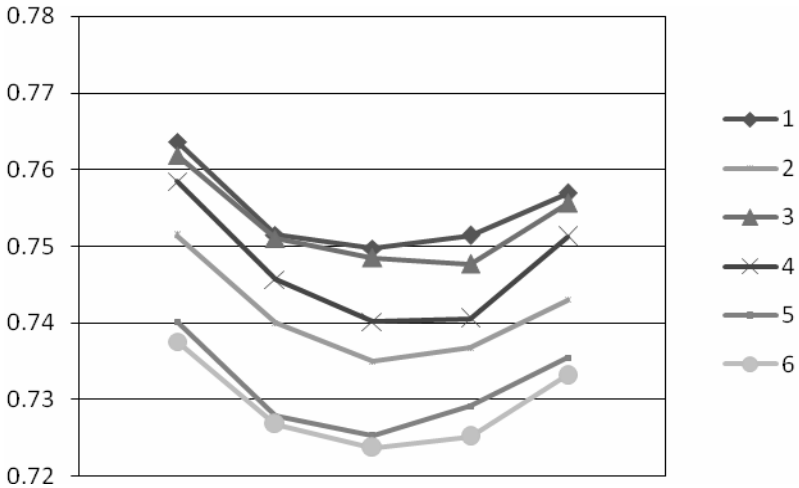


Fig. 4. Predictions calculated using correlation values based on at least 10 common ratings for formulas (1) - (6)

3 A More Reliable Similarity Measure

Based on our initial experiments, we wanted to test for increased scale on real-world data. We used item-to-item, collaborative filtering on the Netflix data [2] to verify that the approach can result in improvements over the standard formula on real problems,

other than that we are following the original approach by Resnick et al. [9]. No clustering or transformation of data was performed or additional information used (such are the dates when the rating was made, the genre of the move etc.). Formula (5) based item-to-item collaborative filtering did not improve the score, but it also performed just as good as Netflix's original algorithm. Better results for the dataset were obtained by replacing absolute values by squares, which effectively amounts to using a different similarity measure: Specifically, we insert $\text{sgn}(r_{MN}) \left| \overline{r_{MN}} \right|^2$ into the equation (5), which replaces $\overline{r_{MN}}$, and also use item-to-item comparisons. The resulting formula is

$$M_j = \frac{\sum_{N \in \text{Movies} \setminus \{M\}} \left| \overline{r_{MN}} \right|^2 (\overline{M} + \text{sgn}(\overline{r_{MN}})(N_j - \overline{N}))}{\sum_{N \in \text{Movies} \setminus \{M\}} \left| \overline{r_{MN}} \right|^2} \quad (7)$$

which is a weighted average of predictions based on other users. The ratings were not normalized, mostly because we were using item-to-item comparisons, since in this case one cannot expect that ratings provided by different users would vary by the same average amount from the mean for a given movie. Rather, one would expect such rating tendencies to be valid for raters and not the items. Adjusting the value of the weights by taking into account the number of common ratings yields the following slightly involved but otherwise straight-forward formula

$$M_j = \frac{\sum_{N \in \text{Movies} \setminus \{M\}} \beta(M, N)^2 (\overline{M} + \text{sgn}(\overline{r_{MN}})(N_j - \overline{N}))}{\sum_{N \in \text{Movies} \setminus \{M\}} \beta(M, N)^2} \quad (8)$$

where $\beta(M, N) = \overline{r_{MN}} \tanh(\lambda \dim_{MN})$, the number of users who have rated movies M and N is given by \dim_{MN} , and λ is a parameter to be determined. $\beta(M, N)^2$, which replaces $\left| \overline{r_{MN}} \right|^2$ in formula (5), includes a penalty function used to scale the similarity measure (here the square of the Pearson correlation) based on the number of ratings the two movies have in common, to construct a new similarity measure. The value λ determines, roughly speaking, the minimal number of users who rated both movies, the contributions coming from other movies with only few raters in common are penalized. This is, in principle, only a minor modification of discarding only loosely (anti-)correlated movies. Appropriate λ can be found empirically. For the Netflix data values between 0.001 and 0.003 seems to work well. Figure 5 shows the contribution of the $\tanh^2(\lambda \dim(M, N))$ term for $\lambda = 0.002$. This approach resulted in a 2.9% improvement over the original root mean squared error of 0.9514 achieved by the original "Cinematch" algorithm.

Additionally, we have also experimented with sorting the ratings for other movies in decreasing order based on the similarity values β and aborting summation after "enough" of the other movies were considered. We used $\sum_{N \in \text{Movies} \setminus \{M\}} r_{MN}^2 \tanh^2(\lambda \dim_{MN}) < 2.5$ as

condition, but the effect was marginal – similar to that of rounding values when they are out of range – affecting only the third digit after decimal point in the RMSE. However, it is possible to do so, and, it did result in a small improvement albeit at the expense of having to determine a suitable cutoff value. We found the values between 2 and 3 to work well with the values of λ in the above range. The only drawback is that one has to change the logic of the application. The changes described previously consisted of storing additional values in calculations which already had to be performed in order to compute predictions based on formula (1). They could therefore be incorporated easily into any system using the approach described by [9].

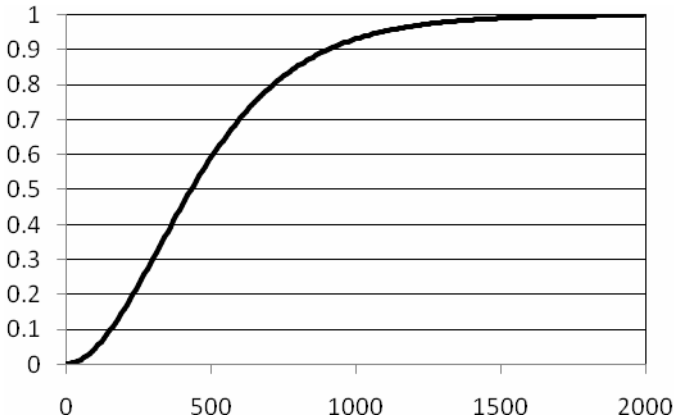


Fig. 5. Trust allocated based on the number of common ratings. The similarity measure is multiplied by this value to obtain the contribution of a movie in the prediction of rating for another movie.

4 Conclusion

Although additional tests should be performed, it seems realistic that the above formulas could be used successfully instead of the standard prediction formula [4]. It is noteworthy that the errors obtained for the formulas (5) and (6) in our test runs were consistently below those for the most commonly used prediction formula (1). We consider them also to be somewhat more appealing because of the way the averages are calculated. We therefore propose that formula (5) be the default formula for collaborative filtering, and one optionally use penalty functions as in formulas (7) and (8). Any system which uses (1), can easily be modified to use (5), (6) or (7), and, if one is willing to empirically determine the additional parameter, also (8). There is no architectural reason not to simply replace the formula and leave the rest of the system unaltered. It is also somewhat surprising that, at least for the data sample used, the normalization seems to have had only a limited effect. Perhaps, one could obtain better results with other norms, for instance $\max_k |U_k - \bar{U}|$, or $\sum_k |U_k - \bar{U}|$.

Experiments with the Netflix database show that the modified formulas allow for improvements over the original collaborative filtering. The lesson here is that one can improve the results by using penalty functions based on the number of common ratings in addition to the similarity measure.

References

1. GroupLens, <http://www.grouplens.org/>
2. Bennett, J., Lanning, S.: The Netflix Prize. In: Proceedings of the KDD Cup and Workshop (2007)
3. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative Filtering. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, Berkeley, pp. 230–237 (1999)
4. Herlocker, J., Konstan, J., Riedl, J.: Empirical Analysis of Design Choices in Neighborhood-based Collaborative Filtering Algorithms. *Informational Retrieval* 5(4), 287–310 (2002)
5. Jin, R., Si, L.: A Study of Methods for Normalizing User Ratings in Collaborative Filtering. In: The 27th Annual International ACM SIGIR Conference, Sheffield, pp. 568–569 (2004)
6. Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM* 40(3), 77–87 (1997)
7. MovieLens, <http://www.movielens.umn.edu/>
8. Netflix prize, <http://www.netflixprize.com>
9. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: Proceedings of ACM Conference on Computer Supported Cooperative Work, pp. 175–186. Chapel Hill (1994)
10. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: WWW 2001: Proceedings of the 10th international conference on World Wide Web, pp. 285–295. ACM, New York (2001)

On Three Classes of Division Queries Involving Ordinal Preferences

Patrick Bosc, Olivier Pivert, and Olivier Soufflet

Irisa – Enssat, University of Rennes 1
Technopole Anticipa 22305 Lannion Cedex France
bosc@enssat.fr, pivert@enssat.fr, soufflet@enssat.fr

Abstract. In this paper, we are interested in taking preferences into account for a family of queries inspired by the relational division. A division query aims at retrieving the elements associated with a specified set of values and usually the results remain not discriminated. So, we suggest the introduction of preferences inside such queries with the following specificities: i) the user gives his/her preferences in an ordinal way and ii) the preferences apply to the divisor which is defined as a hierarchy of sets. Different uses of the hierarchy are investigated, which leads to queries conveying different semantics and the property of the result in terms of a quotient is studied. A special attention is paid to the implementation of such queries using a regular database management system and some experimental results illustrate the feasibility of the approach.

1 Introduction

Queries including preferences have received a growing interest during the last decade [1,3,4,5,6,8,9,10]. One of their main advantages is to allow for some discrimination among the elements of their result (which is no longer a flat set) thanks to the compliance with the specified preferences. However, up to now, most of the research works have focused on fairly simple queries where preferences apply only to selections. The objective of this paper is to enlarge the scope of preference queries by considering more complex ones, founded on the association of an element with a given set of values, in the spirit of the division operation. Moreover, a purely ordinal framework is chosen and the user has only to deal with an ordinal scale, which we think to be not too demanding. Lastly, taking preferences into account will allow for keeping only the best k answers, in the spirit of top- k queries [4]. Knowing that a regular division delivers a non discriminated set of elements, the idea is here to call on preferences related to the divisor. Two major lines for assigning preferences may be thought of, depending on whether they concern tuples individually (see e.g., [3]), or (sub)sets of tuples, which is the choice made in this paper and we will use the words “stratified divisor/division”. Consequently, an element x of the dividend will be all the more acceptable as it is connected with a large number of the subsets (S_i ’s) defined over the divisor. In fact, different roles can be allotted to the divisor when it is described as a hierarchical set. Three of them, which seem to be useful and natural are envisaged:

1. a direct extension of the division in a conjunctive way, where the first layer of the divisor is mandatory and the following ones are considered only desirable: find the elements x connected with S_1 *and if possible ... and if possible* S_n (which has some relationship with bipolarity [7]),
2. a disjunctive view where x is all the more satisfactory as it is connected with all the values of a highly preferred (sub)set of the divisor: find the elements x connected with S_1 *or else ... or else* S_n ,
3. an intermediate approach where x is all the more highly ranked as it is connected with numerous and preferred (sub)sets of the divisor: find the elements x connected with S_1 *and-or ... and-or* S_n .

As an example, let us consider the case of a user looking for wine shops offering Saint Emilion Grand Cru, Pomerol and Margaux and if possible Gewurztraminer Vendanges Tardives and Chablis Premier Cru and if possible Pommard and Chambertin. Of course, “or else” or “and-or” could be used as well. The rest of the paper is organized as follows. Section 2 is devoted to a presentation (in terms of syntax and semantics) of the three types of queries considered. In section 3, it is shown that the result returned in each case can be characterized as a quotient, i.e., a largest relation according to a given inclusion constraint. Implementation issues are discussed in section 4 and some experiments are reported in order to assess the performances of such queries using a commercially available DBMS.

2 Reminders and Syntax

2.1 Some Reminders on the Division

In the rest of the paper, the dividend relation r has the schema (A, X) , while that of the divisor relation s is (B) where A and B are compatible sets of attributes. The division of relation r by relation s is defined as:

$$\text{div}(r, s, A, B) = \{x \mid x \in r[X] \wedge s \subseteq \Omega_r(x)\} \quad (1)$$

$$= \{x \mid x \in r[X] \wedge \forall a, a \in s \Rightarrow (a, x) \in r\} \quad (2)$$

where $r[X]$ denotes the projection of r over X and $\Omega_r(x) = \{a \mid (a, x) \in r\}$. In other words, an element x belongs to the result of the division of r by s iff it is associated in r with at least all the values a appearing in s . The justification of the term “division” assigned to this operation relies on the fact that a property similar to that of the quotient of integers holds. Indeed, the resulting relation res obtained with expression (1) has the double characteristic:

$$\forall t \in \text{res}, s \times \{t\} \subseteq r \quad (3a)$$

$$\forall t \notin \text{res}, s \times \{t\} \not\subseteq r \quad (3b)$$

\times denoting the Cartesian product of relations.

Expressions (3a) and (3b) express the fact that relation *res* is a quotient, i.e., the largest relation whose Cartesian product with the divisor returns a result which is included in the dividend.

Example 1. Let us take a database involving the two relations order (*o*) and product (*p*) with respective schemas $O(np, store, qty)$ and $P(np, price)$. Tuples (n, s, q) of *o* and (n, pr) of *p* state that product *n* has been ordered from store *s* in quantity *q* and that its price is *pr*. Retrieving the stores which have been ordered all the products priced under \$127 in a quantity greater than 35, can be expressed thanks to a division as: $div(o-g35, p-u127, np, np)$ where relation *o-g35* corresponds to pairs (n, s) such that product *n* has been ordered from store *s* in a quantity over 35 and relation *p-u127* gathers products whose price is under \$127. From the extensions of relations *o-g35* and *p-u127* given hereafter:

$$\begin{aligned}
 o-g35 &= \{(p_{15}, s_{32}), (p_{12}, s_{32}), (p_{34}, s_{32}), (p_{26}, s_{32}), \\
 &\quad (p_{12}, s_7), (p_{26}, s_7), (p_{15}, s_{19}), (p_{12}, s_{19}), (p_{26}, s_{19})\} \\
 p-u127 &= \{p_{15}, p_{12}, p_{26}\}
 \end{aligned}$$

the division returns $\{s_{32}, s_{19}\}$, which satisfies (3a) and (3b). ◇

2.2 General Syntactical Framework

The three types of queries studied later are expressed in an SQL-like style where the dividend may be any intermediate relation and the stratified divisor is either explicitly given by the user (case which will be considered further) or results from a series of subqueries. Usually, the division of relation *r* of schema $R(A, X)$ by relation *s* of schema $S(B)$ is expressed thanks to a partitioning mechanism and we suggest a similar expression here:

select top *k* X from r [where condition] group by X
having set(A) contains $\{v_{1,1}, \dots, v_{1,j_1}\}$ *connector* ... *connector* $\{v_{n,1}, \dots, v_{n,j_n}\}$

where *connector* stands for either “and if possible”, or “or else”, or “and-or”. Such a statement induces an ordering over the divisor, namely $(S_1 = \{v_{1,1}, \dots, v_{1,j_1}\}) \succ \dots \succ (S_n = \{v_{n,1}, \dots, v_{n,j_n}\})$ where $a \succ b$ denotes the preference of *a* over *b*. Associated with this preference relation is an ordinal scale *L* with labels l_i ’s (such that $l_1 > \dots > l_n > l_{n+1}$) which will be used to assign levels of satisfaction to elements pertaining to the result of a stratified division (l_1 corresponds to the highest satisfaction and l_{n+1} expresses rejection). Coming back to the context evoked in the introduction, an example of such a query could be:

select top 6 shop-name from wineshops group by shop-name
having set(wine) contains {Saint Emilion Grand Cru, Pomerol, Margaux}
and if possible {Gewurztraminer Vendanges Tardives, Chablis Premier Cru}
and if possible {Pommard, Chambertin}

along with the scale $L = l_1 > l_2 > l_3 > l_4$.

3 Stratified Division-Like Queries

The three types of queries called Q1, Q2 and Q3 are the following:

- find the best k elements associated with S_1 and if possible ... and if possible S_n (Q1),
- find the best k elements associated with S_1 or else ... or else S_n (Q2),
- find the best k elements associated with S_1 and-or ... and-or S_n (Q3).

3.1 Conjunctive Queries (Q1)

As to Q1 queries, to be qualified, an element x must be connected with all the elements having the maximal importance (S_1). In addition, as soon as it is not connected with all the elements of a set S_k , its association with values of any set S_{k+p} does not intervene for its final ranking. An element x is all the more preferred as it is associated with all the values of the succession of sets S_1 to S_i where i is large (if possible n for “perfection”). In other words, x is preferred to y if x is associated with all the values of the sets S_1 to S_p and y is associated with a shorter list of sets. More formally, let us denote:

$$I(x) = \{i \mid S_i \not\subseteq \Omega_r(x)\} \text{ and } imin(x) = \min(I(x)).$$

The grade of satisfaction $sat(x)$ obtained by an element x is expressed thanks to the scale L (implicitly) provided by the user as follows:

$$sat(x) = l_1 \text{ if } I(x) = \emptyset, l_{n+2-imin(x)} \text{ otherwise.} \tag{4}$$

So doing, the satisfaction is seen as a composition of the results of the division of the dividend with each of the layers of the divisor.

Remark. Due to space limitation, we cannot develop the fact that the grade $sat(x)$ may also be expressed thanks to a formula generalizing (2) where the level attached to every value of the divisor is taken into account and the material implication is extended in order to work on elements of an ordinal scale.

Example 2. Let us take the divisor: $\{a, b\} \succ c \succ \{d, e\}$ and the dividend relation:

$$r = \{(a, x_1), (b, x_1), (c, x_1), (d, x_1), (a, x_2), (b, x_2), (a, x_3), (b, x_3), (d, x_3), (e, x_3), (e, x_4), (b, x_5), (d, x_5)\}.$$

One has: $n = 3, I(x_1) = \{3\}, imin(x_1) = 3$ and $sat(x_1) = l_2$; similarly, $sat(x_2) = sat(x_3) = l_3, sat(x_4) = sat(x_5) = l_4$ and the final result is: $x_1 \succ \{x_2, x_3\}. \diamond$

3.2 Disjunctive Queries (Q2)

While Q1 queries have a conjunctive behavior, Q2 queries are meant disjunctive instead, and S_1 is no longer a mandatory subset. Here, the order of the subsets

according to user’s preferences is used so that an element x is all the more preferred as it is connected with all the values of S_k and k is small (ideally 1 for “perfection”). In this case again, the associations with the subsets of higher index ($> k$), and then lower importance, do not play any role in the discrimination strategy. In other words, x is preferred to y if x is associated with all the values of the set S_k (and no S_j with $j < k$) and y is associated with S_p (and no S_m with $m < p$) and $p > k$. Let us denote:

$$I'(x) = \{i \mid S_i \subseteq \Omega_r(x)\} \text{ and } ipmin(x) = \min(I'(x)).$$

The grade of satisfaction attached to an element x is expressed as:

$$sat(x) = l_{n+1} \text{ if } I'(x) = \emptyset, l_{ipmin(x)} \text{ otherwise.} \tag{5}$$

The satisfaction is still a combination of the results of the division of the dividend with each of the layers of the divisor.

Example 3. Let us take the divisor: $\{a, b\} \succ c \succ \{d, e\}$ and the dividend:

$$r = \{ (a, x_1) (d, x_1) (e, x_1) (c, x_2) (d, x_2) (e, x_2) \\ (a, x_3) (b, x_3) (c, x_3) (d, x_3) (e, x_3) (c, x_4) (b, x_5) \}.$$

One has: $n = 3, I'(x_2) = \{2, 3\}, ipmin(x_2) = 2$ and $sat(x_2) = l_2$; similarly, $sat(x_1) = l_3, sat(x_3) = l_1, sat(x_4) = l_2, sat(x_5) = l_4$ and then the final result is: $x_3 \succ \{x_2, x_4\} \succ x_1$. ◊

3.3 Full Discrimination-Based Queries (Q3)

Queries of type Q3 are designed so as to counter the common disability of Q1 and Q2 queries in distinguishing between elements which are equally ranked because additional associations are not taken into account (e.g., x_2 and x_4 in the above example). So, the principle for interpreting Q3 queries is to consider all the layers for which a complete association occurs. An element is all the more preferred as it is associated with a set S_i highly preferred and this same point of view applies to break ties. In this case, the grade of satisfaction for x may be expressed thanks to a vector $V(x)$ of dimension n where $V_i(x) = 1$ if x is associated with all the values of $S_i, 0$ otherwise. Ordering elements boils down to comparing such vectors according to the lexicographic order ($>_{lex}$):

$$x >_{lex} y \Leftrightarrow \exists k \in [1, n] \text{ such that } \forall j < k, V_j(x) = V_j(y) \text{ and } V_k(x) > V_k(y).$$

In this view, the scale L is not used directly even if the order of the elements of the vectors reflects it in the sense that, if $i < j, V_i(x)$ is more important than $V_j(x)$ as $l_i > l_j$. It is however possible to use a scale to perform the comparison of elements in the context of Q3 queries, even if this scale is not the initial one and

is much larger (2^n levels instead of $(n + 1)$ in the original one). Let us consider the function which maps a vector V into an integer score s as follows:

$$sat(x) = \sum_{i=1..n} V_i(x) * 2^{n-i}. \tag{6}$$

It is straightforward to prove that the preference of x over y as defined before is equivalent to $sat(x) > sat(y)$. In addition, it turns out that dealing with such scores is easier than comparing vectors from a calculus point of view.

Example 4. Let us take the divisor: $\{a, b\} \succ c \succ \{d, e\}$ and the dividend:

$$r = \{ (a, x_1) (d, x_1) (e, x_1) (c, x_2) (d, x_2) (e, x_2) \\ (a, x_3) (b, x_3) (d, x_3) (c, x_4) (b, x_5) \}.$$

One has: $V(x_1) = (0, 0, 1)$, $V(x_2) = (0, 1, 1)$, $V(x_3) = (1, 0, 0)$, $V(x_4) = (0, 1, 0)$, $V(x_5) = (0, 0, 0)$, $sat(x_1) = 1$, $sat(x_2) = 3$, $sat(x_3) = 4$, $sat(x_4) = 2$, $sat(x_5) = 0$ and then the final result is: $x_3 \succ x_2 \succ x_4 \succ x_1$. \diamond

3.4 Relationship with the Lexicographic Order

As it has been indicated, Q3 queries potentially use all the layers of the divisor in order to discriminate between the elements returned by a query and is founded on the lexicographic order. It turns out that the other two types of queries can also be situated in this setting. Interpreting any Q1 or Q2 query can be done through a vector accounting for the association with the values of each complete stratum of the divisor, as it is done for Q3 queries. For Q1 queries:

$$sat(x) = l_{n-k+2} \text{ where } k \text{ is the smallest indice s.t. } V_k(x) = 0 \text{ (} n+1 \text{ if none)}.$$

In other words, the comparison of x and y can be based on a modified vector V' obtained from V by propagating to the right the first 0, which yields the equivalence $(sat(x) > sat(y)) \Leftrightarrow (V'(x) >_{lex} V'(y))$. Similarly, for Q2 queries:

$$sat(x) = l_k \text{ where } k \text{ is the smallest indice s.t. } V_k(x) = 1 \text{ (} k = n+1 \text{ if none)}.$$

Here also, the comparison of x and y can be based on a modified vector V'' obtained from V by propagating to the right the first 1, which yields the equivalence $(sat(x) > sat(y)) \Leftrightarrow (V''(x) >_{lex} V''(y))$.

Example 5. Let us come back to the data of example 4, in particular the vectors V obtained. From them, we obtain the modified vectors: $V'(x_1) = (0, 0, 0)$, $V'(x_2) = (0, 0, 0)$, $V'(x_3) = (1, 0, 0)$, $V'(x_4) = (0, 0, 0)$, $V'(x_5) = (0, 0, 0)$ which leads to keeping only x_3 as the result of query Q1. Similarly, the modified vectors V'' are: $V''(x_1) = (0, 0, 1)$, $V''(x_2) = (0, 1, 1)$, $V''(x_3) = (1, 1, 1)$, $V''(x_4) = (0, 1, 1)$, $V''(x_5) = (0, 0, 0)$ and the result of Q2 is: $x_3 \succ \{x_2, x_4\} \succ x_1$. These two results are exactly those obtained with formulas (4) and (5). \diamond

4 Property of Quotient of the Result Delivered

We now show that the result delivered by the three types of queries is a quotient. For space reasons, formal proofs are omitted but the demonstrations of the characterization formulas that we establish are rather straightforward. Here, we have of course to consider the satisfaction level (l_i) assigned to an element x of the result. For Q1 queries, if it is assumed that x is assigned the grade l_i ($i \in [1, n]$) the following property holds:

$$\forall k \in [1, n-i+1], S_k \times \{x\} \subseteq r \tag{7a}$$

$$S_{n-i+2} \times \{x\} \not\subseteq r. \tag{7b}$$

In addition, any value x which is not (at all) in the result (grade of satisfaction l_{n+1}) is such that: $S_1 \times \{x\} \not\subseteq r$, which expresses that its grade cannot even be increased from l_{n+1} to l_n .

Example 6. Let us take the data of example 2 where $n = 3$ and $sat(x_1) = l_2$. One has $S_1 \times \{x_1\} = \{(a, x_1), (b, x_1)\} \subseteq r$, and $S_2 \times \{x_1\} = \{(c, x_1)\} \subseteq r$, whereas $S_3 \times \{x_1\} = \{(d, x_1), (e, x_1)\} \not\subseteq r$, which illustrates the validity of formulas (7a-7b). Moreover, for x_5 with $sat(x_5) = l_4$, it can be observed that $S_1 \times \{x_5\}$ is not included in r . \diamond

Similarly, for Q2 queries, if x got the grade of satisfaction l_i one has:

$$S_i \times \{x\} \subseteq r \tag{8a}$$

$$\forall k \in [1, i-1], S_k \times \{x\} \not\subseteq r. \tag{8b}$$

Example 7. Let us take the data of example 3 where $n = 3$ and $sat(x_1) = l_3$. One has $S_3 \times \{x_1\} = \{(d, x_1), (e, x_1)\} \subseteq r$, whereas both $S_2 \times \{x_1\} = \{(c, x_1)\} \not\subseteq r$ and $S_1 \times \{x_1\} = \{(a, x_1), (b, x_1)\} \not\subseteq r$, and formulas (8a-8b) hold. \diamond

As to Q3 queries, recall that the grade of satisfaction of x is basically expressed as a function of the values of the vector V stating whether x is connected or not with all the values of the different layers of the divisor (formula 6). The following property holds:

$$\forall i \in [1, n] \text{ such that } V_i(x) = 1, S_i \times \{x\} \subseteq r, \tag{9a}$$

$$\forall i \in [1, n] \text{ such that } V_i(x) = 0, S_i \times \{x\} \not\subseteq r, \tag{9b}$$

which means that if the grade $sat(x)$ is increased, some inclusion constraint(s) of type 9a will be violated.

Example 8. Let us come back to example 4 where $V(x_3) = (1, 0, 0)$ and $sat(x_3) = 4$. One has $S_1 \times \{x_3\} \subseteq r$, $S_2 \times \{x_3\} \not\subseteq r$, $S_3 \times \{x_3\} \not\subseteq r$ which prevents from any increase in the value of $sat(x_3)$ according to formula (9a). \diamond

5 Implementation and Experimental Results

In this section, we describe how queries Q1-Q3 can be implemented using a commercial DBMS. Moreover, we report some experiments made in order to assess the extra cost induced by the handling of preferences in division queries.

5.1 Principles of the Algorithms

The general principle retained for implementing the previous queries is to use regular SQL queries for accessing the data, embedded in programs in charge notably of computing the grade of satisfaction (denoted by *sat* in the algorithm hereafter) assigned to each element of the result. The algorithms proposed have the following two characteristics: i) they are inspired by the usual way of expressing a division query by means of a counting, and ii) one takes advantage of the stratification of the divisor so as to first access certain tuples of the dividend. In the algorithm hereafter, *specific-condition* and *specific-conclusion* depend on the type of query under consideration (Q1, Q2 or Q3). The layers are scanned in decreasing order of importance (S_1 to S_n), which really matters only for Q1 and Q2 queries. Moreover, *specific-condition* and *specific-conclusion* are so that for a Q1 (resp. Q2) query, the loop stops as soon as an exhaustive association does not hold (resp. holds) and the grade of satisfaction is computed accordingly. For a Q3 query, all the layers must be examined.

```

declare c1 cursor for select distinct X from r;
open c1; fetch c1 into :x;
while not end-of-c1 do
  i := 1;
  while specific-condition do
    select count(*) into :nb from r where X = :x and A in  $S_i$ ;
    if :nb = card( $S_i$ ) then specific-conclusion endif; i := i + 1;
  endwhile;
  if appropriate then res := res + sat/x endif;
  fetch c1 into :x;
endwhile;
close c1;

```

5.2 Experiments

The final objective of the experiments is to assess the extra cost to pay when dealing with preference division queries. Queries Q1-Q3 are evaluated with dividend relations of different sizes (300, 3000 and 30000 tuples), with and without index, with a same divisor made of five layers involving 3, 2, 1, 2 and 2 values. A reference query is taken, namely a division without preferences where the divisor is made of the ten tuples of the five layers of the previous stratified divisor. The DBMS used is OracleTM8.0 with a 2-processor AlphaTM server 4000 and a 1.5 Gb main memory. The results obtained are gathered in Table 1, knowing that:

- we used synthetic data and the selectivity of each value a from the divisor relatively to any x from the dividend was set to 75%,
- each algorithm is run 8 times in order to avoid any bias,
- the results reported in the case of an index concern an index on X in r ,
- the size of the result is 2 (resp. 15, 234) for a dividend of 300 (resp. 3000, 30000) tuples,
- the time unit corresponds to 1/60 s.

Table 1. Experimental results

size (dividend)	300		3000		30000	
	no idx	idx	no idx	idx	no idx	idx
Reference query	17	16	267	144	16930	1556
Query Q1	24	22	529	221	30948	2231
Query Q2	28	25	447	220	30012	2266
Query Q3	55	49	1074	480	74471	4754

The analysis of these results leads to the following three main comments:

- in the absence of an index, the cost of all the queries is non linear (in the size of the dividend relation r), while on the contrary, it is linear when r is indexed on X . This is clearly due to the construction of the cursor ($c1$) which requires a sort of relation r when no index on X is available;
- the cost of queries Q1 and Q2 are roughly the same and in the range of one half of that of query Q3. Unsurprisingly, this means that for Q1 and Q2, in the average, for a given value x of the dividend the SQL query “select count(*) ...” is performed for one half of the layers, while this query is run for all of the layers for Q3;
- in the presence (resp. the absence) of an index, the extra cost induced by preferences is in the range of 40-50% (resp. 50-100%) for queries Q1 and Q2, and roughly 200% (resp. 300%) for queries Q3. This is not negligible, but this is clearly more acceptable with an index on X .

These first results are encouraging, even if they must be completed to reach definitive conclusions as to the way of developing tools on top of existing DBMS’s for processing such queries.

6 Conclusion

In this paper, preferences for a family of queries stemming from the relational division have been considered. The key idea is to use a divisor made of a hierarchy of subsets of elements. So doing, the result is no longer a flat set but a list of items provided with a level of satisfaction. Three uses of the hierarchy have been investigated, which led to three fairly distinct semantics of the corresponding queries. Moreover, it has been shown that the result delivered in all cases is a

quotient. A special attention has been paid to the implementation of such queries using a regular DBMS. Some experimental results illustrate the feasibility of the approach and tend to prove that the extra cost induced by the handling of preferences can be kept acceptable, if not marginal. Complementary experiments should be undertaken, in particular:

- using dividend and divisor relations of larger sizes,
- founded on other algorithms implementing queries Q1, Q2 and Q3. In particular, one might study what happens with a solution where one regular division query per stratum is submitted to the DBMS,
- replacing Oracle by another system, e.g., MySQL, to observe the stability/variability of the tendency of the measures.

Beyond the experimental aspect, this work opens other perspectives, among which introducing disjunctions in the divisor and investigating anti-division queries [2] (looking for elements connected with none of the values of the divisor in the regular case) in the presence of a stratified divisor.

References

1. Börzsönyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: Proc. of the 17th IEEE Inter. Conf. on Data Engineering, pp. 421–430 (2001)
2. Bosc, P., Pivert, O.: On a parameterized antidi-division operator for database flexible querying. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 652–659. Springer, Heidelberg (2008)
3. Bosc, P., Pivert, O., Rocacher, D.: About quotient and division of crisp and fuzzy relations. *Journal of Intelligent Information Systems* 29(2), 185–210 (2007)
4. Bruno, N., Chaudhuri, S., Gravano, L.: Top-k selection queries over relational databases: mapping strategies and performance evaluation. *ACM Transactions on Database Systems* 27(2), 153–187 (2002)
5. Chomicki, J.: Preference formulas in relational queries. *ACM Transactions on Database Systems* 28(4), 427–466 (2003)
6. Dubois, D., Prade, H.: Using fuzzy sets in flexible querying: why and how. In: Proc. of the Workshop on Flexible Query-Answering Systems (FQAS 1996), pp. 89–103 (1996)
7. Dubois, D., Prade, H.: Handling bipolar queries in fuzzy information processing. In: *Handbook of Research on Fuzzy Information Processing in Databases*, pp. 97–114. IGI Global Publication (2008)
8. Hadjali, A., Kaci, S., Prade, H.: Database preference queries — a possibilistic logic approach with symbolic priorities. In: Hartmann, S., Kern-Isberner, G. (eds.) FoIKS 2008. LNCS, vol. 4932, pp. 291–310. Springer, Heidelberg (2008)
9. Kießling, W., Köstler, G.: Preference SQL — design, implementation, experiences. In: Proc. of the 28th Conference on Very Large Data Bases (VLDB 2002), pp. 990–1001 (2002)
10. Lacroix, M., Lavency, P.: Preferences: putting more knowledge into queries. In: Proc. of the 13th Conference on Very Large Data Bases (VLDB 1987), pp. 217–225 (1987)

Analyses of Knowledge Creation Processes Based on Different Types of Monitored Data

Ján Paralič¹, František Babič¹, Jozef Wagner¹,
Ekaterina Simonenko², Nicolas Spyratos², and Tsuyoshi Sugibuchi²

¹ Centre for Information Technologies, Technical University Košice, Slovakia
{jan.paralic, frantisek.babic, jozef.wagner}@tuke.sk

² Universite Paris-Sud, France
{ekaterina.simonenko, nicolas.spyratos, tsuyoshi.sugibuchi}@lri.fr

Abstract. This paper presents specialized methods for analyzing knowledge creation processes and knowledge practices that are (at least partially) projected in work within a virtual collaborative environment. Support for such analysis and evaluation of knowledge creation processes is provided by historical data stored in a virtual working environment, and describes various aspects of the monitored processes (e.g. semantic information, content, log of activities, etc.). The proposed analytical methods cover different types of analysis, such as (a) statistic analysis, that provides information about processes, and the possibility to visualize such information based on user-selected presentation modes; (b) time-line based analysis that supports visualization of the real process execution with all relevant information, including the possibility to identify and further analyze working patterns in knowledge creation processes (projection of knowledge practices). Experimental evaluation of the proposed methods is carried out within the IST EU project called KP-Lab.

Keywords: Knowledge creation process, knowledge practices, event log, visual analysis, working patterns.

1 Motivation

Knowledge practices [1] constitute an interesting topic of analysis of advanced working processes, especially knowledge creation processes [5] that produce new, potentially useful knowledge. Intelligent systems are being designed and implemented for the effective support of this type of processes [11]. Providing suitable means for reflection on knowledge practices and suitable analysis of them is a real challenge.

Especially in the area of business processes several theories and approaches have been proposed, and applications have been designed and implemented to support the creation, execution and evaluation of a process model [12]. Moreover, tools are available today for the analysis of business processes [2]. The main problem here is that the analysis of business processes is geared today mainly towards orchestration issues and/or performance improvements. The focus in knowledge creation processes is different. The core issue here is the identification, comparison and analysis of knowledge practices projected into the work supported by an intelligent information system.

This paper presents a proposal of analytical features that have been designed and implemented with the aim to efficiently support such kind of analysis. The proposed approach is currently being tested and evaluated within the KP-Lab Project, an IST EU funded project.

In the remaining of the paper, section 2 describes in detail our analytical approaches as well as their evaluation and section 3 offers some concluding remarks.

2 Proposed Analytical Approach

Knowledge practices, both in educational as well as in professional settings, frequently contain some predefined goals and are based on collaboration between all included participants, using relevant resources and useful tools [5]. The whole process is monitored and all the performed actions and modifications (collectively referred to as *events*) are stored into various repositories to provide additional and important information about completed and/or still ongoing processes [8].

The proposed analytical approaches yield two different methods for analysis of knowledge practices. The first (so called *visual analyzer*) is based on an intelligent approach to interactively presenting aggregated data about activities logged within the system or added manually about related external activities; and the second provides features for *time-line based analyses*. Both approaches have been designed and currently being implemented within the knowledge practices environment [5]. An important aspect of the design process was the set of expectations and requirements from potential users and pedagogical researchers.

The sources of data for the proposed analyses are coming from different types of repositories:

- the *knowledge repository*, which contains semantic information about monitored knowledge processes and particular process elements (e.g. shared objects, also called *artifacts*);
- the *content repository*, which includes the content part of shared objects (e.g. documents, video or sound files, etc.);
- the *awareness repository*, which stores logs of activities performed in the knowledge processes within the information system [8];
- and the *user database*, which contains the user profiles.

2.1 Related Work

Research in related areas has produced several existing approaches to analysis of processes that are briefly described in this section; but to our knowledge there is no particular approach focused on knowledge creation processes, trying to analyze the employed knowledge practices.

Process mining provides functionalities for extraction of potentially useful information from event logs. Logs of events are results of monitored activities over the performed processes, especially business processes, but we believe that it has potential to cover different types of processes as well. Processes in this case are represented as workflows. The process analysis consists of several phases: creation of planned process model; monitoring of performed events; creation of actual process model

based on logged events; analyses of acquired model as identification of deviations in process structure; description of causal dependencies between process elements; particular activities represented by events, performance statistics, etc. The work in ProM [2] represents a generic open-source framework for implementing process mining tools in a standard environment. The ProM framework receives as input logs in the Mining XML format (MXML). Currently, this framework has plug-ins for process mining, analysis, monitoring and conversion.

The IST EU project called Super¹ (Semantics Utilized for Process management within and between Enterprises) [4] provides features for semantic business analysis based on utilization of semantic information such as ontologies and semantic annotation. Semantic annotations are used as markers to identify problematic points and violations. Ontologies provide a framework that comprises the relevant concepts for event description.

A framework for analyses and visualizations of collaborative processes was designed within the Kaleidoscope project². This framework, called CAViCoLA (Computer-based Analysis and Visualization of Collaborative Learning Activities) provides functionalities to identify existing complex interactions within examined processes [7]. The analysis results are visualized in an appropriate graphical format that enables users to make their own interpretations and allows them to reflect on their previous activities.

OLAP (Online Analytical Processing) methods are used in analyses of processes in [6]. This approach offers the possibility to create data cubes based on data that describe modeled business processes. Using such data cubes several visual analyses and interactive explorations can be executed.

An interesting approach to analyze interactive processes in collaborative environments is represented by a methodology based on Social Network Analyses [3]. The logs describe events in a web-based system oriented toward collaborative processes with shared documents. It is possible to identify social structures, knowledge building processes and interesting relations or interactions.

The analysis of knowledge creation processes is mainly represented by social or cognitive methods. This means that participants in such processes analyze performed activities based on their experiences and knowledge background with utilization of suitable applications. Our approaches, described in the following subsections, are focused on analysis of particular types of processes that lead to creation, identification or acquisition of new knowledge (such as collaborative creation of scientific articles or productive inquiry way of teaching special courses etc.). The visual analyzer assists users to define various (statistical) queries and to visualize the results in the form of summarized data, in some suitable visual form of presentation. Timeline-based analysis of knowledge creation processes provides researchers with a graphical interactive user interface enabling them to view; inspect and code a selected group of activities from integrated repositories on a timeline. Moreover, new external events may be added, and subsets of activities can be grouped to form patterns. This concept is unique in the area of knowledge creation process analysis.

¹ <http://www.ip-super.org/>

² <http://www.noe-kaleidoscope.org/pub/>

2.2 Visual Analyzer

The visual analyzer is a tool for simplifying statistical query formulation and for highlighting patterns, outliers and gaps of the result, thus helping to identify important individuals, relationships and tasks. These activities lie in the field of visual analytics. The basic idea of visual analytics is to integrate statistical results and information visualization in one tool, allowing humans to directly interact with information, generate overviews and draw conclusions (see [13] for the issues and challenges). The proposed visual analyzer's difference from other similar tools is in its underlying model, the description of which lies outside the scope of this paper (see [9, 11] for more details).

The visual analyzer provides (a) summarized information about performed users' activities based on user defined conditions and (b) suitable visualizations of the summarized information in a form selected by the user (bar chart, pie chart, etc.). Here, by "summarized information" we mean various aggregations of available data about some objects of interest. For example, if the objects of interest are events, then we might be interested in knowing the number of participants, by event; or the number of participants, by event and date; or the number of shared objects used, by event; or the number of comments added, by event and date; and so on. This kind of information is very useful for maintaining the knowledge processes and analyzing them. The implementation of visual analyzer involves several steps:

- The activity logs are listed in some predefined format (e.g. in the form of relational tables).
- A graphical or *visual schema* is extracted from the log format and presented to the user.
- The user has the possibility to create an analytical query visually, based on the visual schema.
- A query, once defined on the visual schema, is translated to executable form and is executed on the stored data.
- The user can select a presentation form from a pop-up menu in the visual schema (e.g. bar chart, pie-chart, etc.)

Visualization of query results allows for quick, intuitive ingestion of the returned information; and as mentioned earlier, summarized information can be extremely useful for different purposes: identification of division of work; identification of most active persons; identification of well collaborating groups of people; and so on. In the remaining of this subsection we first explain (through an example) how visual interaction between the user and the visual analyzer is performed, and then we present the basic implementation components of the visual analyzer.

Visual Interaction

In the current implementation of the visual analyzer, the input is assumed to be a table, called *LOG*, with one or more keys, as in the following example: *LOG(EventId, GroupId, Time, SubjectId, Subject Type, Object Id, Object Type, Action)*.

The visual schema is constructed from the *LOG* table as follows (see Fig. 1).

1. A key K is selected as representing the objects described by the table *LOG*, and it is connected by an arrow $K \rightarrow A$ to every attribute A in *LOG*; in Fig. 1, the attribute *EventId* (i.e. “event identifier”) is the selected key.
2. For each attribute A , a number of indicators is selected, depending on A (e.g. if A is the attribute *Time*, we might select *Day*, *Month* and *Year* as indicators; if A is *SubjectId* we might select *SubjectType* as an indicator, etc.); the indicators of each attribute A are organized in a hierarchy rooted in A .
3. The graph resulting from the previous two steps is an acyclic graph with K as its single root (see [10] for more details); the immediate successors of K are all attributes of *LOG* other than K , and the remaining nodes are the indicators. We refer to this graph as the *visual schema*. The visual schema is implemented so that all its nodes are clickable.

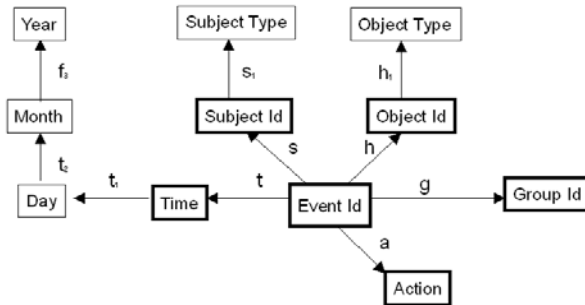


Fig. 1. Visual Schema of the LOG table

The visual analyzer allows users to define analytic queries visually (by clicking on nodes of the visual schema) and to select a mode of presentation of the query result. For example, suppose we would like to ask the query “number of *Group IDs* by *Object ID* and *Day*” and to visualize the result as a bar chart. Moreover, suppose that the visual schema is shown on the screen and the user clicks on “query definition” in the menu accompanying the visual schema. Then the user is prompted by the appearance of the term “grouping attributes”, upon which the user clicks on node *ObjectID*, then on node *Day*, then press a button “end grouping”.

Next, the user is prompted by the appearance of the term “measuring attribute”, upon which the user clicks on node *Group ID* then presses a button “end measuring”. At this point, the system infers the operations applicable on the domain of the attribute *Group ID* and puts them in a pop up menu that appears on the screen.

Next, the user is prompted to click on one operation in the pop up menu (“count” in our example). At this point, the system has all the information required in order to build an SQL group-by query (which is passed on to the relational engine for evaluation against the *LOG* table). Finally, the visual analyzer shows to the user (on the screen) a menu of the available result visualizations, and the user clicks on the desired one (“bar chart”, for example).

We can summarize the above session as follows:

- *Grouping*: Click on node *Object ID*, then on node *Day*, then press the button “end grouping”
- *Measuring*: Click on node *Group ID* then press the button “end measuring”
- *Aggregation*: In the pop up menu of applicable operations, click on operation “count”
- *Result visualization*: In the menu of available visualizations click on “bar chart”.

Thus, the main interest of the visual analyzer in question is the advantage taken from the implementation of the model proposed in [9], and of the mapping of [10], allowing to present data in a user-friendly and suitable way for analysis and to formulate an analytical query as a triple $\langle \textit{grouping attributes, measuring attribute, aggregation operation} \rangle$ plus a visualization choice.

Implementation

The visual analyzer is designed as a client-server application, the client handling the visual interaction and the server managing query transformation and processing. The client of visual analyzer is designed to be a web-application, in charge of interaction with the user: presentation of the visual schema, transformation of users' clicks into an analytical query and presentation of query result graphically.

The server is designed to transform the query, formulated by the user into the format suitable for the evaluating engine, and to transform the result into a format suitable for the client, for visualization.

Fig. 2 shows the schema of interaction between the user, the client (interface), the server and the KP-Lab Awareness services.

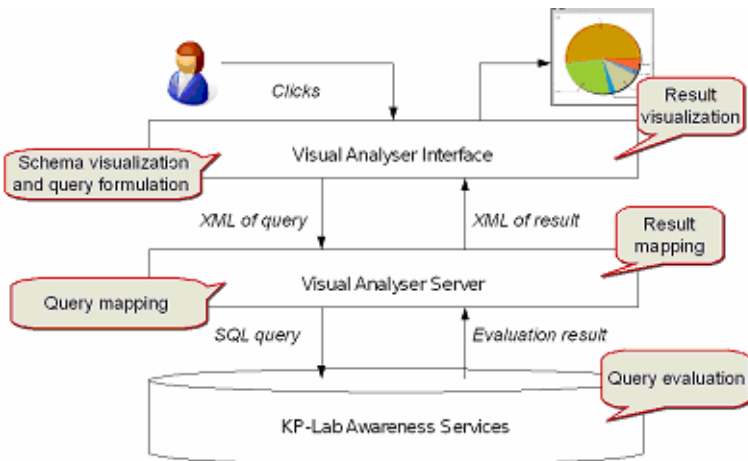


Fig. 2. Visual analyzer interaction sequence

The client is an Adobe Flex component running in a web browser, transforming the user query into an XML document, and sending it to the server. All components communicate through web-services.

The server, which is a Java program, parses the XML document of a query, into a query, formulated as a triple: $Q = \langle \textit{Grouping}, \textit{Measuring}, \textit{Aggregation} \rangle$, and maps it to an SQL query, then sends the SQL query to the KP-Lab Awareness Service for evaluation.

When the server receives the evaluation result back, the server transforms it into an XML format, convenient for visualization by the client.

2.3 Time-Line Based Analysis

Time-line based analysis contains features and methods to visualize the whole process or a particular part with relevant interactions and relations (e.g. with respect to some group, some artifact within particular time-frame). This approach produces a complex view of performed activities and gives the possibility to focus on potentially interesting facts. Information about the evolution of content artifacts provides another, completely different way to monitor ongoing processes and learn from past instances (e.g. with respect to the semantic information associated with the artefacts). This is also one way to reflect on the community's knowledge practices and continue with transformation into innovative ones.

The main functionalities provided in the time-line based analysis are the following.

- Sequences of performed activities in chronological order are visualized via time-line, see Fig. 3.
- External events that were not executed in monitored collaborative system and are relevant to the analyzed knowledge creation process can be included manually by the user.
- Visualization of all interactions and relations between selected elements.
- The important functionality of this tool is the support of coding, i.e. assigning own codes from different coding schemes to particular events (activities) presented on the timeline.
- Finally, the user or researcher is able to define and store selected patterns (pattern is a set of selected and coded events) from the timeline. These patterns are well formalized projections of interesting knowledge practices.

Fig. 3 presents a simple visualization example of the knowledge creation process in which the result is represented by the final version of a paper to be submitted to a conference. The whole process consists of five stages that are represented by changes in the virtual environment – performed activities. Each activity has a relevant version of the document and the actor of the corresponding change. Particular versions are described with semantic information; the list of presented information is selected by the user that creates the visualization.

A strong point of the proposed solution is the possibility to identify subsets of activities that may have crucial importance. These manually selected and annotated subsets of actions can be called critical or working patterns. Such patterns usually lead to some critical moments in a whole process, which can mean, for example, a significant progress, discovery of a new knowledge/approach, or in the opposite,

they may indicate failure of a particular process or its immature ending. Such kind of patterns may also (conceptually) represent interesting knowledge practices emerged within a particular knowledge creation process – either being positive (something like best practices), or negative (worst practices). These patterns are created manually by relevant users based on their findings in order to be able to discover other instances of similar type of sequence and evaluate relevant results of these instances, differences and similarities in other processes. Representation of presented patterns will be based on parameters that describe performed events in the awareness repository, i.e. a series of triples of *<action (type), subject (type), object (type)>*. A couple of similarity measures have been suggested and are being tested for their suitability in different situations. The similarity takes into account two parameters: (1) one parameter measuring how one particular event in a process matches a triple from the critical patterns definition; (2) and a second parameter measuring the position of this event within the sequence defined in the critical pattern.

In this way particular critical patterns from one process can be manually selected by the user and stored (see Fig. 3). Other users then can visualize patterns and use a pattern-matching service to find similar patterns in other processes.

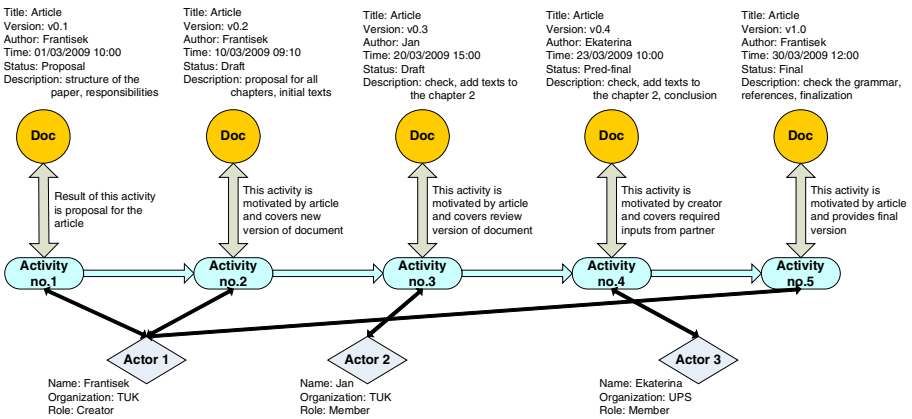


Fig. 3. Mock-up of time-line visualization

Towards that goal, it is important to identify relevant activities that led to the advancement or evolution of a particular shared object (discussion contributions, comments, linked artefacts, or changed conceptual models etc.). In these situations, the semantic context of the relevant activities will be taken into account as one dimension of analysis. This dimension is mainly covered by annotation with concepts of ontologies that are used as main semantic and communication framework for representation of knowledge creation processes.

2.4 Evaluation

The proposed analytical approach was designed and proposed based on long-term discussions with people having different knowledge backgrounds from the domain

of pedagogical research. This group of people contains mainly representatives from the University of Helsinki, the University of Utrecht and the Upper Austria University of Applied Sciences. At the beginning, they presented actual experiences with their courses, how they evaluated results, group collaboration or individual participant's contributions. Discussion about appropriate analytical solutions led to identification of several important requirements such as the possibility to create an overall view of a performed process – some type of general characteristics; possibility to visualize the whole process with relevant parts in the time – dynamic character of this type of processes; evaluation of knowledge creation processes with different types of results; identification of relations between different types of subjects and objects.

These user requirements and expectations were used for initial analyses of existing approaches and subsequent design of innovative technological solutions. Particular results of this process were discussed with potential end-users to identify both strong and problematic points. The final version that is described in this paper has been used in the implementation phase, and the first prototype can be expected before the summer of 2009. This first prototype will be evaluated within internal tests and deployed in real pilot courses within KP-Lab project, starting September 2009.

The Knowledge Practices Laboratory project³ (KP-Lab Project) aims at developing theories, tools, and practical models that enhance deliberate advancement and creation of knowledge as well as transformation of knowledge practices. The first three years of this project were devoted to the research and implementation of tools, practices and theories. The last two years will be devoted mainly to finish longitudinal experiments, dissemination activities, exploitation planning and realization of extended pilots.

The KP-Lab System provides a modular, flexible, and extensible ICT system that supports pedagogical methods to foster knowledge creation in educational and workplace settings. The system provides tools for collaborative work around shared objects, and for knowledge practices in the various settings addressed by the project. Interaction with the users is provided by a virtual user environment (KP-environment) with access to all integrated tools and functionalities. These end-user applications are built on concepts underlying the learning approach, such as collaboration, shared objects, boundary crossing, etc. The KP-environment has been implemented as a web-based environment with Flash technology in order to provide a flexible and interactive solution.

We expect the experimental evaluation of the proposed analytical approach to show the advantages and disadvantages of this solution, as well as its potential in a broader perspective. The main advantage of this solution in the domain of dissemination is its generic format of logs of events in the awareness repository, so proposed and implemented monitoring, logging and analytical services can be used for different types of collaborative or other process oriented systems.

3 Conclusion

Analysis of knowledge creation processes is an important way to identify concrete steps, critical points, critical sequences or key elements in the processes of new knowledge creation. The proposed approaches define two ways to analyze this information: visual analysis and time-line based analysis. Visual analysis provides descriptive statistics

³ <http://www.kp-lab.org/>

based on stored data about logged activities in performed processes. Time-line based analysis provides powerful tools to identify, search for, and compare interesting patterns (reflecting knowledge interesting practices). The implementation phase of these services is still in progress, but the first results of testing and evaluation can be presented during the conference.

Acknowledgments. The work presented in this paper was supported by the following projects: the EU funded integrated IST project KP-Lab (Nr. 27490); the Slovak Research and Development Agency under the contracts Nr. APVV-0391-06 and RPEU-0011-06; the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic under grants Nr. 1/4074/07 and 1/0131/09; and the French Digiteo Project VISIR (Nr 2008-33HD).

References

1. Allert, H., Richter, C.: Practices, systems, and context working as core concepts in modeling socio-technical systems. In: Proceedings of the Fifth International Workshop on Philosophy and Informatics, WSPI 2008. CEUR Workshop Proceedings, vol. 332 (2008)
2. Alves de Medeiros, A.K., et al.: Semantic Process Mining Tools: Core Building Blocks. In: Golden, W., Acton, T., Conboy, K. (eds.) 16th European Conference in Information Systems (ECIS) 2008, CD-ROM (2008) ISBN 13:978-0-9553159-2-3
3. Nurmela, K.A., Lehtinen, E., Palonen, T.: Evaluating CSCL log files by Social Network Analysis. In: Proc. CSCL 1999 Conference, pp. 434–444. Stanford University, Palo Alto (1999)
4. Celino, I., et al.: Semantic Business Process Analysis. In: Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007), June 2007. CEUR-WS, vol. 251 (2007) ISSN 1613-0073
5. Lakkala, M., et al.: Main functionalities of the Knowledge Practices Environment (KPE) affording knowledge creation practices in education. In: Proc. of CSCL 2009 (2009)
6. Mansmann, S., Neumuth, T., Scholl, M.H.: OLAP Technology for Business Process Intelligence: Challenges and Solutions. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2007. LNCS, vol. 4654, pp. 111–122. Springer, Heidelberg (2007)
7. Dimitriadis, Y.: Computer-base Analysis and Visualization of Collaborative Learning Activities (CAViCoLA). In: Kaleidoscope Symposium 2007: Defining the Scientific Evolution of Technology Enhanced Learning, Berlin (December 2007)
8. Paralic, J., Babic, F.: Support of innovative processes in a computer-based collaborative system. In: Basys 2008, Porto, Portugal, pp. 145–152. Springer, New York (2008)
9. Spyratos, N.: A Functional Model for Data Analysis. In: Larsen, H.L., Pasi, G., Ortiz-Arroyo, D., Andreasen, T., Christiansen, H. (eds.) FQAS 2006. LNCS (LNAI), vol. 4027, pp. 51–64. Springer, Heidelberg (2006)
10. Simonenko, E., Spyratos, N., Sugibuchi, T.: Data analysis based on functional dependencies. Technical report, LRI (2008)
11. Tzitzikas, Y.: Emergent knowledge artifacts for supporting dialogical e-learning. *Int. Journal of Web-Based Learning and Teaching Technologies* 2(3), 16–38 (2007)
12. Habala, O., Paralic, M., Bartalos, P., Rozinajova, V.: Semantically-aided Data-aware Service Workflow Composition. In: Nielsen, M., Kucera, A., Miltersen, P.B., Palamidessi, C., Tuma, P., Valencia, F.D. (eds.) SOFSEM 2009. LNCS, vol. 5404, pp. 317–328. Springer, Heidelberg (2009)
13. Thomas, J., Cook, K.: A Visual Analytics Agenda. *IEEE Transactions on Computer Graphics and Applications* 26(1), 12–19 (2006)

Intelligent Information Processing in Semantically Enriched Web

Pavol Návrat, Mária Bielíková, Daniela Chudá, and Viera Rozinajová

Institute of Informatics and Software Engineering,
Faculty of Informatics and Information technologies,
Slovak University of Technology, Ilkovičova 3, 842 16 Bratislava
{navrat,bielik,chuda,rozinajova}@fiit.stuba.sk

Abstract. Acquiring information from the Web is a demanding task and currently subject of a world-wide research. In this paper we focus on research of methods, and experience with development of software tools designed for retrieval, organization, presentation of information in heterogeneous data source spaces such as the Web. We see the Web as a unique evolving and unbounded information system. The presented concepts can be used also in other specific contexts of information systems in organizations that increasingly become worldwide and weaved together considering information processing.

1 Introduction

The World Wide Web allows any kind of information (of any content, but also of almost any media: text, picture, sound) possible to be “put on the Web” and be read by anyone anywhere in the world. All what we put on the Web can contain a reference to other information present on the Web. And the Web can interpret these references so that in case we want it, the referred information is accessed.

We structure the information we put on the Web into pages and pages into sites. That is of course just the technical level of structuring. By mutual references between pages regardless of whether they belong into one site or whether they belong to one author, there can be formed all sorts of connections, expressing relations between information listed on the particular pages. The described technical level of structuring gives presumptions for forming social networks of people communicating with each other through their personal computers. Such computers together with software that enables variety models of communication in some community (e.g., discussion groups) are becoming increasingly also social computers.

In this paper we rely upon our research and experience with the design of methods and development of software tools designated for acquisition, organization and presentation of information and knowledge from large data spaces employing the semantics either explicitly defined or discovered. This research was a part of the research project called NAZOU (“Tools for acquiring, organization and maintenance of knowledge in heterogeneous data sources space”,

nazou.fiit.stuba.sk) [16]. The main goal is improvement of providing current and relevant information from the Web by automatic processing.

Throughout the paper we feature examples of application domain of the mentioned project – the domain of acquisition, organization and presentation of job offers. That does not mean that the described approaches can be used exclusively for the domain of job offers. Most of the devised methods are applicable also in other domains mainly connected to information processing in organizations.

2 Related Work

The problem of information processing is the subject of intensive study worldwide [15]. Especially the idea of the Semantic Web inspired several research groups aiming at effective information processing on the Web. The Semantic Web as “a Web of actionable information – information derived from data through a semantic theory for interpreting symbols” [20] gives an opportunity to reason on documents and convert them automatically through data to information.

Since Tim Berners-Lee presented in 2001 a vision of the Semantic Web, several research projects based on this idea started. Here we can mention especially project AKT – Advanced Knowledge Technologies that has been financed by the British government (www.aktors.org), past and current projects supported by the European Union, e.g. REVERSE – Reasoning the Web (reverse.net), KP-Lab – Knowledge Practices Laboratory (www.kp-lab.org), K-Space – Knowledge Space of Semantic Inference for Automatic Annotation of Multimedia Content (www.k-space.eu), On-To-Knowledge (www.ontoknowledge.org), Knowledge Web (knowledgeweb.semanticweb.org). SIMILE – Semantic Interoperability of Metadata and Information in unLike Environments (simile.mit.edu) – joint project conducted by the MIT Libraries and MIT CSAIL covers several projects aimed at developing open source tools that empower users to access, manage, visualize and reuse the Web content.

There are more projects dealing with semantically enriched data processing in large spaces. Typically, they use ontologies as a base for metadata representation and reasoning, mostly employing RDF/OWL W3C recommendations for representation [7] and deal with issues of ontology querying as a kind of information retrieval [12]. They define new ontologies either domain dependent [6,19] or domain independent [11] and work on tools for ontology specification and maintenance [1]. Most of the projects consider a user as an important stakeholder and research or just employ techniques for personalization [5]. Most of the mentioned projects face the problem with non existing fixed data collections that would serve for experimental comparison of particular approaches to information processing in the Web (such as TREC, trec.nist.gov).

We present a concept that aims to cover the whole process of information processing in large data spaces. We provide methods for solving particular problems. Even though the methods cannot cover all aspects of the “information processing problem” they present contribution by providing a consistent chain of information processing that can be reused in several application domains.

3 From Documents to Information for the User

Coming up from the main goal, which is improvement of providing relevant information from the Web to a user in a way as much automated as possible, we focus on describing approaches in the research of new methods and tools of information processing in large data spaces. Research in this area naturally incorporates creation of models of heterogeneous environment. Data, which we have at disposal, are of uncertain nature and provide us with imperfect information. In connection with retrieving and processing of information that is relevant for the individual user in the given context we not only need data models of the content, but also user and context models.

Our approach is based on forming a procedure that involves software tools, which carry out methods for information processing, transform data acquired from the Web documents to information and knowledge, presented to the user. Software tools carry out the sequence of data acquisition and its processing to information, and thus they have to work on various levels of semantic understanding of individual data sources. For example, on the level of acquisition of data from the Web they work with partly structured text, for which they estimate its relevance with respect to the domain (e.g., whether it is a job offer) and estimate the value of individual parts of the text, or of the whole document (e.g., what is the company name, in what area the job offer is, as on the contemporary Web this information is by no means specified). When organizing the acquired offers, we use the already discovered document characteristics and we form for example a network of companies, which provide job offers in respective areas and estimating similar job offers based on that.

Information processing is accomplished in several steps, sequentially linked with each other, from selection (filtering out) and document acquisition from the Web, which comprise required data (in our case data on job offers), through identification and annotation of those documents that really comprise job offers (in order to extract required data); selection of individual job offers; their analyses and organization to their personalized presentation to the user (see Fig 1).

We transform a part of the Web to the Semantic Web. The existing documents are transformed into a representation, which describes their content using metadata. In such a way, the whole information becomes suitable for automatic processing. Naturally this process is iterative (even if its skeleton is formed as a sequence). This sequence is in fact provided many times repeatedly based on processing of atomic document or part of data. Considering individual tools that realize particular tasks they should synchronize each other to be able to process the required output for the user. The concept of the Web itself is advantageous here – data constitute a basis for tools integration.

4 Intelligent Management of Information in the Domain of Job Offers

Suppose there is a user, who (with support of web services) is looking for information about something he is interested in. Considering our domain we suppose

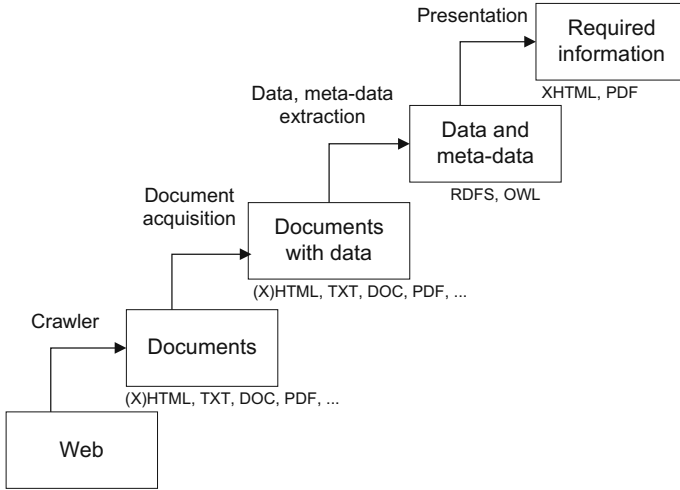


Fig. 1. Transformation of documents to information and knowledge to be presented

there is a user, who is looking for a job with a good salary, reasonable distance from his permanent address and one that corresponds to his education and abilities (and possibly other expectations). For the common problem of retrieving information, the task was to design methods for processing data so that by means of the tools realizing these methods it was possible to construct an information system, which will help the user retrieve relevant information.

Immediately some questions arise like e.g., in what form will the user communicate with the system? What will be prepared before user querying (i.e., can tools do some preprocessing that would help increase effectiveness of information processing)¹? How to prepare for potential arbitrary requirements or expectations of the user? Are we going to offer to him rather a closed system and means enabling querying based on examples or are we going to devise a system more open to individual requirements? How to settle up with continuous changes of data space (job offers originate, some change, or simply become obsolete)?

Indispensable is the phenomenon of user preferences. The concept of a good salary can for various users be different. Reasonable distance from his permanent address is also a concept, which is differently perceived by a wage-earning mother and differently by a young person, who is becoming independent. It can also happen that two different job offers are incomparable. One is better in salary parameters and the second one can be better regarding distance. An answer

¹ While designing methods for information processing we must have regard to the fact that we work with a data space that contains large number of information, more than we are able to process with contemporary means in real time. At the same time the content of the Web constantly changes, which requires a compromise between acquiring “some” relevant information.

should not only contain all (possibly also relevant, or interesting) offers for the particular user, but should be arranged according to their relevance.

That brings us to one possible view of the information system working in semantically enriched document space (see Fig. 2). We distinguish in it web sources (represented by documents, be it static or generated from the hidden Web), and components that acquire relevant documents and process them to be presented for human users. Every proposed representation of documents introduces for a large data space such as the Web a definite loss of information (as it is neither possible, nor effective or realistic to download and save entire web). However the matter is that we choose representation with a reasonable loss of the content. The tendency is even to look for mechanisms with a resulting negative loss, i.e. we enrich sources, which we acquire from the Semantic Web, (ideally) automatically amend to individual terms or parts of documents meaning by metadata.

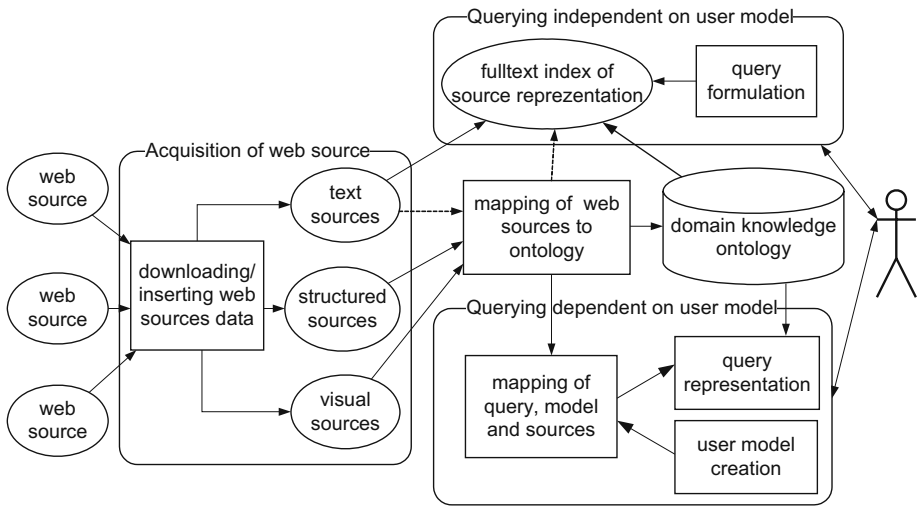


Fig. 2. Acquiring and using data from web sources [17]

Categorical knowledge of application domain – in our case job offers (represented in the Semantic Web applications often by ontologies) are necessary to provide querying on acquired data with the aim of support of search and navigation within the information space (data space of acquired from the Web is transformed to information space now).

The important component is “querying based on the user model” – a concept of personalized information presentation. Personalization can be achieved in such a manner that the recommendation depends only on the user activities (e.g., clicking while browsing) or so that it considers also additional information about the user represented in user model (such as user background) [2], or also on information about other users (employing social relations). Necessary presumption of methods for adaptation of the content and navigation is the possibility

of comparison of individual investigated and presented entities (in our case job offers or their parts in the sense of used representation).

We assume the following sequence of data processing with the goal to provide effectively relevant information (we feature on example of job offers, although the sequence is more general):

- primary documents on the Web,
- acquired documents that contain job offers,
- documents containing relevant data concerning the task of retrieving job offers,
- extracted job offers from the documents identified as relevant,
- job offers (or parts thereof) presented to a particular user or a group of users.

Implementation of the described sequence in view of the presented concept of intelligent management of information can be divided into three relatively independent tasks, which however are interlinked and mutually influence each other: (i) document and data acquisition, (ii) analysis and organization of data and information, (iii) personalized presentation of information including methods for creating and maintaining the user model.

In the following subsections we present an approach to solution of individual tasks along with methods proposed in the NAZOU research project. We evaluated methods for particular tasks and experimented with their collaboration in accomplishing the whole task using the developed software tools that are integrated using our framework for creation of adaptive portal solutions. Detailed information is presented elsewhere and summarized in two volumes of NAZOU research project workshop proceedings [17]. Fig. 3 presents variability of proposed methods and their corresponding tools developed.

4.1 Acquisition of Documents and Data from the Web

We proposed three approaches to acquisition of documents (job offers for us) and data (relevant parts of the documents) from the Web: (i) manual acquisition, (ii) automatic acquisition by browsing the Web, and (iii) automatic acquisition by downloading data from known sites that provide required information.

In case of manual acquisition of offers there is a human who fills the information base of offers. We developed special editor JOE (Job Offer Editor), which enables the providers of job offers to insert offers in such a manner that it is represented by an ontology. The point is that the huge space of the Web has to be narrowed to the particular domain. We seek a representation, which will retain all the essential information. Moreover, as we already mentioned, it will even enrich the data extracted from documents acquired from the Web by using methods of semantic annotation [14]. Annotation is a difficult task as we require machines to find information, which is often evident to a human, but definitely not to a machine [18]. Manual insertion of offers is important especially for experimenting with methods for information processing, as in such a case we can check the input data and relate them to the expected results while organizing and presenting offers [4].

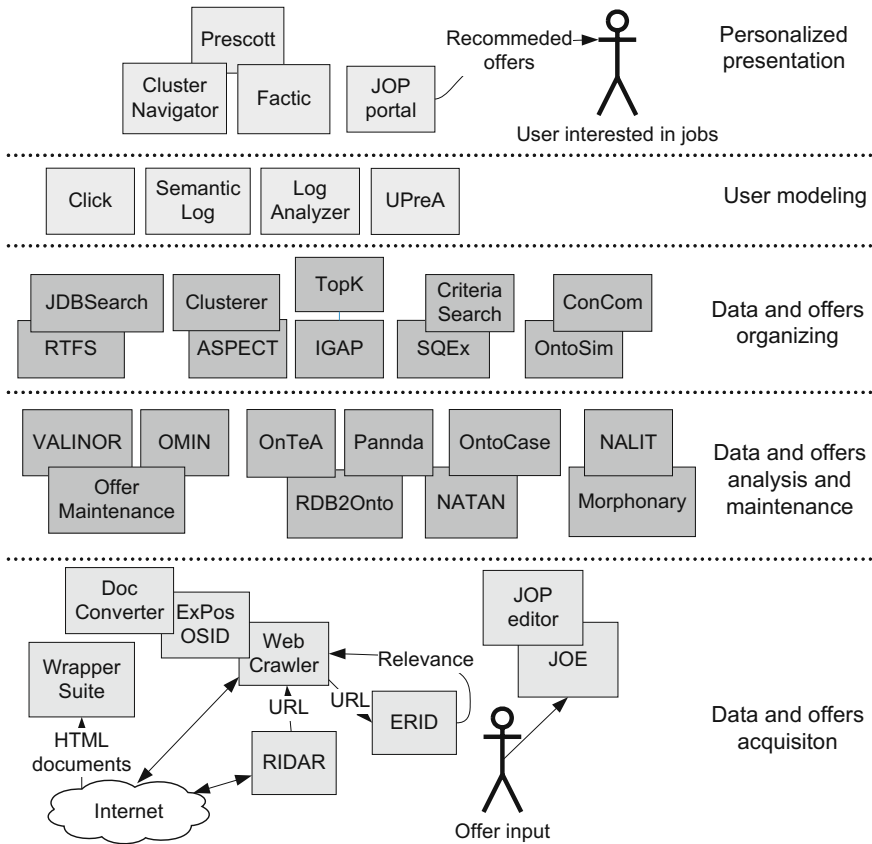


Fig. 3. Tools for acquisition, analysis, organization, and presentation of data and offers

Automatic acquisition of documents by browsing the Web is based on the concept of focused crawling. In the first step, pages that definitely do not contain offers are filtered out. Consequently the offer is extracted from the web page and it is saved into the corporate memory of the information system. We proposed a method of downloading offers realized by chain of several tools: WebCrawler, OSID – Offer Separation for Internet Documents, RIDAR – Estimate Relevance for Internet Documents, ExPoS – Job Offer Extraction from the Web Page [10].

Automatic acquisition of documents by a simple downloading from the known sites is suitable when we know where the information of given type is present (in our case job offers concentrated in several known web portals). This approach is based on constructing wrappers – software tools that extract the required content knowing the structure of the web pages. Important issue in this approach is design of methods for effective creation of wrappers. This includes methods of learning from positive and negative examples so that the creation of a wrapper (which to some extent still requires manual intervention) was effective and so it was able to react to the structure changes of the pages [8].

4.2 Analysis and Organization of Data and Information

In the contemporary Web we find data that provide information and knowledge to humans, however for automatic processing we need to enrich data with its semantics and also with other information needed for effective processing (e.g., indexes important for browsing). There exist several ways how to analyze and organize data so that we get meaningful information. We focus on some aspects that we consider from a certain point of view as representative for this area. They are methods which aim at:

- *annotating and reasoning*: methods serve to enrich acquired data with additional information and meaning [14]; results are used in search when significant information in acquired documents is discovered (e.g., name of the company), or during the presentation by supplementing additional information (e.g., about geographical position of the place) or supplementing relationships between offers important for effective navigation;
- *fulltext indexing*: methods serve to support fulltext browsing in information space (indexed job offers) according to assigned criteria;
- *clustering*: methods for grouping data based on selected criteria; identification of similarity in information space is important for categorization and recommendation, for example in case that the user shows interest in some information (presented offer) we recommend him also similar offers [2];
- *searching*: methods serve to search in information space; besides standard keyword based methods we consider such methods that make use of intelligent query expansion based on an estimation of the user interest,
- *categorization*: methods serve to arrange data according to various criteria; e.g. methods for creating ordered lists and generating rules for classification. As an example we mention a method for induction of regulations for monotonous classification of job offers through which we obtain top-k items concerning assigned preferences of the user [13];
- *text processing*: methods serve for the above mentioned tasks where text analysis is often requested, for example for comparison of offers or their annotation.²

4.3 Information Presentation

When designing methods for presentation we focused on enriching the information space with adaptation to the user and his context. We especially concentrated on adaptive presentation of the content and adaptive navigation in hyperspace [2]. Our goal was to present information to the user in a personalized fashion, i.e. information, which is relevant for him and in addition in such fashion which best suits his needs [9]. For this we proposed the method

² Note that in the field of natural language processing there is a major disproportion between the achieved progress for processing English and other languages (including Slovak). It is given by the languages alone (analysis of a flexive language is more difficult than of English), but also by the volume of applications and effort including means spend on text processing in respective language.

of user behavior analysis, which comes out from defined heuristics with regard to "clicks" of the user (e.g., the meaning of the first activities of the user while browsing information space is provably higher for stating interests of the user than the other ones) [3].

For presentation itself it is possible to use several approaches. It should be noted that we deal with problem of ontology visualization. We proposed two views – one is based on facets that serve for navigation by constraining the information space and the second on visual navigation in clusters [21].

5 Conclusion

The Web contains information about a large number of questions, which can be of interest for us already today – and its content grows day by day. It is becoming one of the most important sources of information, it is just necessary to know how to obtain them from it. Concerning the scale and other properties of the Web, this is not at all a simple task. Without suitable tools the absolute majority of information would stay hidden for the user.

A software tool is an outcome of a development that must be preceded by research of the Web itself, by researching for new methods of data acquisition, organization, and presentation. This research, as this paper tried to outline, has already brought results, but it is clear the research and development must go on. We need to get to know the Web better. It is obvious that it is developing and thus changing constantly. This by no means makes investigating it easier. A change that could make acquiring information easier is to enrich what is written on the Web with at least some indication of its meaning (semantics). This opens room for research of methods that could be more fundamentally different from what we know today.

Acknowledgement. This work was partially supported by the State programme of research and development, SPVV1025/04, by the Slovak Scientific Grant Agency, VG1/0508/09) and by the Slovak Research and Development Agency, APVV-0391-06.

References

1. Ahmad, M.N., Colomb, R.M.: Managing ontologies: a comparative study of ontology servers. In: Bailey, J., Fekete, A. (eds.) Proc. of the Conf. on Australasian Database, pp. 13–22. ACM Press, New York (2007)
2. Andrejko, A., Bielíková, M.: Comparing instances of ontological concepts for personalized recommendation in large information spaces. Computing and Informatics (to appear, 2009)
3. Barla, M., Tvarožek, M., Bielíková, M.: Rule-based user characteristics acquisition from logs with semantics for personalized web-based systems: Updates of logic programs. Computing and Informatics (to appear, 2009)
4. Bartalos, P., et al.: Building an Ontological Base for Experimental Evaluation of Semantic Web Applications. In: van Leeuwen, J., Italiano, G.F., van der Hoek, W., Meinel, C., Sack, H., Plášil, F. (eds.) SOFSEM 2007. LNCS, vol. 4362, pp. 682–692. Springer, Heidelberg (2007)

5. Brusilovsky, P., Kobsa, A., Nejd, W. (eds.): *Adaptive Web 2007*. LNCS, vol. 4321. Springer, Heidelberg (2007)
6. Brusa, G., Caliusco, M.L., Chiotti, O.: A process for building a domain ontology: an experience in developing a government budgetary ontology. In: Orgun, M.A., et al. (eds.) *Proc. of Workshop on Advances in Ontologies*, pp. 7–15. ACM Press, New York (2006)
7. Ding, L., Kolari, P., Ding, Z., Avancha, S.: Using Ontologies in the Semantic Web: A Survey. In: Sharman, R., et al. (eds.) *Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems*, pp. 79–113. Springer, Heidelberg (2007)
8. Frivolt, G., Kisac, I.: Interactive Wrapper Learning for Automatic Data Gathering. In: *Proc. of Tools for Acquisition, Organisation and Presenting of Information and Knowledge (2)*, Research Project Workshop, pp. 63–67 (2007)
9. Gurský, P., Horváth, T., Jirašek, J., Krajčí, S., Novotný, R., Pribolová, J., Vaneková, V., Vojtáš, P.: User preference web search experiments with a system connecting web and user. *Computing and Informatics* (to appear, 2009)
10. Gatial, E., Balogh, Z., Laclavík, M., Ciglan, M., Hluchý, L.: Focused web crawling mechanism based on page relevance. In: Vojtáš, P. (ed.) *Proc. of ITAT, Workshop on Theory and Practice of IT*, pp. 41–46 (2005)
11. Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., von Wilamowitz-Moellendor, M.: Gumo – the general user model ontology. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) *UM 2005*. LNCS (LNAI), vol. 3538, pp. 428–432. Springer, Heidelberg (2005)
12. Hoang, H.H., et al.: Towards a New Approach for Information Retrieval in the SemanticLIFE Digital Memory Framework. In: *IEEE/WIC/ACM Int. Conf. on Web intelligence*, pp. 485–488. IEEE CS Press, Los Alamitos (2006)
13. Horváth, T., Vojtáš, P.: Ordinal Classification with Monotonicity Constraints. In: Perner, P. (ed.) *ICDM 2006*. LNCS (LNAI), vol. 4065, pp. 217–225. Springer, Heidelberg (2006)
14. Laclavík, M., Šeleng, M., Gatial, E., Hluchý, L.: Ontea: Platform for Pattern based Automated Semantic Annotation. *Computing and informatics* (to appear, 2009)
15. Machová, K., Bednár, P., Mach, M.: Various Approaches to Web Information Processing. *Computing and Informatics* 26(3), 301–327 (2007)
16. Návrat, P., Bieliková, M., Rozinajová, V.: Acquiring, Organising and Presenting Information and Knowledge from the Web. In: Rachev, B., Smrikarov, A. (eds.) *Proc. of CompSysTech 2006*, Bulgaria (2006)
17. Návrat, P., Bartoš, P., Bieliková, M., Hluchý, L., Vojtáš, P.: Tools for Acquisition, Organisation and Presenting of Information and Knowledge. In: *Proceedings of Research Project Workshop (2006-2007)*
18. Nekvasil, M., Svátek, V., Labský, M.: Transforming Existing Knowledge Models to Information Extraction Ontologies. In: *11th Int. Conf (BIS 2008)*, pp. 106–117. Springer, Heidelberg (2008)
19. Oberle, D., et al.: DOLCE ergo SUMO: On Foundational and Domain Models in SWIntO. *J. of Web Semantics* 5, 156–174 (2007)
20. Shadbolt, N., Hall, W., Berners-Lee, T.: The Semantic Web Revisited. *IEEE Intelligent Systems*, 96–101 (May/June 2006)
21. Tvarožek, M., Bieliková, M.: Visualization of Personalized Faceted Browser Interfaces. In: Forbrig, P., et al. (eds.) *IFIP Int. Federation for Information Processing. Human-Computer Interaction Symposium*, vol. 272, pp. 213–218. Springer, Heidelberg (2008)

Modeling Ant Activity by Means of Structured HMMs*

Guenael Cabanes¹, Dominique Fresnau², Ugo Galassi³, and Attilio Giordana³

¹ LIPN-CNRS UMR 7030, Université Paris XIII, Villetaneuse, Paris, France

² LEEC, Université Paris XIII, Villetaneuse, Paris, France

³ Dipartimento di Informatica, Università Amedeo Avogadro
Via Bellini 25G, Alessandria, Italy

Abstract. Modeling societies of individuals is a challenging task increasingly attracting the interest of the machine learning community. Here we present an application of graphical model methods in order to model the behavior of an ant colony. Ants are tagged with RFID so that their paths through the environment can be constantly recorded. A Structured Hidden Markov Model has been used to build the model of single individual activities. Then, the global profile of the colony has been traced during the migration from one nest to another. The method provided significant information concerning the social dynamics of ant colonies.

1 Introduction

This paper addresses the problem of analyzing the behavior of an ant colony, in order to discover social rules and social roles, which are still not yet well identified and understood. In the specific case, an ant colony has been observed after creating an artificial climatic mutation, which caused the migration from one nest to another. Facing the emergency, the different elements of the colony change their current social activity, and assume specific roles in order to accomplish the migration in the new nest. Afterwards, they return to normal activities, not necessarily the same they were accomplishing before the emergency. Individual activities are tracked using a network of RFID detectors, which provides rough information about the position in the environment of every single ant. The fundamental issue investigated here is the construction of the model of the activities an individual can undertake, such as: nursing, transporting, foraging, and so on, according to the emerging needs of the colony. Starting from the activity models, global parameters characterizing the global colony behavior are inferred from RFID logs.

The activity models are based on Structured Hidden Markov Models (S-HMM) [1,2], a variant of classical HMM [3], well suited to combine a priori structural information from the domain experts with statistical information inferred from data. The paper describes both the methodological approach and

* This work was supported in part by the *Sillages* project (N° ANR - 05 - BLAN - 017701) financed by the ANR (Agence Nationale de la Recherche).

the in field experimentation. Even if the work presented here is only a preliminary phase of a larger project, the obtained results already provided information considered interesting and novel from domain experts.

2 Experimental Setting

Subject of the experiment is a colony of 57 *Pachycondyla tarsata* workers, tagged with RFID sensors. Each worker had a tag glued to its thorax, consisting of a chip attached to an antenna weighting under 40 mg (i.e., 25% of an ant weight). The picture of a tagged ant and a snapshot of the colony is reported in Figure 1. All individuals have the same physical attitudes and, in principle, can assume any role the social environment requires. In nature, ants live in nests composed by several rooms, interconnected by tunnels, with a single entrance from the external environment. The eggs and the cocoons are kept in the most protected room, i.e. the most distant from the entrance. The nests are usually in the dark (underground) and the humidity level is high. The apparatus, is a complex of two (artificial) nests interconnected through a wide area (foraging area) simulating the external environment where ants can search for food. Every nest is a chain of three rooms connected by tunnels. On each tunnel two RFID detectors are located. We assume that an ant moved in another room when both detectors detect the corresponding RFID in the correct order. Unfortunately, owing to the small dimensions of the RFIDs, the detectors exhibit a remarkable missing rate, which increases the difficulty of the data analysis. The structure of the apparatus is described in Figure 2. The migration from one nest to the other has been induced according the the following procedure. After installing the colony

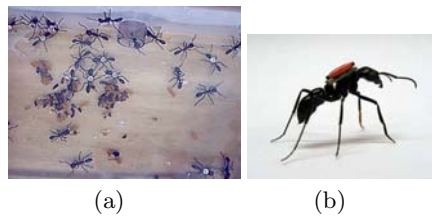


Fig. 1. Ants with RFID tags. (a) Snapshot of a room of the nest. (b) A tagged ant; the global length of an individual is close to one inch.

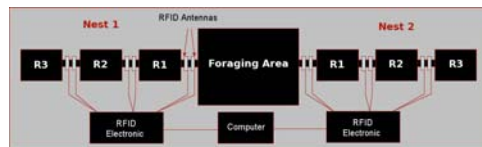


Fig. 2. The experimental apparatus topology. It shows the two nest and the RFID sensorial system.

in one of the nests, we waited until the social life assumed a stable pattern. In this, phase ants could not access the rooms of the other nest closed by a door. Both nests are initially in the dark.

At $time = 0$ a strong neon light (strongly repellent for ants) is switched on over the first nest while the entrance of the second nest, is opened. Then, the colony actions until the entire brood is moved into the second nest (~ 4 hours), are recorded.

3 Modeling Social Activities

The RFID apparatus only provides a partial observation of the individuals. No information is provided concerning what happens inside a room, but only the duration of the permanence of an ant inside it can be known. Moreover, sensors are not reliable. A missing detection rate ranging from 5 to 15 % has been observed.

The goal is to reconstruct the evolution of the activity of the single individuals in the context of the social environment, under the pressure of the simulated ecological mutation. More specifically, we want to discover which kinds of activity are undertaken during the migration phase, how many individuals are dedicated to each activity, and when an individual may change activity.

Achieving this goal requires to solve the following problems: (i) to reconstruct the most likely paths made by ants considering that many transits in the tunnels are unobserved due to missing detections; (ii) to characterize the different activity patterns emerging from the data; (iii) to infer the activity models; (iv) to segment and label the paths according to the activity that most likely produced the observed action sequence.

The major requirements a modeling tool must satisfy are the capacity of handling partially observable status, and modeling the duration of the permanence in the different areas of the environment. To this purpose, the graphical model approach [4] is the most promising one. In particular, two tools emerge as candidate for the specific task of segmenting and labeling sequences in presence of hidden states: HMM [3] and CRFs [5]. Recent findings [4] are in favor of CRF, which in many cases outperformed HMM. Nevertheless, the requirement of modeling duration suggested to us to try first the HMM. In fact, well assessed methods for extending HMMs in order to cope with durations are available, while CRFs have been little investigated in this sense [6]. The adopted tool is then a HMM variant called S-HMM [2], which offers specific features for modeling permanence inside rooms.

S-HMM are block structured according to the paradigm used in Object Oriented Programming. A block consists of a set of states, only two of them (the *initial* state I and the *end* state E) are allowed to be connected to other blocks. Blocks can be nested inside each other.

Two kinds of blocks are possible: *basic blocks* and *composite blocks*. The states of basic blocks produce observable emissions according to the classical HMM paradigm. The states of composite blocks correspond to basic or composite block

defined at a lower abstraction level. When a transition into a state q occurs in a composite block, a call is made to the lower level block associated to q and the activity is suspended until the call returns.

All basic algorithms for computing probability distributions and estimating model parameters from sequences, such as *forward-backward* and *Viterbi* algorithm [3], immediately extend to S-HMM. A detailed description of S-HMM is provided in [21].

3.1 Modeling Duration

The problem of modeling durations in the HMM framework has been principally faced in Signal processing. Two approaches to the problem emerge from the literature. The first one produced an extend modeling tool called Hidden Semi-Markov Model (HSMM), which corresponds to HMM augmented with probability distributions over the state permanence [7,8,9]. The alternative approach is the so called *Expanded HMM* [10]. Every state, where it is required to model duration, is expanded into a network of states, properly interconnected. In this way, the duration of the permanence in the original state is modeled by a sequence of transitions through the new state network where the observation remains constant. The advantage of this method is that the markovian nature of the HMM is preserved. Nevertheless, the complexity increases according to the number of new states generated by expansion.

In the framework of S-HMM, the approach of Expanded HMM is used and specific basic blocks are provided to model the probability distribution of the permanence inside a *macro state*. The specific HMM topology we adopted for the present application is reported in Figure 3 (c). Basically, this model exhibits an Erlang's distribution (Figure 3 (d)), when the Forward-Backward algorithm is used to compute the probability distribution. Basically, an ant activity model assigns a probability distribution over the set of all possible paths an ant accomplishing a specific activity can go through the artificial environment. Let s be a sequence of observations. Comparing the different probability assigned to s by a set of different activity models, it is possible to infer the activity that most likely generated s .

3.2 Ant Activity Model Architecture

The observation of a path is a sequence of pairs $\langle t_i, s_i \rangle$ collected from the RFID sensors, being t_i the time of the detection in msecs, and s_i the id of the sensor. In the S-HMM framework a discrete time is assumed. Then, a transformation from the numeric representation from the sensors to a discrete (symbolic) representation has been defined, which preserves the accuracy implicit in the original coding. The symbolic sequences are encoded using an alphabet $\mathbf{A} = \{A, B, C, D, E, F, G, H, I, J, K, L, .\}$, where letters from A to L correspond to the RFID detectors from 1 to 12, respectively, and “.” denotes a time interval, in which no observation from the sensorial apparatus is received. The transformation, from

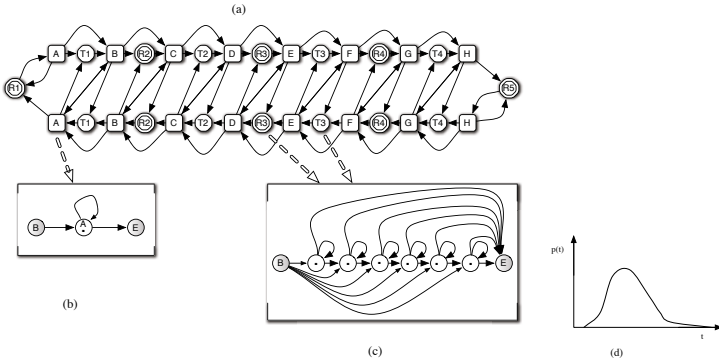


Fig. 3. Structured HMM used for modeling ant behavior. (a) High level model; (b) basic block encoding the sensor behavior; (c) basic block modeling the duration of the permanence in a room; (d) shape of the probability distribution encoded by the duration model.

a numeric to a symbolic sequence is obtained by subdividing the time into discrete intervals of one *second*. A numerical sequence is scanned from the beginning to the end moving ahead of one interval at time. If a signal from a detector is found, the corresponding symbol in the symbolic sequence is appended; otherwise a “.” is inserted. After the translation, the permanence in a room, is be represented as a string of “.”. Moreover, undetected transit in a tunnel will be report as a “.”, as well.

After experimenting with different model architectures, the one reported in Figure 3 has been chosen. It is a two level S-HMM, where the upper level models the long range path through the environment, while the lower level models the observations detected by the RFID sensors, and the duration of the permanence inside rooms and tunnels. States in the upper level defines a double chain sharing the ends. The states denoted with single circles represent the permanence in a tunnel, while the ones denoted with double circle represent the permanence in the rooms of the nests or in the foraging area. States denoted with squares represent sensors. Each state is associated to a block at lower level, which models the probability distribution for the permanence in the associated location, or the process of generating the observable emissions of the sensors. Referring to Figure 3(a), the upper chain models the action of going from the old nest to the new nest, while the other models the action of going from the new nest to the old nest. Changes of directions cause switching from one chain to the other.

4 Learning from the Activity Traces

From the model architecture described in Figure 3, the activity models have been estimated in order to construct an *activity tagger*. This program is used to infer the most likely path of an ant, and to segment and label it according to the activity that most likely generated the signals reported by the RFID detectors.

From the labeled paths the global behavior of the colony along the migration phase has been reconstructed.

4.1 Activity Tagger Architecture

An activity tagger is a three-level S-HMM, obtained by layering a new block on top of the activity models. As shown in Figure 4, this new layer defines a fully connected graph among the blocks modeling the different activities. Then, the activity tagger interprets ant traces as a sequence of segments each one corresponding to a different activity phase. By exploiting the S-HMM compositional properties [11] the activity tagger can be refined by training single blocks independently as well as the entire structure using the classical Baum-Welch algorithm.

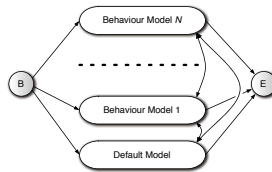


Fig. 4. An Activity tagger is a S-HMM used for modeling colony behavior

The standard method for interpreting a sequence using an HMM (S-HMM) makes use Viterbi algorithm [12,3] in order to find the maximum likelihood path in the model state space, which corresponds to the observed events. In our case this method does not work because the HMM modeling the duration of the permanence in the rooms and in the tunnels, requires forward-backward algorithm.

Then we adopted an alternative method, also described in [3], which consists in finding, at each time t , the maximum likelihood state q_t of the model, as defined by the following equation:

$$q_t = \operatorname{argmax}_i (\alpha_t(i)\beta_t(i)) \quad 1 \leq i \leq N \tag{1}$$

In expression (1) $\alpha_t(i)$ is the classical function that estimates the probability for model λ of being in state q_i after generating (in all possible ways) the sequence of observations from t_0 to time t . Symmetrically, $\beta_t(i)$ is the probability for λ of generating the remaining part of the sequence from t to T starting from status q_i .

4.2 Learning Procedure

The complete learning procedure for learning the activity tagger integrates data-mining algorithms with the manual action of an expert of the domain. The domain expert is very good in detecting where the activity pattern changes, and in providing an episodic interpretation of fragments of the paths, but it is

poorly performing in tasks requiring the systematic analysis on large amount of data. On the other hand, the learning algorithms is very good in discovering regularities and similarities among different episodes discovered by the expert. From this cooperation, the groups of characteristic activities are progressively individuated and modeled. The procedure consists of the following steps, which are repeated until the convergence to stable models is achieved:

Let \mathcal{L} be the set of sequences to be labeled. Let, moreover L_i a subset of \mathcal{L} , used for iteration i .

1. label sequences using the current tagger version;
2. refine the assigned labels with the help of an expert of the domain;
3. segment every sequence according to the assigned label;
4. cluster segments according to the assigned label;
5. from every cluster C_k estimate a model λ_k ;
6. construct a new tagger using the models learned in the previous step, and optionally train it using the Baum-Welch algorithm;
7. add new sequences extracted from \mathcal{L} to L_i obtaining a new learning set L_{i+1} .

Finally, after the procedure iteration stops, all sequences in \mathcal{L} are labeled using the tagger constructed at the last step.

Notice that, the first time the procedure is executed, no tagger exists. In this case, the first step has been carried using an algorithm based on Kohonen maps [13,14], which was able at providing a rough segmentation. Then the domain expert manually corrected the output of the program. As this task is time consuming, we started with a small learning set extracted from the a set \mathcal{L} containing 57 sequences (one for each individual of the colony). The procedure has been iterated three times incrementing the learning set up to 40 sequences. Afterwards, all 57 sequences have been labeled using the final tagger. An example of labeled sequence is reported in Figure 5.

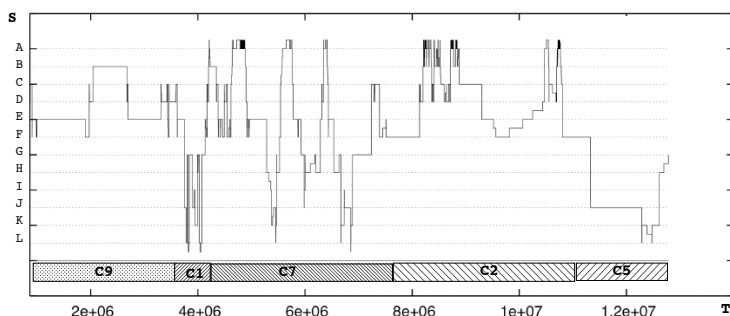


Fig. 5. Example of labeled sequence

4.3 Tracing the Colony Profile

From the labeled sequences three groups of global parameters have been extracted tracing the profile of the ant colony during the migration phase:

1. The temporal evolution of the number \mathcal{N}_j of individuals present in every room R_j of the old and of the new nest.
2. The temporal evolution of the number n_{ij} of individuals involved in the activity characterized by model λ_i , in every room R_j .
3. The global number $N_i = \sum_{j=1}^7 n_{ij}$ of the individual involved in each activity λ_i .

5 Extracted Knowledge Analysis

Nine groups of activities (A1, ... A9) emerged as easy to characterize by means of a S-HMM. A last group (A0) has been defined, which corresponds to activity segments where a clear pattern is not detectable.

The first information extracted from the sensor logs, using the activity tagger, is the path made by every ant during the migration phase. From it, the global parameters \mathcal{N}_j ($1 \leq j \leq 7$), reporting the temporal evolution of the number of individuals in the different rooms, have been computed. The results are described by the diagram of Figure 6. Three facts are emerging:

- (i) Since the beginning a small number of ants is present also in rooms R'_1 , R'_2 and R'_3 of the new nest. This means that the colony started to explore the new nest as soon as the door has been opened. Anyhow, this does not produce remarkably effects until the neon light begun to trouble the colony.
- (ii) After the migration phase was concluded, no more ants were present in the internal rooms R_2 and R_3 of the old nest, but the presence of ants in room R_1 was remarkably higher than in the corresponding room R'_1 of the new nest before the migration. This can be explained considering the combined effect of the of the residual pheromone, which acts as an attractor for the ants in the old nest, and the neon light, which acts as a repellent.
- (iii) The final repartition of individuals on rooms R'_2 and R'_1 is quite different than the one in the corresponding rooms of the old nest before the migration phase. More specifically, the percentage of individuals in room R'_2 is much higher. The explanation is that ants were busy with cocoons located in room R'_2 . The distribution shows tendency to returning to the normal values at the end of the period of observation. Figure 7 shows the evolution of parameters N_i corresponding to the number of individuals involved in activity $A0, \dots, A9$. From the analysis of the distribution of the activities on the rooms (n_{ij}), it appears that $A2$ and $A9$ are just variants of a same type of activity. This is also confirmed by the distance from the two models computed according to the measure described in 3. Moreover, $A8$ exhibits the same pattern as $A2$ and $A9$, with the difference that it occurs in the rooms of the new nest while the others occur in the old nest. Also in this case the distance among the models supports this hypothesis.

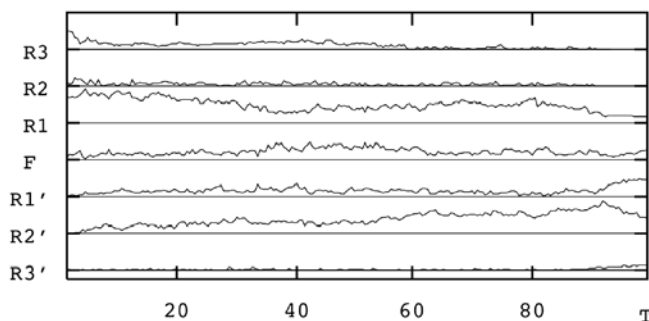


Fig. 6. Distribution of the individuals on the different rooms of the environment during the migration phase

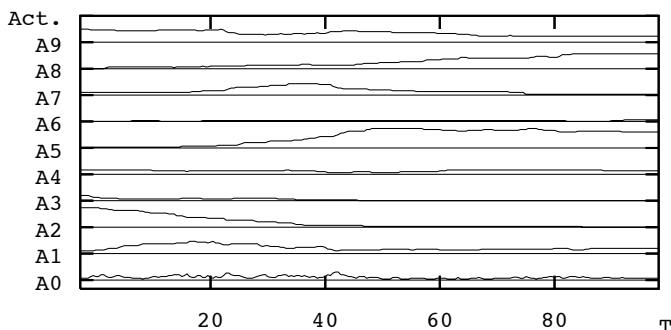


Fig. 7. Evolution of the number of individual involved in the different activities

Finally, a regular pattern, where activities ($A2|A9$), $A1$, $A7$, and $A8$ regularly occur in sequence has been found in about the 60% of the individuals. This sequence find a plausible explanation. $A2$ and $A9$ correspond to a routine phase of exploration of the old nest, $A1$ is a short phase where the new nest is actively explored, $A7$ is the phase in which eggs, cocoons and food are moved in the new nest, and $A8$ is the return to the normal activity of routinary exploration of the new nest.

6 Final Remarks

We have presented a novel application of graphical model methods to the interpretation of data collected from an ant colony. Very encouraging results have been obtained using S-HMM, a variant of HMM, well suited for modeling duration. On the basis of these finding a new research phase will start where the experiment will be repeated with a larger colony (120 individuals) in a more complex environment. Further experimentation with other kinds of graphical

models, such as CRF will also be investigated. It is worth noting that the presented application is very innovative with respect to the current state of the art. In fact, even if several authors addressed the problem of modeling insect colonies, this has been made with different goals. In general the interest has been to study the emerging behavior of the colony, seen as a complex system, starting from a simple model of the individuals. In our case, we started from the opposite point of view proposing a method for observing and modeling the medium/long term behavior of real individuals, as it has been induced by the conditioning of the social environment, in an emergency condition. Finally, RFID methods for tracking the positions of people or animals begin to be quite a diffused practice, which is attracting the interest of data mining community. Nevertheless, no methods for constructing a complex model of the behavior of an individual traced by an RFID had been proposed until now.

References

1. Galassi, U.: Structured Hidden Markov Models: A General Tool for Modeling Process Behavior. PhD thesis, Università degli Studi di Torino, Dottorato di ricerca in Informatica (April 2008)
2. Galassi, U., Giordana, A., Saitta, L.: Incremental construction of structured hidden markov models. In: Veloso, M.M. (ed.) IJCAI, pp. 798–803 (2007)
3. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of IEEE* 77(2), 257–286 (1989)
4. Murphy, K.P.: Dynamic Bayesian Networks: Representation, Inference and Learning. Ph.D thesis, Dpt. of Computer Science, UC, Berkeley (2002)
5. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. 18th International Conf. on Machine Learning, pp. 282–289. Morgan Kaufmann, San Francisco (2001)
6. Natarajan, P., Nevatia, R.: View and scale invariant action recognition using multiview shape-flow models. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8 (2008)
7. Levinson, S.E.: Continuous variable duration hidden markov models for automatic speech recognition. *Computer Speech and Language* 1, 29–45 (1986)
8. Pytkkonen, J., Kurimo, M.: Using phone durations in finnish large vocabulary continuous speech recognition (2004)
9. Tweed, D., Fisher, R., Bins, J., List, T.: Efficient hidden semi-markov model inference for structured video sequences. In: Proc. 2nd Joint IEEE Int. Workshop on VSPETS, Beijing, China, pp. 247–254 (2005)
10. Josep, A.B.: Duration modeling with expanded hmm applied to speech recognition
11. Galassi, U., Giordana, A., Saitta, L.: Structured hidden markov models: A general tool for modeling agent behaviors. In: *Soft Computing Applications in Business. Studies in Fuzziness and Soft Computing*, vol. 230, pp. 273–292. Springer, Heidelberg (2008)
12. Forney, G.D.: The viterbi algorithm. *Proceedings of IEEE* 61, 268–278 (1973)
13. Cabanes, G., Bennani, Y., Chartagnat, C., Fresneau, D.: Topographic connectionist unsupervised learning for RFID behavior data mining. In: IWRT, pp. 63–72 (2008)
14. Cabanes, G., Bennani, Y.: A local density-based simultaneous two-level algorithm for topographic clustering. In: IJCNN, pp. 1176–1182. IEEE, Los Alamitos (2008)

Modern Approach for Building of Multi-Agent Systems

Lukasz Chomatek and Aneta Poniszewska-Marańda

Institute of Computer Science, Technical University of Lodz, Poland
lukaszch@ics.p.lodz.pl, anetap@ics.p.lodz.pl

Abstract. Different approaches for distributed programming in modern hardware architectures allows the developers to build the efficient solutions of complicated technical and information problems. The technologies such as Web Services allow the applications to create a cross-platform for data exchange. The multi-agent systems, where a communication between the agents is essential for proper work of such applications can be developed using the technology of Service Oriented Architecture (SOA). The presented article presents how to apply the modern programming technologies, design patterns and software architectures to building standards of multi-agent systems.

1 Introduction

The most common architecture for the multi-agent systems was developed about six years ago and published as a standard by FIPA organization. Since then many new technologies appeared in the computer science, so we decided to check whether they are applicable to these standards.

Different approaches for distributed programming in modern hardware architectures allows the developers to build the efficient solutions of complicated technical and information problems. The technologies such as Web Services allow the applications to create a cross-platform for data exchange. The multi-agent systems, where a communication between the agents is essential for the proper work of such applications can be developed using the technology of Service Oriented Architecture (SOA). The presented article presents how to apply the modern programming technologies, design patterns and software architectures to building standards of multi-agent systems.

The first part of this paper briefly describes a standard architecture for the multi-agent systems, the second part deals with some technologies and design patterns which can be applied to build a scalable and functional multi-agent system frameworks. The third part shows the example of implementation of these techniques - contains the simplification of FIPA architecture for the multi-agent systems.

2 Overview of FIPA Architecture

In 2002 the Foundation on Intelligent Physical Agents (FIPA) published the standards describing the multi-agent systems [4,5]. According them the agents

are the programs that expose and consume some services. The main parts of the multi-agent system according FIPA standards are shown on Fig. 1. The information about the names of agents and their addresses are stored in *Agent Management System* (AMS) [4,5]. The *Directory Facilitator* (DF) is an optional component in FIPA architecture [4]. This module manages the information about the services exposed by an agents in the system, like services' names, parameters and type of returned values. The communication between agents is possible using the *Message Transport System* (MTS) [4].

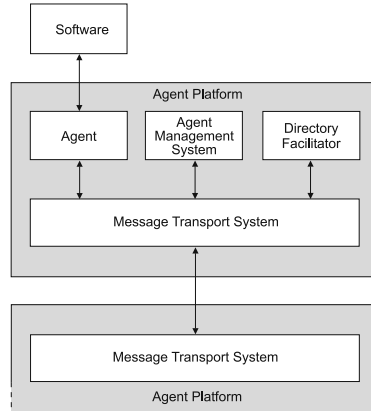


Fig. 1. The parts of FIPA multi-agent systems [4]

The authors of FIPA specifications considered to use different languages for the agents on different platforms. The task of Message Transport System is translated to the messages in the language suitable for the agents residing on a platform. The main languages described in FIPA standards are FIPA-SL and FIPA-ACL (*Agent Communication Language*) [6]. The first of mentioned languages is used to define the content of a message, while the second one describes its envelope. The example of a message written in those languages is as follow:

1. (query-if
2. :sender (agent-identifier name: StockClient)
3. :receiver (set (agent-identifier name:StockServer))
4. :content
5. "((exist (company XYZ) (stock GPW)))"
6. ...)

The first line describes an action that the sender agent uses. The fifth line is a content line - the agent *StockClient* asks the agent *StockServer* if a company XYZ exists on the stock exchange named GPW. If the company does not exist, the agent *StockServer* sends a following reply:

1. (inform
2. :sender (agent-identifier name: StockServer)
3. :receiver (set (agent-identifier name:StockClient))
4. :content
5. "((not (exist (company XYZ) (stock GPW))))")
6. ...)

3 Building of Multi-Agent Systems

The specificity of operations of intelligent agents enforces for taking into consideration their specific aspects [9][10]:

- Cooperation with existing objects - an agent has to communicate with the external objects that most often have different interfaces than he himself. If an agent can take the information from external world, he should change it on the information comprehensible for other agents.
- Cooperation with other agents - it should proceed used certain protocols. The communication between two agents should be possible and the sending messages should be comprehensible for these agents.
- Synchronization - an agent most often realizes his actions using some threads.
- Migration - an agent has to have an access to the resources independently on a place where he is.
- Automatic concluding - an agent chooses a service or a plan of operation based on his knowledge.
- Autonomous behavior - an agent has to be able to realize his task.

The problem of agent synchronization can be solved using the solutions given on the new programming platforms. The problems with cooperation of agents with other agents or external applications and the migration problem can be solved due to employment of the service oriented architecture [23].

The analysis of modern programming techniques shows that some practices can be applied in newly designed multi-agents systems:

- use Service Oriented Architecture (SOA) to simplify and improve the possibilities of agent communication,
- make Directory Facilitator the mandatory part of multi-agent system,
- try to apply the enterprise design patterns such as dependency injection to coordinate the communication of agents on a single machine,
- simplify the architecture using the Windows Communication Foundation (WCF) [11][12].

3.1 Multi-Agent Systems Based on SOA

The introduction of web services allowed the developers to connect the applications based on different software and hardware platforms, for example Java and .NET Framework. The Web Services use a specific protocol to expose a schema of transferred data and allow the clients to make the synchronous calls of exposed methods.

The Windows Communication Foundation (WCF), that is a part of .NET Framework 3.5 introduces an extension of web service technology [11,12]. The main advantages of this platform are the possibility of asynchronous service calls, and the communication using many protocols such as TCP or named pipes.

The newest version of .NET technology offers a great support for applications built in Service Oriented Architecture [2,3]. The schema of Service Oriented Architecture is presented on Fig. 2. In the SOA a system contains the applications that can be divided into three groups:

- *Service Provider* is an application that expose a service that other application can connect to; the services are often used for complicated remote calculation tasks,
- *Service Consumer* is an application that needs some data from other application,
- *Service Registry* keeps the information such as service addresses, names, parameters and returned values of the exposed methods.

Comparing this schema with the multi-agent system architecture proposed by FIPA, we can find that *Service Registry* has the same task as *Directory Facilitator* and *Service Provider* and *Service Consumer* are both the Agents. The role of *Message Transport System* is assigned to the web services or its extensions.

The use of web services determines each agent to expose some services, if it wants other agents to be able to communicate with it. If any agent does not expose a web service but it wants to expose some data, any other protocol can be used by other agents to communicate with it.

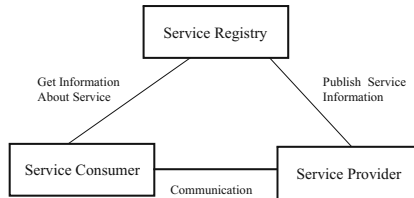


Fig. 2. Service Oriented Architecture Schema

The service oriented architecture is used in the process of creation of the multi-agent systems. It gives the system more elastic and moreover the interaction between the agents is simpler and its safety is on the upper level [2,3].

Figure 3 presents the skeleton of proposed architecture of the multi-agent system based on the service oriented architecture. An agent uses the service directories: *Directory Facilitator* and *Agent Management System*, he makes accessible certain services and he profits by the services offered by other agents.

The proposed skeleton makes possible to create the elastic multi-agent system. This system contains the agents that communicate even if they do not know the addresses of other agents and it contains the base of services that is used by these

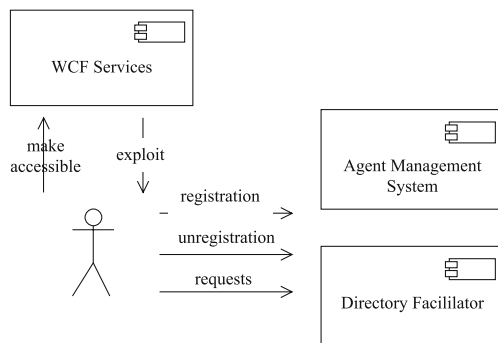


Fig. 3. Architecture skeleton of multi-agent system using SOA

agents. The access to the set of services, being under control of the Directory Facilitator, is possible also for a user who can get to know how the services act in the system. The agents taking a role of clients can establish the connections with the services offered by other agents in the dynamic manner.

3.2 Implementation of Directory Facilitator

However, in the FIPA specification the *Directory Facilitator* (DF) component is considered as an optional part of the multi-agent system. If such a system is being implemented respecting the SOA principles, this component should be prepared. For the proper identification of exposed service only its address is needed, so it is a mandatory information to be kept in the DF database. The Web Service Definition Language (WSDL) is used to prepare a proxy class for the client application, to generate a special proxy, that makes the communication act transparent for the programmers.

However, the service proxy classes can be generated only with the WSDL documents, efficient Directory Facilitator should store other information:

- remote service interface name,
- names of methods in the interface,
- parameters of these methods,
- returned value type.

The complete information about a service allows the potential clients to choose a proper service to communicate with, instead of querying all services about their interfaces. The methods of Directory Facilitator, that allow to fulfill its task are:

- *Register* - stores a new service in a database,
- *Deregister* - removes the information about a service from a database,
- *GetServicesByContract* - gets the information about the services implementing the desired interfaces.

The good practice is to implement the DF as a web service, so that all agents can use common way to invoke its methods.

3.3 Use of Dependency Injection

Dependency Injection design pattern allows the users to resolve the references to objects they want to communicate with from a container [11]. First of all, the objects are registered in a container with a unique name. To hide the real classes of the objects, they are stored as the concrete interface implementations. An example of use of this pattern can be as follows (Fig. 4):

Let consider a situation, when an instance of *classA* wants to invoke a *method1* from *InterfaceB*. First of all the instance calls a generic method *getByInterfaceInterfaceB_i()* from the container object. Next, the container looks up for *InterfaceB* instance and passes a reference to a *classA* instance.

The main idea of *Dependency Injection* pattern is the same as Directory Facilitator [11]:

- the client object receives the ability to communicate with the other object dynamically,
- there is a special object that keeps the information about the objects in the system,
- the objects in the system do not have to keep the references to other objects.

On the other hand, the dependency injection containers give the client objects the references to real interface implementations, not to data required to build the proxy object. In the situations when the agents are located on the same machine, the use of dependency injection is an efficient way to provide the communication protocol for the agents.

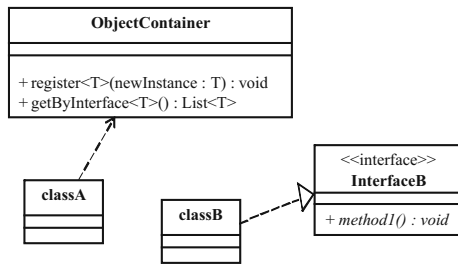


Fig. 4. Exemplary classes for dependency injection

4 Creation of Stock Exchange Agent System

Using the techniques mentioned in the previous section, the Stock Exchange Agent System (SEAS) was developed. The main goal of the system development process was to:

- simplify the architecture of multi-agent system providing the system scalability,
- use the efficient communication patterns for data exchange between the agents,

- provide the transparency of classes for the system developers,
- check the possibilities of agents to predict the shares prices on the stock exchange.

4.1 SEAS Multi-Agent System

The Stock Exchange Agent System, the multi-agent system, is designed to support the investment decisions on the the financial market. The architecture of SEAS is shown on Fig. 5. There are two types of agents in the client side application:

- *client agent* used to communicate with the database and the indicator agents in the range of accessing the data about the prices of shares and the values of indexes,
- *reasoning agent* used to rank the companies and to look-ahead the future share prices basing on the earliest data.

The agents on the sever side are:

- *database agent* that has to supply other agents the information about the prices of shares
- and *indicator agents* realizing the functionality of different technical indexes.

The administration operations, i.e. registration, unregistration and realization of the queries about the agents are possible in the *Directory Facilitator* and in the *Agent Management System* that operate as the system services.

The role of *Service Facilitator* is to keep the information about the services of agents and respond for the agent queries. The previous task of *Agent Management System* was to keep the basic information about the agents, as in FIPA specifications [4,5], however our experience shows that this component is unnecessary in this system.

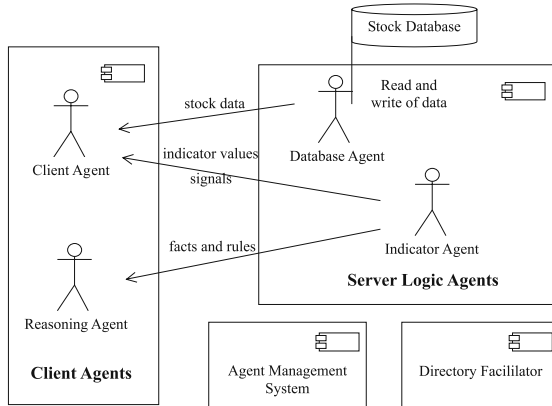


Fig. 5. SEAS architecture

4.2 Simplified Architecture

As SEAS was built in the .NET Framework, the WCF technology is used to provide the communication abilities of agents because of built-in support for the multi-protocol two-way communication. Due to its advantages, the WCF plays the role of *Message Transport System* (Fig. 1). Moreover, the SEAS is designed in Service Oriented Architecture and the use of *Directory Facilitator* is mandatory for the agents in order to accelerate the service discovery process. In FIPA architecture the implementation of this component is optional [4], however in SEAS it is important because it is the central repository of the services exposed in the system.

Furthermore, the *Agent Management System* is mandatory in FIPA architecture [4]. If there is no *Directory Facilitator*, the agents can ask AMS for addresses of other agents and then ask those agents for the exposed services. In SEAS this functionality is omitted, the agents do not play the role of the communication addresses. The AMS is implemented in SEAS, however it is obsolete.

The simplification of the multi-agent system architecture for SOA makes it easy to develop the new agents and extend the administrative services such as Directory Facilitator. The use of Windows Communication Foundation simplified the problem of message exchange between the agents.

4.3 Agents as WCF Service Hosts

The agent classes in the SEAS inherit from *BasicAgent* class. When a *BasicAgent* object is constructed, only one needs to pass a list of services to host as a parameter. Each agent has a collection of the instances of *ServiceHost* class that is used to host the WCF services. Moreover, it is very easy to create dynamically the services.

```

1. foreach (var service in servicesToHost)
2. {
3.     Uri uri = new Uri("net.tcp://localhost/Agents/" +
4.     name + "/" + service.Name);
5.     ServiceHost host = new ServiceHost(service,
6.     new Uri[] { uri });
7.     host.AddServiceEndpoint(service.Namespace
8.     +"."+service.Name, new NetTcpBinding(),
9.     "net.tcp://localhost/Agents/" + name +
10.     "/" + service.Name + "/Endpoint");
11.     hosts.Add(host);
12. }

```

The address of concrete service depends on the agent name and service type. An agent cannot host two services of the same type, but it was found unnecessary during the system development process. After creating its own services, the agent has to register them in Directory Facilitator.

To call a WCF service, the service proxy class should be created and next it allows to invoke the remote methods. This proxy is usually created in a static way by special tools. For example, to generate a proxy class in .NET, the *svcutil* tool should be used. This tool generates the XML schema file with a service description based on the information from the specified service (service meta data) and then generates the C# client class.

Because the agents do not know the agent with which they are going to communicate, the proxy creation should be a dynamic process. To create the proxy class dynamically, the generic class should be built and implements the *ICommunicationObject* interface. One of the core part of this class are *ChannelFactory*, created as follows:

```
_factory = new ChannelFactory<IServiceContract>(binding, remoteAddress);
```

After creating a communication channel factory, a communication channel should be created by invoking:

```
IDesiredService client = _factory.CreateChannel();
```

where *IServiceContract* is a template parameter type for the generic class.

To discover a service, the agent has to ask DF about the description of services of concrete type. Next, it creates a proxy and invokes the remote method.

4.4 Communication Pattern in SEAS

The communication between the agents in the SEAS system is based on the "recommend" [7,8] action from FIPA-ACL [6] specification. At first, an agent B registers the services in Directory Facilitator database. Next, an agent A queries the DF about the specific service. Directory Facilitator gives the agent A an address of agent B. As the agent A knows the address and in general the binding method, it can communicate with this agent. The last part of the communication act is the call of remote method. In SEAS, the remote method call is synchronous, but WCF technology allows the asynchronous method call as well (Fig. 6).

This communication pattern matches the possibilities given by SOA and WCF technology. The use of "broker" or "recruit" [7,8] pattern would be inappropriate in the synchronous communication, as in these patterns the communication act is divided into communication between DF with agent A and DF with agent B. The most significant part of this model is that the agent A knows the address

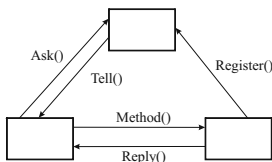


Fig. 6. Communication pattern in SEAS

of agent B and can reuse it. Moreover, three two-way method calls are needed for the first communication, but if agent A wants to reuse the given address, it only has to call one remote method. The brokering and recruiting are a better solution for an asynchronous communication.

5 Conclusions

The development process of Stock Exchange Agent System showed that the techniques mentioned in Section 3 can be successfully applied in the frameworks for multi-agent system. The FIPA architecture with its generality and scalability is a good background for new multi-agent system frameworks but in the modern architectures Directory Facilitator should play bigger role than Agent Management System.

For the network systems the use of Service Oriented Architecture is the better choice rather than the container-managed application with the implementation of dependency injection. It is caused by the fact that in most platforms a self-hosting of the web services can be implemented and there is no need to use an extra layer for the agent-service communication. The second point is that the dynamic service proxies can be also easily created, so there is no need for the agents to keep the static references to other agents, that they want to communicate with. In the single-machine systems a role of Directory Facilitator can be played by the dependency injection design pattern.

References

1. Vasters, C.: Introduction to Building WCF Services, MSDN Library (2005)
2. Krafzig, D., Banke, K., Slama, D.: Enterprise SOA: Service-oriented Architecture Best Practices. Prentice Hall PTR, Englewood Cliffs (2005)
3. Hasan, J., Duran, M.: Expert service-oriented architecture in C# 2005. Apress (2006)
4. FIPA, FIPA Abstract Architecture Specification (2002)
5. FIPA, FIPA Agent Management Specification (2004)
6. FIPA, FIPA ACL Message Structure Specification (2002)
7. Finin, T., Labrou, Y., Mayfield, J.: KQML as an agent communication language. In: Conference on Information and Knowledge Management, USA (1994)
8. Hayden, S.C., Carrick, C., Yang, Q.: Architectural Design Patterns for Multiagent Coordination. In: International Conference on Agent Systems, USA (1999)
9. Herrero, A., Corchado, E., Pellicer, M., Abraham, A.: Hybrid Multi Agent - Neural Network Intrusion Detection with Mobile Visualization. In: 2nd International Workshop on Hybrid Artificial Intelligence Systems (2007)
10. Kendall, E., Malkoun, M., Jiang, C.: Multiagent System Design Based on Object-Oriented Patterns (1997)
11. Newton, K.: The Definitive Guide to the Microsoft Enterprise Library. Apress (2007)
12. Peiris, C., Mulder, D., Cicoria, S., Bahree, A., Pathak, N.: Pro WCF: practical Microsoft SOA implementation. Apress (2007)

Relational Sequence Clustering for Aggregating Similar Agents

Grazia Bombini, Nicola Di Mauro, Stefano Ferilli, and Floriana Esposito

Università degli Studi di Bari, Dipartimento di Informatica, 70125 Bari, Italy
{bombini,ndm,ferilli,esposito}@di.uniba.it

Abstract. Many clustering methods are based on flat descriptions, while data regarding real-world domains include heterogeneous objects related to each other in multiple ways. For instance, in the field of Multi-Agent System, multiple agents interact with the environment and with other agents. In this case, in order to act effectively an agent should be able to recognise the behaviours adopted by other agents. Actions taken by an agent are sequential, and thus its behaviour can be expressed as a sequence of actions. Inferring knowledge about competing and/or companion agents by observing their actions is very beneficial to construct a behavioural model of the agent population. In this paper we propose a clustering method for relational sequences able to aggregate companion agent behaviours. The algorithm has been tested on a real world dataset proving its validity.

Keywords: Sequence Clustering, Relational Sequence Similarity.

1 Introduction

Many clustering methods are based on flat descriptions, in which data points are represented as a fixed-length attribute vector. However, datasets belonging to real-world domains include heterogeneous objects related to each other in multiple ways. For instance in the field of Multi-Agent System (MAS), multiple agents (artificial or human) interact with the environment and with other agents. From a behaviour analysis point of view, MASs are in many aspects similar to human society. Indeed, in order to act effectively an agent should be able to recognise the behaviours adopted by other agents. Actions taken by an agent are sequential, and thus its behaviour can be expressed as a sequence of actions. Each action can be related in different way with respect to other agents and to the environment in which agents interact. Inferring knowledge about competing and/or companion agents by observing their actions is very beneficial to construct a behavioural model of the agent population. In order to reach this goal, it is necessary to define a measure able to assess the similarity between different behaviours described by relational sequences.

In the field of unsupervised data analysis, clustering algorithms provide a useful methods to explore complex data structures. Clustering methods have been exploited in many disciplines, such as data mining, document retrieval,

image segmentation and pattern classification [1,2,3]. A clustering method tries to aggregate data on the basis of a *similarity* (or *dissimilarity*) criteria, where groups (or *cluster*) are defined by a set of similar objects. The two key issues in clustering are the *object representation* and the design of the *similarity measure*. Sequence Clustering concerns grouping a set of sequences into clusters by using a similarity criteria capturing the difference between sequences. When the sequence are expressed in a relational language, we name the task *Relational Sequence Clustering*. Sequence data can be generated from a variety of domains, such as DNA sequencing, speech processing, customer transaction and robot sensor analysis to name a few [4,3].

The solution that we propose is to represent each relational sequence by a set of *features* and then to exploit these features to compute a *similarity value* between sequences. This solution presents two problems: (P1) how to extract the features from relational sequences, and (P2) how to asses a similarity value between two feature-based sequence descriptions. This paper represents, to our knowledge, the first proposal for clustering relational sequences. In particular, we will use a relational learning algorithm to mine meaningful features among the relational sequences and we will use these features to construct a feature vector for each sequence. Then we adapt the Tanimoto measure [5] to compute the similarity between feature vectors, and finally the Partition Around Medoids (PAM) algorithm [6] will be used to aggregate the sequences. The proposed method has been applied to a real world problem and the results prove its validity.

2 Relational Sequences: Representation and Mining

In this section we present a method based on relational pattern mining, to extract meaningful features able to represent relational sequences.

2.1 Sequence Features

Before to design a clustering method, it is necessary to define an appropriate *similarity measure* between sequences. The measure will be applied to data objects represented as a set of features expressing the special properties (features) of a sequence in a specific domain. A way to represent a sequence as a feature vector is to use the patterns occurring in it as true features.

Given an alphabet of symbols \mathcal{A} , and let be $k \geq 1$ a positive integer, then a **k -gram** (k -mers), is a sequence σ of symbols over \mathcal{A} of length k ($\sigma \in \mathcal{A}^k$, $|\sigma| = k$). For a given sequence $\sigma = (s_1 s_2 \dots s_t)$, the k -grams of interest are all subsequences $\sigma' = (s_i s_{i+1} \dots s_{i+k-1})$ of length k occurring in σ .

Given a sequence $\sigma = (s_1 s_2 \dots s_t)$ with $|\sigma| = t$, we define K_σ as the set of all k -grams Ω_k of σ , $1 \leq k \leq t$:

$$K_\sigma = \bigcup_{k=1}^t \Omega_k = \bigcup_{k=1}^t \{\omega_{k1}, \omega_{k2}, \dots, \omega_{kn_k}\} \quad (1)$$

where $\omega_{ki} = (s_i s_{i+1} \dots s_{i+k-1})$, and $n_k = t - k + 1$.

Given a set of sequences $\mathcal{S} = \{\sigma_i\}_{i=1}^n$, \mathcal{K} is the set of all k -grams on all the sequences belonging to \mathcal{S} : $\mathcal{K} = \bigcup_{i=1}^n K_{\sigma_i}$, where \mathcal{K} represents the set of all features over \mathcal{S} .

2.2 Logical Background

A relational sequence is represented by a set of Datalog [7] atoms, based on a first-order *alphabet* consisting of a set of *constants*, a set of *variables*, a set of *function symbols*, and a non-empty set of *predicate symbols*. Each function symbol and each predicate symbol has an *arity*, representing the number of arguments the function/predicate has. Constants may be viewed as function symbols of arity 0. An atom $p(t_1, \dots, t_n)$ (or atomic formula) is a predicate symbol p of arity n applied to n terms t_i (i.e., a constant symbol, a variable symbols, or an n -ary function symbol f applied to n terms t_1, t_2, \dots, t_n). A *ground term* or *atom* is one that not contain any variables. A *clause* is a formula of the form $\forall X_1 \forall X_2 \dots \forall X_n (L_1 \vee L_2 \vee \dots \vee \bar{L}_i \vee \bar{L}_{i+1} \vee \dots \vee \bar{L}_m)$ where each L_i is a literal and X_1, X_2, \dots, X_n are all the variables occurring in $L_1 \vee L_2 \vee \dots \bar{L}_i \vee \dots \bar{L}_m$. Most commonly the same clause is written as an implication $L_1, L_2, \dots, L_{i-1} \leftarrow L_i, L_{i+1}, \dots, L_m$, where L_1, L_2, \dots, L_{i-1} is the *head* of the clause and L_i, L_{i+1}, \dots, L_m is the *body* of the clause. Clauses, literals and terms are said to be *ground* whenever they do not contain variables.

A *substitution* θ is defined as a set of bindings $\{X_1 \leftarrow a_1, \dots, X_n \leftarrow a_n\}$ where $X_i, 1 \leq i \leq n$ is a variable and $a_i, 1 \leq i \leq n$ is a term. A substitution θ is applicable to an expression e , obtaining the expression $e\theta$, by replacing all variables X_i with their corresponding terms a_i . A conjunction A is θ -*subsumed* by a conjunction B , denoted by $A \preceq_{\theta} B$, if there exists a substitution θ such that $A\theta \subseteq B$. A clause c_1 θ -*subsumes* a clause c_2 if and only if there exists a substitution σ such that $c_1\sigma \subseteq c_2$. c_1 is a *generalization* of c_2 (and c_2 a *specialization* of c_1) under θ -subsumption. If c_1 θ -subsumes c_2 then $c_1 \models c_2$.

Definition 1 (Relational (sub)sequence). A relational sequence is an ordered list of atoms. Given a sequence $\sigma = (s_1 s_2 \dots s_m)$, a sequence $\sigma' = (s'_1 s'_2 \dots s'_k)$ is a subsequence (or pattern) of the sequence σ , indicated by $\sigma' \sqsubseteq \sigma$, if

1. $1 \leq k \leq m$;
2. $\exists j, 1 \leq j \leq m - k$ and a substitution θ s.t. $\forall i, 1 \leq i \leq k: s'_i \theta = s_{j+i}$.

A subsequence occur in a sequence if exists at least a mapping from elements of σ' into the element of σ such that the previous condition are hold. In our case, that subsequence is a relational pattern.

The *support* of a sequence σ in a set of sequences \mathcal{S} corresponds to the number of sequences in \mathcal{S} containing the sequence σ : $\text{support}(\sigma) = |\{\sigma' | \sigma' \in \mathcal{S} \wedge \sigma \sqsubseteq \sigma'\}|$.

Now we can translate the concept of k -grams to the relational case.

Definition 2 (Relational k -gram). Given an alphabet of atoms \mathcal{A} , a relational k -gram is a relational sequence σ of length k defined over \mathcal{A} .

Given a set of relational sequences $\mathcal{S} = \{\sigma_i\}_{i=1}^n$, \mathcal{K} is the set of all relational k -grams on all the sequences belonging to \mathcal{S} : $\mathcal{K} = \bigcup_{i=1}^n K_{\sigma_i}$, where K_{σ_i} is the set of all relational k -grams over the sequence σ_i . In particular, \mathcal{K} represents the set of all relational features over \mathcal{S} . We define $\mathcal{K}(\alpha) \subseteq \mathcal{K}$, the set of relational k -grams having a support greater than $\alpha - 1$: $\mathcal{K}(\alpha) = \{\sigma \mid \sigma \in \mathcal{K} \wedge support(\sigma) \geq \alpha\}$.

2.3 Mining Relational Sequential Patterns

In order to select the best set of features, we use an Inductive Logic Programming (ILP) [8] algorithm, based on [9], for discovering relational patterns from sequences. It is based on a level-wise search method, known in data mining from the APRIORI algorithm [10]. It takes into account the sequences, tagged with the belonging class, and the α parameter denoting the minimum support of the patterns. It is essentially composed by two steps, one for generating pattern candidates and the other for evaluating their support. The level-wise algorithm makes a breadth-first search in the lattice of patterns ordered by a specialization relation. Starting from the most general patterns, at each level of the lattice the algorithm generates candidates by using the lattice structure and then evaluates the frequencies of the candidates. Since the monotonicity of pattern frequency (if a pattern is not frequent then none of its specializations is frequent), in this phase some patterns may be discarded.

The generation of the patterns actually present in the sequences of the dataset, is based on a top-down approach. The algorithm starts with the most general patterns. These initial patterns are all of length 1 and are generated by adding an atom to the empty pattern. Then, at each step it tries to specialize all the potential patterns, discarding those that do not occur in any sequence and storing the ones whose length is equal to the user specified input parameter *maxsize*. Furthermore, for each new refined pattern, semantically equivalent patterns are detected, by using the θ -subsumption relation, and discarded. In the specialisation phase, the specialisation operator under θ -subsumption is used. Basically, the operator adds atoms to the pattern. Finally, the algorithm may use a background knowledge \mathcal{B} (a set of Datalog clauses) containing constraints on how to explore the lattice.

3 Clustering Relational Sequences

Now we propose a distance function to measure the dissimilarity between two relational sequences, and the how those distances will be used in the Partition Around Medoids (PAM) algorithm to aggregate similar sequences.

3.1 Distance Function over Relational Sequences

A sequence distance function is a function d that maps a pair of sequences to a non-negative real number to measure the (dis)similarity between two sequences. A sequence distance satisfies the follow properties:

- $d(x, y) > 0$ for sequence x and y such that $x \neq y$;
- $d(x, x) = 0$ for all sequences x ;
- $d(x, y) = d(y, x)$ for all sequences x and y ;
- $d(x, y) \leq d(x, z) + d(z, y)$ for all sequences x, y and z .

Given a set of sequences \mathcal{S} , we apply the algorithm previously described in Section 2.3, and completely reported in [9], to find all the relational k -grams $\mathcal{K}(\alpha)$ over the set \mathcal{S} with a support at least equal to α . $\mathcal{K}(\alpha)$ is the ordered set of features \mathcal{F} that will be used to compute the boolean vector representation of each sequence in the following way. Given a sequence $\sigma \in S$, and $\mathcal{F} = \mathcal{K}(\alpha) = \{\omega_i\}_{i=1}^n$ the set of relational k -grams over \mathcal{S} , the *feature vector* of σ is a vector $V_\sigma = (f_1(\sigma), f_2(\sigma), \dots, f_n(\sigma))$ where

$$f_i(\sigma) = \begin{cases} 1 & \text{if } \omega_i \sqsubseteq \sigma \\ 0 & \text{otherwise} \end{cases}$$

Now, the function distance $d_r(\cdot, \cdot)$ between two relational sequences σ_1 and σ_2 is computed using the classical Tanimoto measure [5]:

$$d_{r_1}(\sigma_1, \sigma_2) = \frac{n_{1\sigma_1} + n_{1\sigma_2} - 2n_{1\sigma_{12}}}{n_{1\sigma_1} + n_{1\sigma_2} - n_{1\sigma_{12}}} = \frac{2(n - n_{1\sigma_{12}})}{2n - n_{1\sigma_{12}}} \tag{2}$$

where $n_{1\sigma_i} = n = |\mathcal{F}|$ is the number of the features, and $n_{1\sigma_{12}} = |\{f_i | f_i(\sigma_1) = f_i(\sigma_2)\}|$ is the number of features with the same value in both σ_1 and σ_2 . However, this basic formulation takes into account features not appearing (with value 0) in the sequences, and in case of a lot of feature this can lead to underfitting.

Equation (2) may be extended in the following way:

$$d_{r_2}(\sigma_1, \sigma_2) = \frac{n_{2\sigma_1} + n_{2\sigma_2} - 2n_{2\sigma_{12}}}{n_{2\sigma_1} + n_{2\sigma_2} - n_{2\sigma_{12}}} = \sum_{i=1}^n \frac{f_i(\sigma_1) + f_i(\sigma_2) - 2f_i(\sigma_1)f_i(\sigma_2)}{f_i(\sigma_1) + f_i(\sigma_2) - f_i(\sigma_1)f_i(\sigma_2)} \tag{3}$$

where $n_{2\sigma_i} = \sum_{j=1}^n f_j(\sigma_i)$ is the number of the features holding in the sequence σ_i , and $n_{2\sigma_{12}} = |\{f_i | f_i(\sigma_1) = f_i(\sigma_2) = 1\}|$ is the number of features that hold both in σ_1 and σ_2 .

3.2 Partition Around Medoids Algorithm

Based on an appropriate objective function, a *partitional clustering* algorithm obtains a single partition of n objects into a set of k clusters. To clustering the sequences, we use the well-known k -medoids method Partition Around Medoids (PAM) [6]. Given $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ a set of objects, let $\{u_h\}_{h=1}^k$ be the k cluster representatives, named *medoids*, and l_i be the cluster assignment of an object x_i , where $l_i \in \mathcal{L}$ and $\mathcal{L} = 1, \dots, k$, the goal of the the k -medoids algorithm is to find the best clustering solution \mathcal{C} optimizing the objective function $\mathcal{J}(\mathcal{C})$.

In order to find k clusters, PAM find a representative object u_i (medoid) for each cluster. This representative object is meant to be the most centrally located

object for each cluster. Once the k medoids have been selected, each non-selected object is grouped with the medoid to which it is the most similar. More precisely, if \mathbf{x}_j is a non-selected object, and \mathbf{x}_i is a (selected) medoid, then \mathbf{x}_j belongs to the cluster represented by \mathbf{x}_i if $d(\mathbf{x}_j, \mathbf{x}_i) = \min_{h=1, \dots, k} d(\mathbf{x}_j, \mathbf{x}_h)$, where $d(\mathbf{x}_j, \mathbf{x}_i)$ denotes the dissimilarity, or distance, between objects \mathbf{x}_j and \mathbf{x}_i . PAM starts with an arbitrary selection of k objects. Then in each step, a swap between a selected object \mathbf{x}_i and a non-selected object \mathbf{x}_h is made, as long as such a swap would result in an improvement of the quality of the clustering. In particular, to calculate the effect of such a swap between \mathbf{x}_i and \mathbf{x}_h , PAM computes costs C_{ih} for all non-selected objects \mathbf{x}_j . In this work we used the function reported in Equation (3) as a similarity function in the PAM algorithm.

Tightness. Finally, the quality of the chosen medoids is measured by the average dissimilarity between a non-selected object and the medoid of its cluster, as reported in Equation (4).

$$tightness(\mathcal{C}) = \frac{1}{n} \sum_{i=1, \dots, n} d(\mathbf{x}_i, u_i) \tag{4}$$

3.3 Evaluation of Clustering Solution

To evaluate the goodness of a cluster solution, it is necessary to calculate an intra-cluster similarity (how the objects within a cluster are similar) and inter-cluster similarity (how object from different clusters are dissimilar).

Entropy. Entropy indicates how much homogeneous a cluster is. For each cluster C_j in the clustering result \mathcal{C} we compute p_{ij} , the probability that a member of the cluster C_j belongs to class i as $p_{ij} = n_j^i/n_j$ where n_j is the number of objects contained in the cluster C_j , and n_j^i is the number of data objects of the i -th class that were assigned to the cluster C_j .

The entropy of each cluster C_j may be calculated using the following formula

$$E(C_j) = - \sum_{i=1}^c p_{ij} \log(p_{ij}) = - \sum_{i=1}^c \frac{n_j^i}{n_j} \log \frac{n_j^i}{n_j} \tag{5}$$

where the sum is taken over all the c classes. The entropy of the entire clustering solution is then defined to be the sum of the individual cluster entropies weighted according to the cluster size:

$$E(\mathcal{C}) = \sum_{r=1}^k \frac{n_r}{n} E(C_j) \tag{6}$$

A perfect clustering solution should be the one that leads to clusters that contain objects from only a single class, in which case the entropy will be zero. In general, the smaller the entropy values, the better the clustering solution is.

Purity. The purity measures the extend to which each cluster contained data objects from primarily one class. The purity of the cluster C_j is defined to be

$$P(C_j) = \frac{1}{n_j} \max_i(n_j^i) \quad (7)$$

which is nothing more than the fraction of the overall cluster size that the largest class of objects assigned to that cluster represents. The overall purity of the clustering solution C is obtained as a weighted sum of the individual cluster purities and is given by

$$\mathcal{P}(C) = \sum_{r=1}^k \frac{n_r}{n} P(C_r). \quad (8)$$

4 Experimental Results

Relational Dataset. In order to validate the method we performed experiments on the Greenberg data set [11]. From a behaviour analysis point of view, MASs are in many aspects similar to human's society. Indeed, in order to act effectively an agent (artificially or human) should be able to recognise the behaviours adopted by other agents. Actions taken by an agent are sequential, and thus its behavior can be expressed as a sequence of actions.

A Unix command sequence (session) can be seen as the sequence of actions taken by an agent (user) at each session. The Greenberg data set consists of 168 logs of different users of the unix chs, divided into four groups: 52 *computer scientists* (CS), 36 *experienced programmers* (EP), 55 *novice programmers* (NP) and 25 *non-programmers* (NNP). Each Greenberg's log file corresponding to a user keeps track of an entire login session. Each login session is denoted by a starting and ending time record. Each command belonging to a session, has been annotated with the current working directory, alias substitution, history use and error status. Furthermore, each command name may be followed by some options and some parameters. Each session represents a sequence and a log file is a collections of sequences.

Each shell log has been represented as a set of logical ground atoms [12] as follows: `command(e)` is the predicate used to indicate that `e` is a command. The command name has been used as a predicate symbol applied to `e`; `parameter(e,p)` indicates that `p` is the parameter of `e`. The parameter name has been used as a predicate symbol applied to `p`; `current_directory(c,d)` indicates that `d` is the current directory of the command `c`; `next_c(c1,c2)` indicates that the command `c2` is the direct command successor of `c1`; `next_p(p1,p2)` indicates that the parameter `p2` is the direct parameter successor of `p1`. For instance the following shell log

```

man mklib
man -k mklib
should be translated as
command(c1). '$man'(c1).
next_p(c1,c1p1). parameter(c1p1,'$mklib'). next_c(c1,c2).
command(c2). '$man'(c2).
next_p(c2,c2p1). parameter(c2p1,'$-k').
next_p(c2p1,c2p2). parameter(c2p2,'$mklib').
    
```

Results. The results obtained by the PAM algorithm on a dataset made up of 209 sequences extracted from 8 selected users (agents), two for each class, defined on 153 different command names are reported in Figure 1. In particular, we have 89 sequences for CS, 39 sequences for EP, 39 for NP and 42 for NNP.

In the first step, the set $\mathcal{K}(\alpha)$ of frequent k -grams has been mined. Here, α denotes the support of each k -gram $\sigma \in \mathcal{K}(\alpha)$ corresponding to the ratio $support(\sigma)/|\mathcal{S}|$, where \mathcal{S} is the set of sequences to be clustered. In this experiment, α has been set to 0,02, 0,03, 0,04 and 0,05, and the algorithm extracted, respectively, 567, 153, 84 and 58 k -grams.

Even if we know the corresponding class label of each sequences we dropped it considering all the sequences as belonging to the same dummy class. In the second step, after having extracted the set of features, the sequences have been clustered adopting a values of k (the number of clusters) belonging to the set

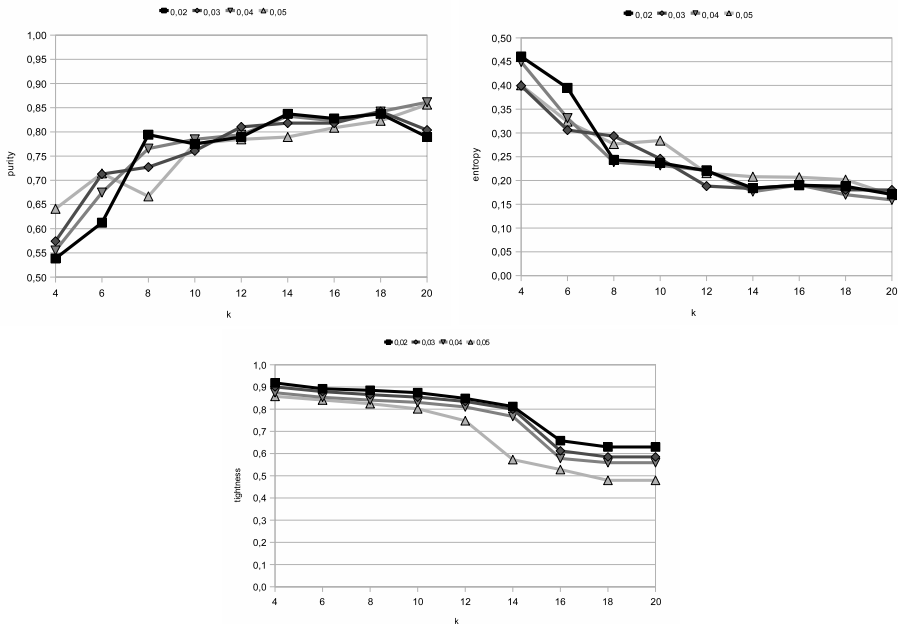


Fig. 1. Entropy (top-left), purity (top-right) and tightness (bottom) values obtained by PAM with different number of cluster

{4, 6, 8, 10, 12, 14, 16, 18, 20}. Finally, we can compute for each obtained cluster the corresponding purity value knowing the previously dropped class label for each sequence. Figure 1 shows the variation of the purity for all clustering solutions, and the corresponding plot of the entropy and tightness values. As we can see, good results have been obtained with large value of k , and choosing α equals to 0,05 corresponding to the case of mining patterns with a large support.

5 Conclusions and Related Works

In [13] the authors investigate the problem of clustering sequence based on their structural features. In order to characterise the structural properties of a given sequence, the authors used the conditional probability distribution (CPD) of the next symbol (right after a segment of some fixed length L). The difference between the two CPDs corresponding to the two sequences is assumed to be the distance between them. The similarity between two CPDs can be measured by the variational distance or by the Kullback-Leibler divergence between the CPDs. The CPDs are represented in a concise way by probabilistic suffix tree, a variant of a suffix tree. The computation of a CPD can be expensive for large L . To define the CPD only the frequent sequences in a cluster are used. To discover clusters is designed an algorithm, it compute the distance of a sequence for each cluster. However, the algorithm reported in [13] works on sequences of flat symbols.

Recent advances in artificial intelligence leading to the growth of structured data sequences and the recent interest in statistical relational learning have motivated the development of probabilistic models for relational sequences [14], such as Logical Hidden Markov Models [15] and Relational Conditional Random Fields [16]. These models have been applied to model the probabilistic nature of the sequence, but they have not been never used for clustering.

In this paper we propose a similarity measure and a clustering approach for relational sequences applied to agent behaviour aggregation. Experimental results proved the validity of the proposed approach. As a future work, we will investigate methods for extracting patterns with a high discriminative power, and we will compare different similarity functions.

Acknowledgment

This work is partially supported by the Italian MIUR-FAR project “Computational Biology Laboratory for Molecular Biodiversity” (DM19410).

References

1. Berkhin, P.: A Survey of Clustering Data Mining Techniques. In: Grouping Multidimensional Data, pp. 25–71. Springer, Heidelberg (2006)
2. Filippone, M., Camastra, F., Masulli, F., Rovetta, S.: A survey of kernel and spectral methods for clustering. *Pattern Recognition* 41(1), 176–190 (2008)

3. Xu, R., Wunsch II, D.: Survey of clustering algorithms. *IEEE Transactions on Neural Networks* 16(3), 645–678 (2005)
4. Dong, G., Pei, J.: *Sequence Data Mining (Advances in Database Systems)*. Springer, Secaucus (2007)
5. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, 2nd edn. Wiley Interscience, Hoboken (2000)
6. Kaufman, L., Rousseeuw, P.J.: *Finding groups in data: an introduction to cluster analysis*. John Wiley and Sons, New York (1990)
7. Ullman, J.D.: *Principles of Database and Knowledge-Base Systems*, vol. I. Computer Science Press, Rockville (1988)
8. Muggleton, S., De Raedt, L.: Inductive logic programming: Theory and methods. *Journal of Logic Programming* 19/20, 629–679 (1994)
9. Esposito, F., Di Mauro, N., Basile, T.M.A., Ferilli, S.: Multi-dimensional relational sequence mining. *Fundamenta Informaticae* 89(1), 23–43 (2008)
10. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.: Fast discovery of association rules. In: *Advances in Knowledge Discovery and Data Mining* (1996)
11. Greenberg, S.: Using unix: Collected traces of 168 users. Research Report 88/333/45, Department of Computer Science, University of Calgary, Alberta, Canada (1988); Includes tar-format cartridge tape
12. Jacobs, N., Blockeel, H., Comma, T.U.: From shell logs to shell scripts. In: Rouveirol, C., Sebag, M. (eds.) *ILP 2001. LNCS (LNAI)*, vol. 2157, pp. 80–90. Springer, Heidelberg (2001)
13. Yang, J., Wang, W.: Cluseq: Efficient and effective sequence clustering. In: *Proceedings of the 19th International Conference on Data Engineering*, pp. 101–112 (2003)
14. Kersting, K., De Raedt, L., Gutmann, B., Karwath, A., Landwehr, N.: Relational sequence learning. In: De Raedt, L., Frasconi, P., Kersting, K., Muggleton, S.H. (eds.) *Probabilistic Inductive Logic Programming. LNCS (LNAI)*, vol. 4911, pp. 28–55. Springer, Heidelberg (2008)
15. Kersting, K., Raedt, L.D., Raiko, T.: Logical hidden markov models. *Journal of Artificial Intelligence Research (JAIR)* 25, 425–456 (2006)
16. Gutmann, B., Kersting, K.: Tildecrf: Conditional random fields for logical sequences. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) *ECML 2006. LNCS (LNAI)*, vol. 4212, pp. 174–185. Springer, Heidelberg (2006)

FutureTrust Algorithm in Specific Factors on Mobile Agents

Michał Wolski¹ and Mieczysław Kłopotek²

¹ Institute of Computer Science, University of Podlasie, ul. Sienkiewicza 51,
08-110 Siedlce, Poland
michal@iis.ap.siedlce.pl

² Institute of Computer Science, Polish Academy of Sciences, ul. J.K. Ordonia 21,
01-237 Warszawa, Poland
klopotek@ipipan.waw.pl

Abstract. In this paper we made a comparative analysis of the well-known reputation systems with our proposal in the light of external factors. We present a new meta-heuristic formula to use in reputation systems and show how different well-known global and local reputation metrics are optimized by it. The presented experiments are to continue our research into the behavior of mobile agents in an open environment.

Keywords: Mobile agents, Open environment reputation, Simulation, Trust.

1 Introduction

On the other hand autonomous agents often need to ask help from other agents in order to achieve their goals, and thus, individual goals can also be achieved by such means as coalition formation, task delegation, cooperation or various other kinds of social interactions. In any multi-agent system, the agents may thus be required to be “social”, that is, to articulate their own behaviors with the behaviors of other agents, based on their representations of the others agents’ behaviors and minds. In this case, there is a need for agents (or another objects in interaction) to have to trust each other. It is important to recognize the enemy as soon as possible.

Due to these needs we developed a metric that allows to determine the trustworthiness quickly. We can define the term *trust* as “the opinion or view of one about something” [1] or as “... a particular level of the subjective probability with which an agent will perform a particular action, both before he can monitor such action (or independently of his capacity to monitor it) and in a context in which it affects his own action” [1].

We described introductory studies on our metrics, in [1], [2], [3], [4]. In this article we describe further investigations on the specific conditions that may prevail in the open environment of mobile agents. We developed a kind of metametric, that we call FutureTrust, that can be combined with traditional metrics in order to increase the

metric learning rate. We have applied the FutureTrust approach to studied global and local metrics.

We concentrate in this paper on our results concerning the influence of factors such as:

- Test of metrics' behavior depending on values of threshold trust
- Test of metrics' behavior depending on transaction probability values

In the second section we present a group of agents which are the subject of our research. In the third section we present our optimizing-formula. Fourth and fifth sections are devoted to experimental investigations and their summary.

All the results of experiments, presented in this work, have not been previously published.

2 Agents

In the studies of Aberer [5], Buchegger [6], Kamvar [7], Michiardi [8], Xiong [9] and Yu [10] several different types of the agents' behaviors may be found. The most common are 7 kinds of archetypes: Honest, Malicious, Evil (Conspirative), Selfish, Disturbing, Fugitive, Harmonizing.

An Honest agent initiates only good transactions, as, regardless of the type of transactions, it will cooperate in any event, offering what a partner expects from it.

A Malicious agent randomly initiates good, neutral, as well as bad transactions. It tries to destroy the system with its behaviour, through always negative assessment of transactions.

A Disturbing agent tries to build its reputation through good transactions (it behaves like an Honest agent). When it obtains high level of reputation it abuses this trust and then carries out bad transactions (just like a Malicious one), as long as its value of reputation has not decreased too much. Then it again carries out good transactions, until it has rebuild sufficiently high reputation, so as to change its conduct again. Such a behaviour may simulate sudden changes in the network, as well as variable security policy.

The above three archetypes are characteristic for both global and local reputation systems. All the 7 kinds of archetypes are described more fully in [2], [12], [13], [14], [17].

3 Research

Let us imagine the following setting. There is an infrastructure of hosts (“infrastructure agents”) that mobile agents cooperate with in order to achieve their goals. The host behavior belongs to categories described in the preceding section. The mobile agents form families. By an agent family we understand a group of agents that have:

- a common producer,
- a base node where they can communicate each other,
- common goal for work,
- common reputation metrics to recognize nodes.

When agents begin to inspect open environment they don't know anything about network topology, node hostility or friendliness. It is not so important for an individual agent to survive, but rather the important factor is the capability of survival of a large portion of them. So their collective capability to identify and avoid badly behaving hosts is of primary interest. Note that the trust level is not a complete characterization of survival capability, because the trust level; exceeding some threshold decides on possibility of interaction or its exclusion, and hence the potential risk of damage. Nonetheless the collected experience is an important factor.

A number of trust systems have been introduced in the past, that differ in their capability to learn the intrinsic node behavior. In this paper we introduce a new one and compare its performance with 2 known base trust system algorithms. We verify by simulation how agents family based on given trust system can explore unknown network.

The trust metric proposed in this paper is of generic nature in the following way: By using a traditional metric, agents of a family encode their experience with the network in terms of the trust levels assigned to host nodes. We claim that the new (FutureTrust) generic metric can accelerate this process, that is the experience is collected earlier, which is vital for survival.

The basis for the assumptions for formula enabling determination of probable value of trust in time is a statement that in an unknown environment we cannot predict a node with which an agent will meet. The node may be positively disposed towards the agent and carry out a transaction with it or allow it to work in accordance with a set task. There is also a probability that an encountered node will be willing to interfere with agent's internal structure to change or destroy it, namely it will be definitely hostile. This randomness in the results of encounters between an agent and a node, which in the least expected time may result in a destruction of an agent, caused that previously known measures of trust could become insufficient to describe the open environment. Basing only on the experience from previous interactions in the constantly changing "world" may appear insufficient. Also for this reason, it has become crucial to find such a method of trust determination that at small quantity of data – small quantity of experience, would enable moderately accurate determination of the actual measure of trust.

The outlined situation is quite similar to the situation that we have on the stock exchange, where at a very small quantities of data – historical rate of shares - one has to determine (predict) the future rate. Hence, there is some analogy between quite unpredictable shaping of rate of securities and shaping of trust in the open (unpredictable) environment of mobile agents.

When we talk about open environment or open systems we think about one of scenarios mentioned below:

- a system open in the sense that agents can enter and leave at any given time. This means that an agent could change its identity on re-entering and avoid punishment for any past wrongdoing
- a system that allows agents with different characteristics (for example, policies, abilities, roles) to enter it and interact with each other
- no agent can know everything about its environment

In connection with presented observations, it seems purposeful to apply mathematic mechanisms used in financial engineering to determine future values of trust (so-called Black-Scholes model). These mechanisms, adapted to conditions and parameters which occur in the open environment of mobile agents, should enable determination of the future value of trust.

Reputation metrics can be subdivided into two major groups: local and global ones. In global reputation system, there is only a single reputation value per node that is stored in a common repository [15]. Local reputation systems provide different reputation values for a node depending on the current position of agent(s) in the network. In our research we use mobile agent's in open environment.

In our work we explore different types of reputation systems. The main groups of metrics are Beta systems and Weighted systems. The aforementioned group of metrics are described more fully in [12] and [13].

The effectiveness of a reputation system and its metric depends on its resistance against several types of attacks. The success of non-honest agents is the measurement for the quality of the metric. The different agent types implemented for the simulation are called *honest*, *malicious*, *evil*, *selfish*, and *disturbing*. These types differ in their behavior when transacting and rating. But they have in common, that their decision whether they are willing to transact with another peer or not, is based on the reputation of this peer. Thus we first present our model for acceptance behavior and after that the differences between the agent types.

In our research we consolidate known reputation metrics with stochastic process, which can forecast reputation with particular probability in defined time in future. We sought to minimize risk relevant to forecasting of trust value and we want to answer the question: "What trust value will have this family of nodes in the next iteration, next 10, 100 iterations?" To come at a solution, we use two simplifying assumptions: trust value (comes from nodes behavior) is similar to random walk - simple stochastic process - reputation in short time period has normal distribution characteristic and for any time in the future reputation has log-normal distribution character.

Based on log-normal distribution we can forecast future value of trust (equation 1)¹

$$\ln(R_T) \approx \phi[\ln(R) + \mu \cdot T, \mu\sqrt{T}] \quad (1)$$

T – time (number of iterations)

R – present reputations (computed by known trust metrics)

R_T – future reputation (in T iterations)

μ – variable responsibility for fluctuation of trust metrics

φ – normal distribution

Before we compute future trust value we have to store information about positive and negative transactions. We do it by storing necessary data in table [idnode, good, bad]. We add new value in this table [idnode, 1,0] for positive transaction and [idnode, 0,1] for other situations.

¹ In other words, we treat R_T as a random variable.

Based on equation 1 we can compute FutureTrust (equation 2)

$$R_t = \begin{cases} e^{\ln(R) + \mu T + c \cdot \sqrt{\mu T}} \sum bad < \sum good \\ e^{\ln(R) + \mu T - c \cdot \sqrt{\mu T}} \sum bad > \sum good \end{cases} \quad (2)$$

Where c – value of accepted deviations (e.g. if confidence=95% then $c=1.96$)

Bad – amount of negative transaction

Good - amount of positive transaction

A number of properties of the FutureTrust measure have been studied and compared with known reputation systems, of metrics in [1], [2], [3], [4]. The present study focuses on various aspects of environment and agents themselves.

In order to test resistance of reputation metrics to attacks by the agents, several parameters should be examined. In the present study the following indicators have been studied: average reputation (average trust), number of transactions, benefits from transactions, number of agents

We collect separate statistics for interaction with representatives of the different types of nodes of the infrastructure, mentioned above.

Whenever we talk about the *average reputation*, we refer to the mean value of all reputations of node agents from a specific type. We store the correct rating an agent should receive for each transaction.

The purpose of this experiments are comparison of known metrics of reputation with the suggested optimization formula FutureTrust.

In our research we test the effectiveness of trust algorithms by simulating an environment adhering to some predefined model, which is unknown for mobile agents. The mobile agents move from one node to another. During its journey an agent interacts with nodes and learns their behavior over a number of encounters. Knowledge about node behavior will have to be stored in common repository, in form of a “reputation level”, which is then compared to the “intrinsic” one² (the one from the predefined simulation model). Each algorithm has been tested on the same network, created in a random way, with topological features similar to the Internet. For global and local metrics one has compared FutureTrust metric with traditional, non-modified metrics in specific factors such as:

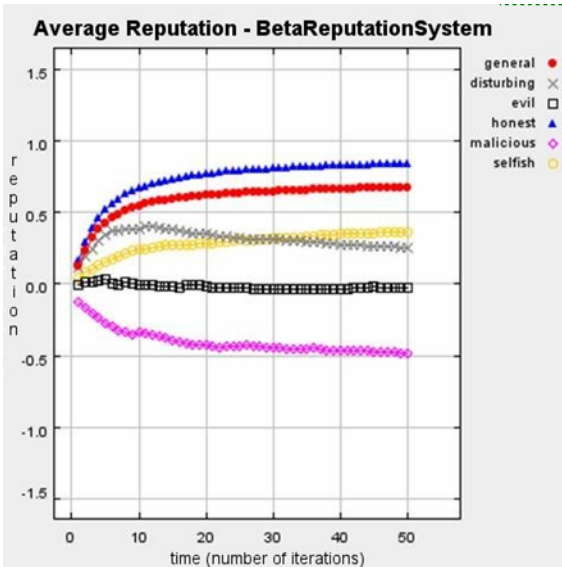
- test of metrics' behaviour depending on values of threshold trust
- test of metrics' behaviour depending on transaction probability values

We will concentrate on changes of average reputation in above factors.

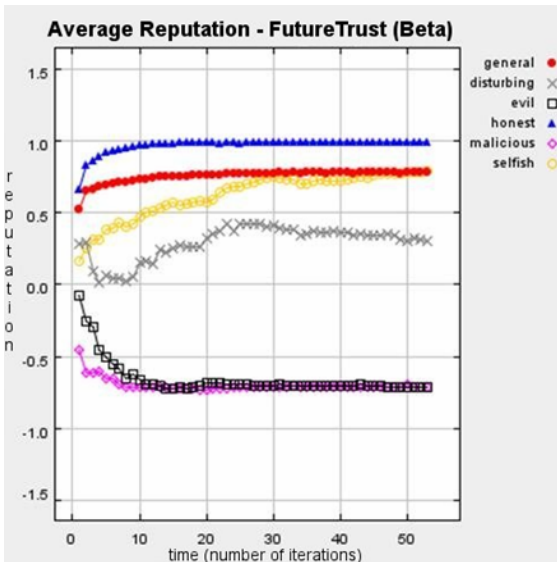
4 Results of Experiments

In this paper we compare FutureTrust metrics with one global metric (BetaSystem [16]) and one local metric (Weighted Advanced Passing[13]). The results seem to be representative for the body of comparative experiments with other trust metrics that we performed.

² By an intrinsic reputation level we mean the reputation level that would be attained by the node if it could be traced for a sufficiently long time under the given reputation model[18].



a)



b)

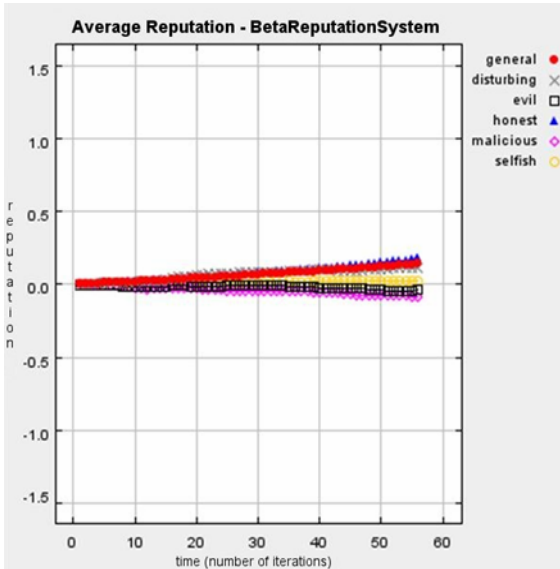
Fig. 1. Average trust for BetaSystem measure (a) and FutureTrust (BetaSystem) measure (b) at trust threshold of 0.1

For easier comparison we assume that the reputation level computed for each metric can range from -1 to 1, and we rescaled or shifted the respective scales, whenever necessary.

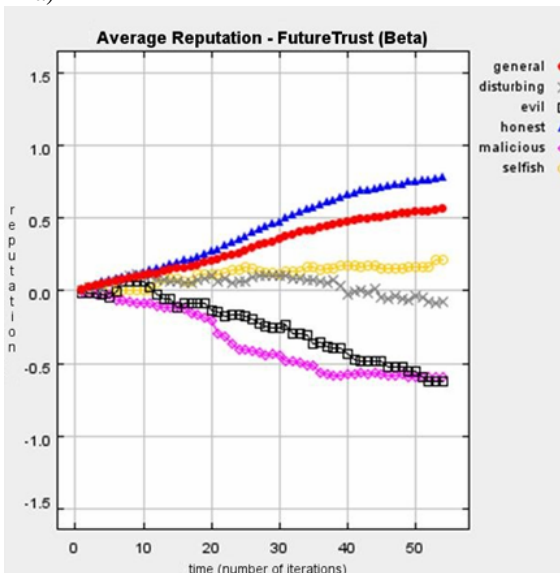
4.1 Test of Metrics' Behaviour Depending on Values of Threshold Trust

From the point of view of agents moving either within local or within open environment it is important to determine proper trust threshold that specifies barrier of transaction acceptance. The purpose of this experiment is to find out how value of the actual trust threshold affects behaviour of agents. The trust threshold in the simulation environment may have values from 0 to 1, with assumption that the smaller values, the earlier accepted value is obtained. The tests were carried out for original BetaSystem and Weighted Advanced Passing metrics, as well as FutureTrust modifications. Experiments were conducted at values of trust threshold equal to 0.1; 0.5; 0.9.

Our earlier studies of BetaSystem system, [1], [2], [3], [4] were confined to the default threshold of 0.5. At reduced values of trust threshold (Fig.1) it can be noted that the described global system of BetaSystem reputations does not operate.



a)



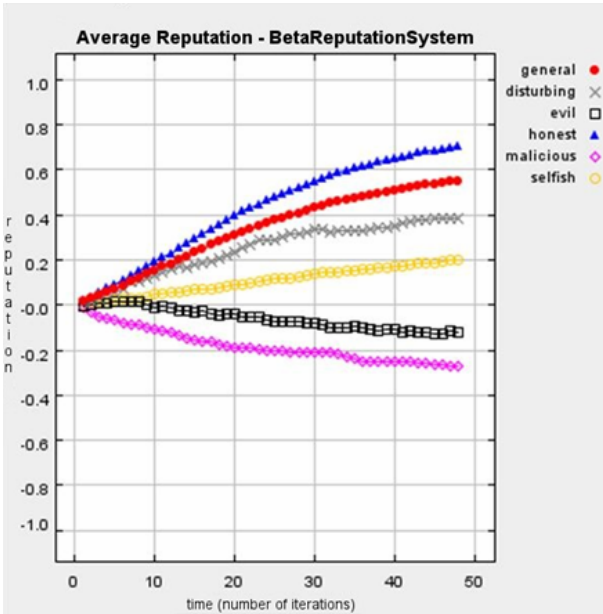
b)

Fig. 2. Average trust for BetaSystem measure (a) and FutureTrust (BetaSystem) measure (b) at trust threshold of 0.9

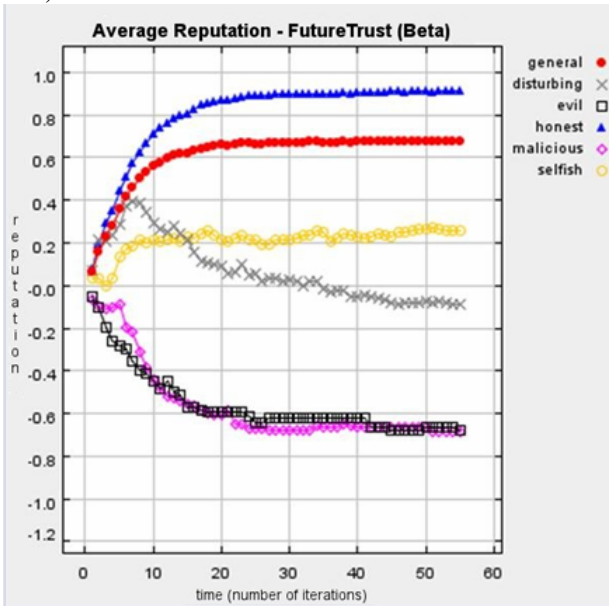
Properly and does not recognize “intrinsic reputation” of node agents of Evil type (Fig.1a), as the average value stays “neutral” over time. Whereas, application of optimization FutureTrust measure (Fig.1b) restores relevant proportions in assessment of positive and negative node agent behaviours (the evil agents loose their reputation over time). The exception is behavior of Disturbing agents, which is overestimated. It is probably caused by a high degree of transaction acceptance, which give them much space for loosing and restoring reputation without punishment.

At increase in trust threshold one may note (Fig.2a) that BetaSystem metric aims very slowly to the expected value of trust for the different types of agents. Application of FutureTrust measure enables to shorten time that is needed for a proper assessment of a particular type of agent's behaviour.

The studies on behaviour at the default trust threshold (0.5) have been published earlier (see [1], [2], [3], [4]).



a)



b)

Fig. 3. Average trust for BetaSystem measure (a) and FutureTrust over BetaSystem measure (b) at transaction probability of 25%

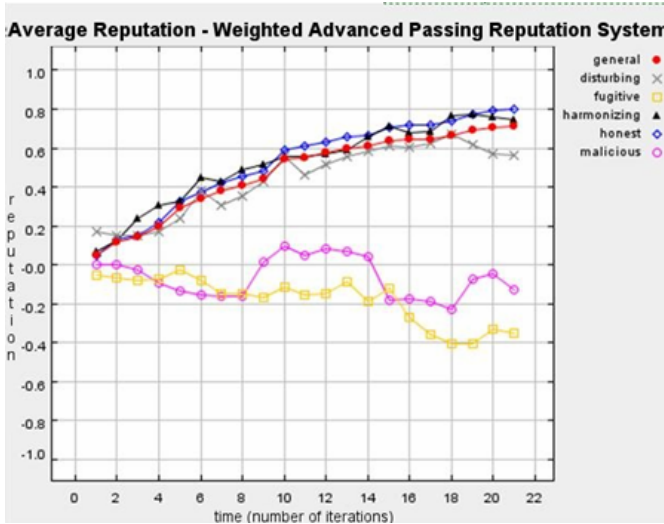
As a result of conducted experiments relating change in values of trust threshold for Beta-System one may notice that reduction of threshold increases average value of trust for this measure, while an increase reduces it.

Application of FutureTrust measure allows more precise determination of proper trust for different types of node agents. Additionally, in the case of high value of trust threshold, FT metric enables faster obtaining of proper trust value for the particular type of agent's behaviour.

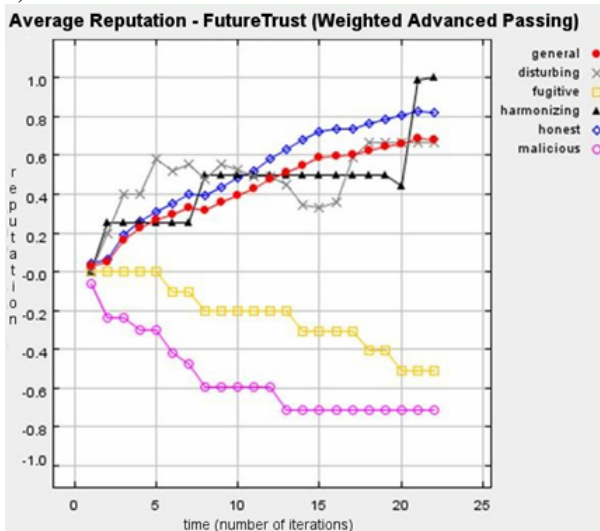
Results for WAP were similar.

4.2 Test of Metrics' Behaviour Depending on Transaction Probability Values

The probability of transaction between an agent is another element of tests that determines pace in which an agent becomes familiar with the surrounding reality. The smaller value, the less frequently an agent enters in interaction with another agent and, at the same time, its effectiveness is smaller.



a)



b)

Fig. 4. Average trust for Weighted Advanced Passing (WAP) (a) and FutureTrust over WAP (b) measures at the probability of transactions of 25%

In [1-4], the default settings of the environment value of probability is set at level of 75% were studied. In the current experiment the probability of transaction was lowered to 25%. On the basis of conducted experiments (Fig. 3) it can be seen that decrease in transaction probability lowers significantly capacity of Beta-System measure to quickly recognize the “intrinsic” reputation of node agents. Whereas, the application of FutureTrust measure restores speed of actions to Beta-System metric. Acceleration of measure's actions is nearly 50-fold.

With regard to local metric (WAP) reduction of probability of transactions (Fig. 4) will not cause significant changes beyond increase in medium trust for algorithms of Disturbing type. On the contrary, the application of FutureTrust measures for tested Advanced Passing Weigheted algorithm causes substantial reduction in average values of trust for Fugitive and Malicious agents.

5 Conclusion

As shown, by combining the proposed FutureTrust formula with BetaSystem measure not only do we achieve trust forecasts, but also get better distinction between agents of various behavioural types.

Experiments related to acceptance threshold proved that the application of the FutureTrust optimization formula in global reputation systems in the case of low and high threshold of trust enable to improve results of a base measure. The reason for this state of affairs is the fact that at high acceptance threshold for transactions first of all mechanisms which "algorithmize" FT measure operate and at low threshold of acceptance for transactions forecasting mechanism operates. FutureTrust formula operates in a similar way during optimization of local reputation systems.

To sum up, the thesis formulated at the beginning of the study that the application of trust forecasts based on Black-Scholes model enables faster recognition of real trust of objects present in the open environment of mobile agents is correct.

In the further studies one should check how other, perhaps new trust measures perform the role of base measures for FutureTrust formula.

Another problem to be solved is a situation in which data provided in the course of a transaction for many reasons may not be assigned from an agent- bidder directly to an inquiring agent which can make it impossible to determine right level of reputation. Finally, the suggested forecasting measure needs to be implemented in real operating mobile agents so as to check the to what extent the simulations studies apply to real world circumstances.

References

- [1] Wolski, M., Kłopotek, M.A.: A Concept of Reputation for Mobile Agents Environments, Świnoujście (2006)
- [2] Wolski, M., Kłopotek, M.A.: Future Trust Forecast in Open Mobile Agent Environment (2007)
- [3] Kłopotek, M.A., Wolski, M.: Comparative Study of Trust Algorithms for Mobile Agent in Open Environment. Proceedings of Artificial Intelligence Studies 3(26) (2006)
- [4] Kłopotek, M.A., Wolski, M.: Simple Reputation Metrics for Mobile Agent in Open Environment. In: Proceedings of the International Multiconference on Computer Science and Information Technology, vol. 1 (2006)
- [5] Aberer, K., Despotovic, Z.: Managing Trust in a Peer-2-Peer Information System. In: Proceedings of the Tenth International Conference on Information and Knowledge Management (2001)
- [6] Buchegger, S., Le Boudec, J.-Y.: A Robust Reputation System for Mobile Ad-hoc Networks, in Technical Report, Switzerland (2003)
- [7] Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The EigenTrust Algorithm for Reputation Management in P2P Networks. In: Proceedings of the Twelfth International World Wide Web Conference (2003)
- [8] Michiardi, P., Molva, R.: CORE: A Collaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks, in Research Report RR-02-062 (2001)
- [9] Xiong, L., Liu, L.: PeerTrust: Supporting Reputation-Based Trust in Peer-to-Peer Communities. IEEE Transactions on Knowledge and Data Engineering. vol. Special Issue on Peer-to-Peer Based Data Management (2004)

- [10] Yu, B., Singh, M.P.: A social mechanism of reputation management in electronic communities. In: Klusch, M., Kerschberg, L. (eds.) CIA 2000. LNCS (LNAI), vol. 1860, pp. 154–165. Springer, Heidelberg (2000)
- [11] Feldman, M., Papadimitriou, C., Chuang, J., Stoica, I.: Free-Riding and Whitewashing in Peer-to-Peer Systems. In: Proceedings of ACM SIGCOMM 2004 (2004)
- [12] Schlosser, A., Voss, M., Brückner, L.: A Workshop on Reputation in Agent Societies as part of 2004 IEEE/WIC/ACM International Joint Conference on Intelligent Agent Technology (IAT 2004) and Web Intelligence, WI 2004 (2004)
- [13] Schlosser, A., Voss, M., Brückner, L.: On the Simulation of Global Reputation Systems. *Journal of Artificial Societies and Social Simulation* 9(1), 4 (2005)
- [14] Zacharia, G., Maes, P.: Trust Management through Reputation Mechanisms. *Applied Artificial Intelligence* (2000)
- [15] Sabater, J., Sierra, C.: Reputation and Social Network Analysis in Multi-Agent Systems. ACM, New York (2002)
- [16] Josang, A., Ismail, R.: The Beta Reputation System. In: Proceedings of the 15th Bled Conference on electronic Commerce (2002)
- [17] Huynh, D., Jennings, N.R., Shadbolt, N.R.: Developing an Integrated Trust and Reputation Model for Open Multi-Agent System. Springer, Heidelberg (2006)
- [18] Wolski, M., Kłopotek, M.: Mobile Agent Reputation Based on Future Trust on Local and Global Environment. In: Proceedings of 17th International Conference Intelligent Information Systems (IIS), Krakow (2009)

Ensembles of Abstaining Classifiers Based on Rule Sets

Jerzy Błaszczyński, Jerzy Stefanowski, and Magdalena Zajac

Institute of Computing Science, Poznań University of Technology,
60-965 Poznań, Poland

{jerzy.blaszczyński,jerzy.stefanowski,magdalena.zajac}@cs.put.poznan.pl

Abstract. The role of abstaining from prediction by component classifiers in rule ensembles is discussed. We consider bagging and Ivotes approaches to construct such ensembles. In our proposal, component classifiers are based on unordered sets of rules with a classification strategy that solves ambiguous matching of the object's description to the rules. We propose to induce rule sets by a sequential covering algorithm and to apply classification strategies using either rule support or discrimination measures. We adopt the classification strategies to abstaining by not using partial matching. Another contribution of this paper is an experimental evaluation of the effect of the abstaining on performance of ensembles. Results of comprehensive comparative experiments show that abstaining rule sets classifiers improve the accuracy, however this effect is more visible for bagging than for Ivotes.

1 Introduction

In recent years there has been much research on multiple classifiers also called ensembles of classifiers. Various approaches have been proposed to construct them with respect to either a phase of generating component classifiers or a phase of aggregating predictions of these classifiers (see [8] for a review).

We are particularly interested in ensembles containing classifiers based on *sets of rules* induced from diversified training samples. This interest results from previous research of two authors on various rule induction algorithms [13,14]. Furthermore, one can notice that *rule ensembles* such as SLIPPER or LRI proved to be competitive classifiers to more popular decision tree ensembles, see e.g., [4,12,15]. A rule classifier has an interesting property. Let us note that a rule assigns class only to these objects that it covers. Moreover, it covers only a bounded part of problem space as opposed to a decision tree. This property makes a rule classifier interesting to concern the research topic of this paper. Namely, studying changes in the aggregation phase of the ensemble when some component classifiers may *abstain* from predicting class labels.

A set of rules does not have to maintain the ability to refrain that is typical for a single rule. Most rule sets classifiers are designed to always assign a class label for a new object, e.g., by using ordered priority lists of rules with a default class label [11] or specialized strategies for solving ambiguous conflicts

with unordered rules [6]. However, there exist solutions where the classifier may not produce its class prediction in case of uncertainty as to the classified objects. Such classifiers called *abstaining classifiers* have already been studied in the framework of ensembles. Most of the research concerns refraining from the final decision in case of disagreement between votes of component classifiers, e.g., see a study [12] showing that it may improve the final accuracy. Some researches allow single classifiers to give no answer. For instance, rule ensembles like SLIPPER [4] are based on a weighted combination of *single rules* (being component classifiers) and a rule is excluded from voting if the new objects is not covered by it. However, according to our best knowledge there are no similar abstaining solutions for ensembles where component classifiers are based on *sets of unordered rules* induced by sequential covering algorithms (which are the most popular techniques for inducing rules [7]).

Therefore, the first aim of our paper is to present a framework for constructing such an ensemble of component classifiers based on rule sets that abstain when no rule is covering classified object. Motivation behind it is that each set of unordered rules usually covers a subspace in the problem space which can be seen as an area of its expertise. Thus, it is more likely that if one classifier abstains from classifying an object, other more expertised sets of rules should classify the object. To achieve this aim, we propose to use sets of rules induced by the MODLEM algorithm [13]. MODLEM proved to be competitive to other rule and tree classifiers and was successfully applied inside pairwise coupling and bagging [14]. Moreover, it naturally joins with classification strategies based on matching a description of a classified object to rules. Such strategies can be easily adopted to abstaining from a class prediction. To become independent of one specific solution we choose two different classification strategies: the first, introduced by Grzymala, based on rule support [6] and the other proposed by An [1], which employs a rule discrimination measure. Following similar motivations we decide to investigate two different approaches to constructing the ensemble: *bagging* [2] based on bootstrap sampling and *Ivotes* [3] using a sequential adaptive approach.

The second, not least important, aim of our paper is to experimentally evaluate the influence of abstaining component classifiers based on rule sets induced by MODLEM on the final accuracy of the ensemble. Although following some related results one could expect an improvement of classification accuracy, our contribution include a comprehensive comparative study on several benchmark data sets, where we want to examine more deeply the degree of changes with reference to different classification strategies and different ensembles.

2 Related Research

The idea of classifiers refraining from class predictions has been considered in machine learning, in particular in cases when classification is uncertain. The classified object may be located either in the boundary between classes or very far from any class. Some techniques as threshold classifiers producing distributions of membership to several class, e.g., neural networks, may naturally abstain

from classification if none of predictions exceeds a preferred threshold. Such an unknown decision may be suitable in some domains, e.g., in medical diagnosis.

The concept of abstaining in ensembles of classifiers is considered at two levels. At the first level, abstaining occurs in the final decision of the ensemble. In this case, the ensemble may abstain from classification for uncertain objects, which are characterized by the smallest difference between the number of indications for the most often predicted class label and the number of indications for the second predicted label. Such a kind of abstaining classifier has been analysed in [12] showing theoretically that it may improve PAC bounds of error for rule ensembles. The same work also contains preliminary experiments with the authors' proposal of stochastic algorithm showing benefits of abstaining from making uncertain predictions and comparing it to bagged versions of popular rule induction algorithms RIPPER and PART. A method for optimization of similar abstaining classifiers using ROC analysis was presented in [10]. Moreover, this work contains an interesting review of other previous works on refraining from classification. As they are not directly related to our research, we skip them. At the second level, component classifiers may refrain from prediction. This has been postulated in [4,5]. However, the result of abstaining at this level on the accuracy of the ensemble is, to our best knowledge, not deeply studied.

The most similar ensemble to presented in our paper is the one produced by SLIPPER [4]. More precisely, such a conclusion can be drawn with respect to presented further ensemble of rule sets classifiers produced by Ivotes. Nevertheless, SLIPPER is different to the approach we take to create an ensemble. It uses a single rule as a component classifier while we use a set of rules. Moreover, it aggregates component classifiers by a linear combination. We apply majority voting between component classifiers and we use mentioned before classification strategies inside each of component classifiers.

3 Our Framework for Abstaining Rule Ensembles

To construct ensembles we first chose *bagging*. It was introduced by Breiman [2] with a key concept of bootstrap sampling, i.e., uniform sampling with replacement objects, from the original learning set. Having several independent bootstrap samples, a set of classifiers is generated by the same learning algorithm and the final decision is formed by aggregating predictions of classifiers with the majority equal weight voting scheme.

The second considered ensemble comes from the *Pasting Small Votes* idea. Its original motivation was handling massive data which does not fit into computer's memory. Breiman proposed using the so-called pasting votes where many component classifiers are trained on relative small subsets of the original training data [3]. He introduced two strategies for implementing this idea: *Rvotes* and *Ivotes*. In *Rvotes* training sets are sampled randomly from large data sets (similarly to *bagging*). *Ivotes* sequentially generates small data sets using the *importance* sampling. According to this sampling each new training data sample should have approximately 50% objects that were misclassified by the ensemble

including previously generated classifiers. The content of the particular training data sample for each subsequent classifier relies on sampling with replacement where the sampling probability results from the out-of-bag estimate [3]. We chose Ivotes as it is more similar to boosting idea and may be more accurate than standard bagging [8].

We decided to generate sets of rules by the MODLEM algorithm, which was originally introduced by Stefanowski in [13]. Due to the space limit we skip its more precise presentation (see [14] for details). Briefly speaking, it is based on the scheme of a *sequential covering* and it generates an *unordered minimal set of rules* for every decision concept. It is particularly well suited for analysing data containing a mixture of numerical and qualitative attributes, inconsistent descriptions of objects or missing attribute values. Searching for the best single rule and selecting the best condition is controlled by a criterion based on a modified entropy measure. As it will be further explained, MODLEM unordered sets of rules are better suited to introduced abstaining with partial matching strategies than ordered lists of rules.

Let us remind that both considered ensembles are based on manipulating presence of objects in bootstrap samples to produce *diversified training* samples. MODLEM is an *unstable algorithm* in the sense of Breiman's postulate [2], i.e., small perturbation of data may results in large changes in the induced rules, see also [14]. This is a desirable property for ensembles like bagging. Using unpruned structure should increase diversity of component classifiers, as it was also noted by Breiman and others [9].

The set of induced rules needs to be combined with a specific *classification strategy* to constitute a classifier. Most of these strategies are based on matching the new object's description to condition parts of rules. If rules are ordered into a priority list (as it is done in e.g., in popular C4.5 rules [11]; another kind of exception list with default rule is used in RIPPER) the first matched rule from the list is "fired" to classify a new object. Unlike this option, in our case the set of rules is *unordered* and all rules are tested for matching. This may lead to three situations: a unique match (to one or more rules from the same class); *matching more rules* from different classes or *not matching* any rules at all. In both last situations a suggestion is ambiguous, thus, proper resolution strategy is necessary. We skip descriptions of some early proposals of solving it, e.g., by Michalski in AQ family or Clark et al. in CN2. Review of the different strategies, which could be combined with MODLEM is given in [14].

For our experiment we choose the strategy introduced by Grzymala-Busse in [6] as it has been successfully applied in many experiments. Briefly speaking it is based on a voting of matching rules with their supports. The total *support* for a class K is defined as: $sup(K) = \sum_i^m sup(r_i)$, where r_i is a matched rule that indicates K , m is the number of these rules, and $sup(r)$ is the number of learning objects satisfying both condition and decision parts. A new object is classified to the class with the highest total support. In the case of not-matching, so called *partial matching* is considered where at least one of rule conditions is satisfied by the corresponding attributes in the new object's description x . In this case,

a matching factor $match(r, x)$ is introduced as a ratio of conditions matched by the object x to all conditions in the rule r . The total support is modified to $sup(K) = \sum_i^p match(r, x) \times sup(r_i)$, where p is the number of partially-matched rules, and object x is assigned to the class with its highest value.

As an alternative strategy we apply proposal of Aijun Ann [11] because it also considers partial matching and its experimental verification with ELEM algorithm (simpler sequential covering than in MODLEM) showed that it is competitive to C4.5 rules and CN2. It uses a rule quality measure different than the rule support, i.e., a *measure of discrimination*: $Q_{MD} = \log \frac{P(r|K) \times (1 - P(r|\neg K))}{P(r|\neg K) \times (1 - P(r|K))}$, where P denotes probability. For more technical details of estimating probabilities and adjusting this formula to prevent zero division see [11]. Its interpretation says that it measures the extend to which rule r discriminates between positive and negative objects of class K . Inside the ELEM2 classification strategies it is used in similar formulas for decision scores as in the Grzymala's strategy - the only difference concerns putting Q_{MD} in place of $sup(r)$. Therefore, the difference between classification strategies is choosing another rule quality measure.

We propose to adopt both strategies to the abstaining from prediction by *switching off the partially matching* phase. It corresponds to the fact the induced rules establish an area of expertise for a classifier (i.e., a subspace of problem space that is covered by the rules). If an object completely matches a rule, it may be treated as being close to this area. Otherwise, in case when it is not matched by any rule, it is far from the area of expertise and it can be classified as unknown. Moreover, assuming that classifiers are generated from diversified samples it is more likely that their areas of expertise do not overlap. This should result in an ensemble of experts being able to classify new objects better than any of component classifiers.

4 Experiments

The main goal of experiments is to evaluate the influence of abstaining of component classifiers on the final accuracy of the ensemble. As we discussed in the previous section, we are conducting experiments for two different approaches to construct ensembles: bagging and Ivotes. Moreover, we study the use of two different classification strategies with matching object's descriptions to rule (either An's proposal with discrimination measure and Grzymala's proposal of using rule support). We additionally carried out the experiments for the MODLEM classifier, to show that these classification strategies are useful for working with the single classifier. In all versions the classifiers were based on unpruned sets of rules induced by MODLEM - it was always induced with standard options as described in [14].

A number of classifiers in bagging ensemble was 20 because we noticed that Ivotes ensemble usually consisted of less but close to 20 base classifiers. Size of the learning sample used by Ivotes algorithm was set to 50%. This value was chosen because of the data sets used in experiments. As they were not so big as idea of Pasting Small Votes assumes, we have to keep reasonable size of training

Table 1. Characteristics of data sets

Data set	Objects	Attributes	Classes
breast-w	699	9	2
bupa	345	6	2
credit-german	1000	20	2
crx	690	15	2
diabetes	768	8	2
ecoli	336	7	8
glass	214	9	7
heart-cleveland	303	13	5
hepatits	155	19	2
ionosphere	351	34	2
pima	768	8	2
sonar	208	60	2
vehicle	846	18	4
vowel	990	13	11

sample. Moreover, one of the author's experience shows that default version of Ivotes classifier has good results on smaller data sets when size of the learning sample is higher than 40%. They also showed that Ivotes is competitive with the standard bagging when size of the learning sample is set close to 50%.

All experiments were carried out on 14 data sets from the UCI repository¹. Their characteristics are given in Table 1. We chose them because they were often used by other researchers working with rule ensembles.

The classification accuracy was estimated by the stratified 10-fold cross-validation, which was repeated several times. Tables with results always contain an average classification accuracy with a standard deviation. Moreover in brackets we present a rank of the best performance among all variants of classifiers for the given data set (the smaller, the better). We show them because they are used in the statistical test further described. Last row of each table shows an average rank scored by a given classification strategy.

In the first experiment we evaluate the use of both classification strategies in the single classifier based on MODLEM induced rules. As the full usage of mentioned classification strategies is a natural mechanism of improving final accuracy of a single rule classifier, these results do not contribute to the abstaining ensembles but they are given for comparison reasons only. In Table 2 we show two variants of single classifiers: (1) use of complete strategies with partial matching (which is called no abstain), (2) classification without partial matching (called abstain). The second experiment concerns using abstaining inside ensembles of MODLEM based classifiers. Results of this experiment are presented in Tables 3 and 4.

We use a statistical approach to compare difference in performance between classifiers in variants which we mentioned above. First, we apply Friedman test

¹ See <http://www.ics.uci.edu/~mlern/MLRepository.html>

Table 2. Accuracy of a single classifier with different classification strategies

Data set	Classification strategy			
	Discrimination measure		Rule support	
	abstain	no abstain	abstain	no abstain
breast-w	92.73±1.00 (3)	94.71±0.67 (1)	92.53±1.06 (4)	93.88±0.70 (2)
bupa	59.88±2.00 (4)	66.96±2.64 (2)	60.41±1.70 (3)	68.35±1.95 (1)
credit-german	62.72±1.41 (4)	68.26±1.51 (2)	63.56±1.51 (3)	71.26±0.92 (1)
crx	77.28±0.87 (4)	81.97±1.22 (2)	77.33±0.76 (3)	83.33±0.90 (1)
diabetes	64.17±1.29 (3)	70.73±0.90 (2)	63.85±1.12 (4)	71.20±0.87 (1)
ecoli	74.52±1.48 (3)	77.56±1.65 (1.5)	74.05±1.74 (4)	77.56±1.42 (1.5)
glass	62.71±2.18 (3)	70.28±2.10 (1)	61.96±2.60 (4)	70.09±1.72 (2)
heart-cleveland	71.42±2.03 (4)	76.83±1.71 (2)	71.75±2.09 (3)	77.76±1.79 (1)
hepatitis	56.65±3.32 (3)	70.19±2.53 (2)	56.39±2.96 (4)	80.90±0.77 (1)
ionosphere	88.21±1.59 (3)	90.20±1.43 (2)	87.81±1.19 (4)	90.83±0.58 (1)
pima	64.87±1.32 (3)	71.95±1.41 (2)	64.61±0.88 (4)	72.01±0.98 (1)
sonar	67.21±2.16 (4)	75.77±3.38 (1)	67.79±1.72 (3)	75.00±1.01 (2)
vehicle	66.50±0.80 (3)	71.39±1.03 (1)	66.19±1.04 (4)	67.54±1.06 (2)
vowel	75.076±0.67 (3)	75.80±0.57 (2)	74.63±0.84 (4)	76.14±0.76 (1)
average rank	3.36	1.68	3.64	1.32

Table 3. Comparison of using different classification strategies in Bagging

Data set	Classification strategy			
	Discrimination measure		Rule support	
	abstain	no abstain	abstain	no abstain
breast-w	96.28±0.52 (1)	96.17±0.28 (2)	96.08±0.45 (3)	95.34±0.29 (4)
bupa	73.28±1.15 (3)	73.51±1.74 (2)	73.10±1.57 (4)	74.67±0.75 (1)
credit-german	76.10±0.99 (2)	75.26±0.74 (4)	76.30±0.55 (1)	75.50±0.45 (3)
crx	86.35±0.25 (1)	85.54±0.42 (4)	86.26±0.17 (3)	86.32±0.30 (2)
diabetes	75.05±0.68 (3)	75.00±0.44 (4)	75.36±0.71 (1)	75.26±0.68 (2)
ecoli	84.29±0.35 (2)	82.56±0.30 (3)	84.70±0.24 (1)	81.01±0.29 (4)
glass	77.29±1.09 (2)	75.98±1.05 (3.5)	77.38±1.44 (1)	75.98±1.27 (3.5)
heart-cleveland	80.92±1.44 (3)	80.40±1.87 (4)	81.19±0.86 (2)	81.52±1.52 (1)
hepatitis	81.42±1.89 (3)	77.81±1.90 (4)	81.68±2.26 (2)	82.19±1.33 (1)
ionosphere	93.33±0.39 (2)	92.54±0.71 (4)	93.50±0.38 (1)	93.22±0.33 (3)
pima	75.47±0.62 (3)	74.92±0.78 (4)	75.91±0.83 (1)	75.76±0.63 (2)
sonar	83.56±0.71 (2.5)	83.56±1.23 (2.5)	84.04±0.64 (1)	81.73±1.43 (4)
vehicle	75.67±0.70 (1)	75.08±0.66 (3)	75.53±0.80 (2)	72.70±0.57 (4)
vowel	94.34±0.26 (1.5)	88.18±0.55 (4)	94.34±0.18 (1.5)	91.86±0.19 (3)
average rank	2.14	3.43	1.75	2.68

to globally compare performance of four different classifiers on multiple data sets [7]. The null-hypothesis in this test is that all compared classifiers perform equally well. It uses ranks of each of classifiers on each of the data sets. The lower rank, the better classifier. We started from analyzing results of single MODLEM classifiers presented in Table 2. Friedman statistics for these results gives 59.59 which

Table 4. Comparison of using different classification strategies in Ivotes

Data set	Classification strategy			
	Discrimination measure		Rule support	
	abstain	no abstain	abstain	no abstain
breast-w	96.76±0.13 (1)	96.33±0.24 (2)	95.80±0.24 (3)	95.18±0.36 (4)
bupa	72.08±1.43 (3)	71.30±0.63 (4)	72.17±0.47 (2)	73.82±1.98 (1)
credit-german	75.37±0.09 (3)	75.87±0.21 (1)	75.67±0.25 (2)	75.23±0.76 (4)
crx	86.33±0.14 (2)	86.28±0.48 (3)	85.99±1.20 (4)	86.71±0.60 (1)
diabetes	75.74±0.93 (2)	75.17±0.32 (3.5)	75.17±0.12 (3.5)	76.13±0.37 (1)
ecoli	84.42±1.25 (2)	83.43±0.70 (3)	85.32±1.38 (1)	81.55±0.24 (4)
glass	74.92±0.79 (2.5)	74.92±1.88 (2.5)	75.08±1.23 (1)	73.52±2.10 (4)
heart-cleveland	82.95±0.56 (1.5)	82.95±0.82 (1.5)	81.63±1.12 (4)	81.96±2.02 (3)
hepatitis	84.09±0.80 (2)	75.27±1.69 (4)	84.95±0.80 (1)	82.15±0.80 (3)
ionosphere	93.16±0.23 (3)	92.78±0.59 (4)	93.73±0.23 (1)	93.45±0.23 (2)
pima	76.22±0.75 (1)	75.56±0.06 (4)	75.91±0.38 (2)	75.82±0.85 (3)
sonar	79.49±1.26 (2)	78.37±3.74 (3)	79.65±1.94 (1)	76.60±1.63 (4)
vehicle	74.23±0.60 (3)	74.55±0.20 (2)	75.14±0.22 (1)	73.68±0.15 (4)
vowel	91.78±0.50 (1)	86.13±0.62 (4)	91.58±0.98 (2)	91.18±0.59 (3)
average rank	2.07	2.96	2.04	2.93

exceeds the critical value 2.84 (for confidence level 0.05). We follow the same procedure with results of bagging presented in Table 3 and results of Ivotes presented in Table 4. In case of bagging, Friedman statistics gives 6.03. In case of Ivotes, Friedman statistics gives 2.47. Thus, we can reject the null hypothesis, at given confidence level 0.05, for single classifier and bagging. On the other hand, the value of Friedman statistic for Ivotes is close to critical value (p -value for this test is 0.07). We have not presented complete post-hoc analysis of differences between classifiers. However, we show the average ranks of each of classifiers in tables. The results of Friedman test and observed differences in average ranks between classifiers allow us to state that there is a significant difference between them.

We continue our comparison with examination of importance of difference in classification performance between each pair of classifiers. We apply Wilcoxon test [7] with null-hypothesis that the medians of results on all data sets of the two compared classifiers are equal. Let us remark, that in the paired tests ranks are assigned to the value of difference in accuracy between compared pair of classifiers. When we apply this test to results of single MODLEM classifiers, it detects statistically important difference in pairs between classifiers that abstain and those that does not (p -values for both classification strategies are around 0.0001). In case of bagging, Wilcoxon test indicates an important difference between classifiers that abstain regardless of classification strategy and this that use discrimination measure while not abstaining (p -values in this case are around 0.005). A difference is also reported between abstaining classifier that use rule support and the one that is not abstaining and use rule support (p -value 0.058). Moreover, there is a statistically important difference between abstaining classifier that use discrimination measure and abstaining classifier that use rule

support (p -value equal to 0.043). In case of examining Ivotes results, the situation is slightly different. In this case, statistically important differences are found only when abstaining classifiers that use discrimination measure are compared in pairs with not abstaining classifiers (p -value around 0.05).

5 Conclusions

Let us summarize results of experiments. First of all, we conclude that introducing abstaining of classifiers by excluding partial matching for rule sets has improved the total accuracy of the ensemble. However, the statistical analysis clearly shows that the range of this improvement depends on the type of ensemble and classification strategy.

First, we conclude that classification improvements are more significant for bagging than for Ivotes. This conclusion is further confirmed by values of average ranks. We can attribute this effect to the adaptive nature of Ivotes. In importance sampling consecutive classifiers should be more focused on learning objects misclassified by classifiers constructed in previous iterations. This may reduce the effect of abstaining. We suspect that similar behavior may be observed for other boosting approaches. On the other hand, in bagging, classifiers are constructed on independent samples. Moreover, each of the classifiers is constructed separately. Errors made by each of the component classifiers in learning phase do not affect the other classifiers. This makes the effects of abstaining more visible as it is not compensated during learning.

Although the aim of our experiment was not to compete with other rule ensembles, we also refer our best results against literature results of SLIPPER and other variants of bagging with rules [4,12], noticing comparable accuracy.

Analysing the results of using classification strategies, with discrimination measures or rule support, we claim that abstaining helped for both of them. The advantage depends on the particular ensemble. Generally speaking, Grzymala's strategy with rule support is a bit more effective, in particular for bagging. However, the other strategy also works surprisingly well in abstaining ensembles. Although its classification performance is worse than for using rule support, the difference of accuracies between variants with and without abstaining mechanism are larger than for the strategy with rule support. We can interpret it by specificity of evaluating discrimination measures. For unpruned sets of rules, which is a case in our experiment, the values of this measures are very similar among rules (most of them equal to 1). So the matching strategy is not powerful, as just counts nearly equally important rules. In the other strategy values of rule support are strongly diversified and may more contribute to class prediction. We also noticed that this difference much influences the performance of the single classifier (see Table 2) where the partial matching significantly improved the accuracy, however, using rule support is definitely more effective.

Finally, in our experiments we also recorded the average number of classifiers that refrain from predictions. We can conclude that for data sets, where abstaining has improved accuracy, the number of these abstaining classifiers is not

very high – on average usually between 2 and 4 classifiers (e.g., with respect to 20 components in bagging). This observation can lead us to a research question whether it is worth to further increase the level of abstaining. In our framework, it is possible to modify multiple matching part of classification strategy and produce unknown answer in case of uncertainty between two competitive class assignments. This could be a topic of future research.

References

1. Aijun, A.: Learning classification rules from data. *Computers and Mathematics with Applications* 45, 737–748 (2003)
2. Breiman, L.: Bagging predictors. *Machine Learning* 24(2), 123–140 (1996)
3. Breiman, L.: Pasting small votes for classification in large databases and on-line. *Machine Learning* 36, 85–103 (1999)
4. Cohen, W., Singer, Y.: A simple, fast and effective rule learner. In: *Proc. of the 16th National Conference on Artificial Intelligence AAAI 1999*, pp. 335–342 (1999)
5. Freund, Y., Schapire, R.E., Singer, Y., Warmuth, M.K.: Using and combining predictors that specialize. In: *Proceedings of the 29th ACM symposium on Theory of Computing*, pp. 334–343 (1997)
6. Grzymala-Busse, J.W.: Managing uncertainty in machine learning from examples. In: *Proc. 3rd Int. Symp. in Intelligent Systems*, pp. 70–84 (1994)
7. Kononenko, I., Kukar, M.: *Machine Learning and Data Mining*. Horwood Pub., England (2007)
8. Kuncheva, L.: *Combining Pattern Classifiers. Methods and Algorithms*. Wiley, Chichester (2004)
9. Mease, D., Wyner, A.: Evidence Contrary to the Statistical View of Boosting. *Journal of Machine Learning Research* 9, 131–156 (2008)
10. Pietraszek, T.: Optimizing abstaining classifiers using ROC analysis. In: *Proc. of the 22nd Int. Conf. on Machine Learning, ICML 2005*, pp. 665–672 (2005)
11. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco (1992)
12. Ruckert, U., Kramer, S.: Towards tight bounds for rule learning. In: *Proc. of the 21st Int. Conf. on Machine Learning, ICML 2004*, pp. 711–718 (2004)
13. Stefanowski, J.: The rough set based rule induction technique for classification problems. In: *Proc. of the 6th European Conf. on Intelligent Techniques and Soft Computing EUFIT 1998*, pp. 109–113 (1998)
14. Stefanowski, J.: On combined classifiers, rule induction and rough sets. In: Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J.W., Orłowska, E., Polkowski, L. (eds.) *Transactions on Rough Sets VI. LNCS*, vol. 4374, pp. 329–350. Springer, Heidelberg (2007)
15. Weiss, S.M., Indurkha, N.: Lightweight rule induction. In: *Proc. of the 17th Int. Conf. on Machine Learning, ICML 2000*, pp. 1135–1142 (2000)

Elicitation of Sugeno Integrals: A Version Space Learning Perspective

Henri Prade¹, Agnes Rico², and Mathieu Serrurier¹

¹ IRIT - Université Paul Sabatier

118 route de Narbonne 31062, Toulouse Cedex 9, France

² LIRIS - Université Claude Bernard Lyon 1

43 bld du 11 novembre, 69100 Villeurbanne, France

Abstract. Sugeno integrals can be viewed as multiple criteria aggregation functions which take into account a form of synergy between criteria. As such, Sugeno integrals constitute an important family of tools for modeling qualitative preferences defined on ordinal scales. The elicitation of Sugeno integrals starts from a set of data that associates a global evaluation assessment to situations described by multiple criteria values. A consistent set of data corresponds to a non-empty family of Sugeno integrals with which the data are compatible. This elicitation process presents some similarity with the revision process underlying the version space approach in concept learning, when new data are introduced. More precisely, the elicitation corresponds to a graded extension of version space learning, recently proposed in the framework of bipolar possibility theory. This paper establishes the relation between these two formal settings.

1 Introduction

Sugeno integrals [10,11,8] are an important family of combination functions for multiple criteria aggregation which are qualitative in the sense that they can be defined on bounded ordinal scales. They generalize weighted minimum and weighted maximum combinations, by allowing a form of synergy between criteria. A Sugeno integral, which is defined by a monotonically set-increasing allocation of weights to all subsets of criteria, returns a global evaluation for any n -tuple representing the values of n criteria (the empty set and the whole set of criteria are respectively associated with the lower and the upper bounds of the scale). This global evaluation always lies between the minimum and the maximum of the criteria values.

Conversely, a Sugeno integral (and more generally the family of all Sugeno integrals) that returns *known* global evaluations for collections of particular n -tuples, can be seen as a representation of this set of data. This raises the problem of the elicitation of Sugeno integrals compatible with a set of data that provides global satisfaction level assessments for situations described by tuples of criteria values. More generally, it may be natural to assume, in such a qualitative setting, that the global evaluations are only known to be lower and upper bounded by some values.

Each *new* piece of data constituted by a n -tuple of criteria values together with the associated global evaluation further constrains the family of Sugeno integrals that are compatible with a considered set of data. Provided that the set of data including the new piece remains compatible, the current family of Sugeno integrals that are compatible with the data is updated by the input of this new piece of data. Such a process presents some analogy with what happens in the version space approach to concept learning [5], where the version space is further restrained by the consideration of new pieces of data.

The paper explores this analogy in details. It turns out that the updating procedure in the case of Sugeno integrals exactly corresponds to a graded extension of the version space approach recently proposed in the setting of bipolar possibility theory [7]. This paper can be viewed as building a bridge between the previously cited reference and another paper by the same authors [6], also dealing with Sugeno integral elicitation. However, the present paper focuses on the ideal case where it exists a family of Sugeno integrals compatible with all the data, while [6] handles situations with inconsistent data.

The paper is organized in the following way. Section 2 restates the necessary background on Sugeno integrals. Section 3 studies the constraints induced on a Sugeno integral family by a set of compatible data. Section 4 provides a reminder on the version space approach viewed as a bipolar revision process. Section 5 shows how learning Sugeno integrals amounts to a continuous version space problem.

2 Sugeno Integrals: A Reminder

Let $C = \{C_1, \dots, C_n\}$ be a set of n evaluation criteria. A n -tuple of evaluations of some item on the basis of the n criteria is denoted $a = (a_1, \dots, a_n)$ where $a_i \in [0, 1] \quad \forall i \in \{1, \dots, n\}$. Thus, a n -tuple a is a function from C to the real interval $[0, 1]$.

Discrete Sugeno integrals are particular aggregation functions [9,10], which are defined through the specification of a fuzzy measure, or capacity v . This capacity is a mapping from 2^C to $[0, 1]$, such that:

- $v(\emptyset) = 0$;
- $v(C) = 1$;
- if $G \subset G' \subseteq C$ then $v(G) \leq v(G')$.

Given two capacities v_1 and v_2 such that $v_1 \leq v_2$ ¹, the set of the capacities v satisfying $v_1 \leq v \leq v_2$ is a lattice. More precisely, the considered set is a partially ordered set according to \leq in which any two elements have a supremum and an infimum.

A Sugeno integral of a function a from C to $[0, 1]$ with respect to a capacity v is defined by:

$$S_v(a) = \bigvee_{i=1}^n a_{\sigma(i)} \wedge v(C_{(i)}) \tag{1}$$

¹ i.e., $v_1(G) \leq v_2(G)$ for all subset G of C .

where σ is a permutation on $\{1, \dots, n\}$ such that $a_{\sigma(1)} \leq \dots \leq a_{\sigma(n)}$. \bigvee and \bigwedge denote *max* and *min* respectively. Moreover $C_{(i)} = \{C_{\sigma(i)}, \dots, C_{\sigma(n)}\}$.

As first pointed out in [4], $S_v(a)$ is the median for $2n - 1$ terms. Namely

$$S_v(a) = \text{median}(\{a_{\sigma(1)}, \dots, a_{\sigma(n)}\} \cup \{v(C_{(i)}), i = 2, \dots, n\})$$

A noticeable property of Sugeno integral is:

$$\bigwedge_{i=1}^n a_i \leq S_v(a_1, \dots, a_n) \leq \bigvee_{i=1}^n a_i. \tag{2}$$

A straightforward consequence is that $\forall c \in [0, 1], S_v(c, \dots, c) = c$ for any capacity v .

3 Sugeno Integrals Compatible with a Set of Data

The problem considered in this paper is the elicitation of a family of Sugeno integrals that are compatible with a set of data. Here, a set of data is a collection of tuples $a = (a_1, \dots, a_n)$ associated with a global rating α . It is assumed that $\forall i \ a_i \in [0, 1]$ and $\alpha \in [0, 1]$.

Definition 1. *A pair (a, α) is compatible with a Sugeno integral S_v if and only if $S_v(a) = \alpha$.*

In the following, we study the constraints induced by a pair (a, α) on the Sugeno integrals compatible with it and we fully characterize this family.

In this section for convenience, we assume that the a_i 's are already increasingly ordered i.e. $a_1 \leq \dots \leq a_n$. According to equation (2) and [8], there exists a Sugeno integral that satisfies $S_v(a) = \alpha$ if and only if $a_1 \leq \alpha \leq a_n$. In the following we assume that this condition holds for the pairs (a, α) considered. For discussing the equation $S_v(a) = \alpha$, it is useful to distinguish two cases.

Definition 2. *A data (a, α) is*

- a *DIF* type piece of data if $\forall i \in \{1, \dots, n\} \ a_i \neq \alpha$;
- a *EQU* type piece of data if $\exists i \in \{1, \dots, n\} \ a_i = \alpha$.

DIF Case: $\forall i \in \{1, \dots, n\} \ a_i \neq \alpha$.

First we study the DIF case. Let (a, α) be a DIF type piece of data and i be the index such that $a_1 \leq \dots \leq a_{i-1} < \alpha < a_i \leq \dots \leq a_n$. We can then define two particular capacities $\check{v}_{a,\alpha,DIF}$ and $\hat{v}_{a,\alpha,DIF}$:

Definition 3

$$\forall X \in 2^C, X \neq \emptyset, C \quad \check{v}_{a,\alpha,DIF}(X) = \begin{cases} \alpha & \text{if } \{C_i, \dots, C_n\} \subseteq X \\ 0 & \text{otherwise} \end{cases}$$

and

$$\forall X \in 2^C, X \neq \emptyset, C \quad \hat{v}_{a,\alpha,DIF}(X) = \begin{cases} \alpha & \text{if } X \subseteq \{C_i, \dots, C_n\} \\ 1 & \text{otherwise} \end{cases}.$$

It can be shown that:

$$\{v \text{ s.t. } S_v(a) = \alpha\} = \{v \text{ s.t. } \check{v}_{a,\alpha,DIF} \leq v \leq \hat{v}_{a,\alpha,DIF}\}.$$

Thus $\check{v}_{a,\alpha,DIF}$ and $\hat{v}_{a,\alpha,DIF}$ are the lower and upper bounds of the lattice of capacities which define the family of Sugeno integrals compatible with the pair (a, α) in the DIF case.

EQU Case: $\exists i \in \{1, \dots, n\} a_i = \alpha$.

We now study the EQU case. Let (a, α) be a EQU type piece of data and i and j be the indexes such that $a_1 \leq \dots \leq a_{j-1} < a_j = \dots = a_{i-1} = \alpha < a_i \leq \dots \leq a_n$. We can then define two particular capacities $\check{v}_{a,\alpha,EQU}$ and $\hat{v}_{a,\alpha,EQU}$:

Definition 4

$$\forall X \in 2^C, X \neq \emptyset, C \quad \check{v}_{a,\alpha,EQU}(X) = \begin{cases} \alpha & \text{if } \{C_j, \dots, C_{i-1}, \dots, C_n\} \subseteq X \\ 0 & \text{otherwise} \end{cases}$$

and

$$\forall X \in 2^C, X \neq \emptyset, C \quad \hat{v}_{a,\alpha,EQU}(X) = \begin{cases} \alpha & \text{if } X \subseteq \{C_i, \dots, C_n\} \\ 1 & \text{otherwise} \end{cases}.$$

It can be shown that:

$$\{v \text{ s.t. } S_v(a) = \alpha\} = \{v \text{ s.t. } \check{v}_{a,\alpha,EQU} \leq v \leq \hat{v}_{a,\alpha,EQU}\}.$$

Thus $\check{v}_{a,\alpha,EQU}$ and $\hat{v}_{a,\alpha,EQU}$ are the lower and the upper bounds of the lattice of capacities that define the family of Sugeno integrals compatible with the pair (a, α) in the EQU case.

A set of data are compatible if there exists a non empty family of Sugeno integrals that are compatible with each pair (a, α) in the data set. Otherwise, it means that there does not exist a representation of the data set by a unique family of integral and that several families are necessary, each covering a distinct subpart of the data set. Let us consider $\mathcal{D} = \{(a_i, \alpha_i)_{i \in \{1, \dots, P\}}\}$ a data set that contains P pairs. In order to simplify notations we note \check{v}_i the lower bound associated with (a_i, α_i) and \hat{v}_i the upper bound associated with (a_i, α_i) . These bounds are given by Definitions 3 and 4 for the DIF and the EQU case respectively. Then the lower and upper bound of the family of compatible Sugeno integrals, if it exists, are respectively

$$\check{v} = \bigvee_{i=1}^P \check{v}_i \text{ and } \hat{v} = \bigwedge_{i=1}^P \hat{v}_i.$$

Thus, when a new piece of information (a, α) is considered, \check{v} and \hat{v} are then revised by $\check{v}_{revised} = \check{v} \vee \check{v}_{a,\alpha}$ and $\hat{v}_{revised} = \hat{v} \wedge \hat{v}_{a,\alpha}$

4 Version Space Learning as a Bipolar Revision Process

Let \mathcal{X} denote a feature space used for describing examples. Examples in \mathcal{X} may be for instance vectors of values taken by attributes. In the bipolar possibilistic setting [2], two $[0, 1]$ -valued possibility distributions, δ and π over \mathcal{X} are considered: δ describes to what extent configurations are guaranteed to be possible and π describes what is not impossible. These two distributions are consistent in the bipolar framework iff $\forall x \in \mathcal{X}, \delta(x) \leq \pi(x)$. Given δ and π , when a new piece of information of the form π_{new} or δ_{new} is presented, a revision process takes place:

$$\forall x \in \mathcal{X}, \pi_{revised}(x) = \min(\pi(x), \pi_{new}(x)) \text{ and } \delta_{revised}(x) = \max(\delta_{new}(x), \delta(x)).$$

Note that the inequality $\delta_{revised} \leq \pi_{revised}$ still holds provided that the data are consistent.

Let us describe the version space learning framework [5]. Let $\mathcal{U} = \{0, 1\}$ be a so called concept space, where 1 means that the example of the concept is positive (here guaranteed to be possible i.e. $\delta(x) = 1$), and 0 means that the example is negative (here totally impossible i.e. $\pi(x) = 0$). An hypothesis h is a mapping from \mathcal{X} to \mathcal{U} . \mathcal{H} denotes the hypotheses space. The version space framework takes as input a set $\mathcal{S} = \{(x_i, u_i)_{i=1, \dots, m} \text{ s.t. } x_i \in \mathcal{X} \text{ and } u_i \in \mathcal{U}\}$ of m training examples, and a hypothesis space \mathcal{H} . The set \mathcal{H} is supposed to be equipped with a partial preorder \succeq expressing generality, formally: $h_1, h_2 \in \mathcal{H}, h_1 \succeq h_2$, iff $\{x \in \mathcal{X} | h_1(x) = 1\} \supseteq \{x \in \mathcal{X} | h_2(x) = 1\}$ (or equivalently $\{x \in \mathcal{X} | h_1(x) = 0\} \subseteq \{x \in \mathcal{X} | h_2(x) = 0\}$).

Definition 5. *An hypothesis h is sound w. r. t. a training example $(x, u) \in \mathcal{S}$ iff $h(x) = u$.*

If $\forall (x_i, u_i) \in \mathcal{S}, h$ is sound w.r.t (x_i, u_i) , then h is said to be sound w. r. t. \mathcal{S} . This framework defines the version space $\mathcal{V} = \{h \in \mathcal{H} | h \text{ is sound with } \mathcal{S}\}$. Let \mathcal{V}_{spe} and \mathcal{V}_{gen} be the sets of hypotheses that are respectively the lower and the upper bounds of \mathcal{V} with respect to the partial preorder \succeq .

We can view [7] an hypothesis as a possibility distribution μ_h , which states what feature configurations x are possible: $\forall h \in \mathcal{H}, \forall x \in \mathcal{X}, \mu_h(x) = h(x)$. Then, given a set of hypotheses, we define two possibility distributions: the most specific one

$$\forall H \in 2^{\mathcal{H}}, \forall x \in \mathcal{X}, \delta_H(x) = \min\{\mu_h(x); h \in H\}$$

and the most general one

$$\forall H \in 2^{\mathcal{H}}, \forall x \in \mathcal{X}, \pi_H(x) = \max\{\mu_h(x); h \in H\}.$$

Let $\delta_{\mathcal{S}}$ and $\pi_{\mathcal{S}}$ be the possibility distributions that correspond respectively to the most general and the most specific distributions revised, according to the bipolar revision process, by the examples in \mathcal{S} . The set of hypotheses in \mathcal{V}_{spe} contains the most specific hypotheses that cover all the positive examples and no negative examples. It means that these hypotheses must identify, if possible,

only the situations that are *guaranteed to be possible* w. r. t. the set of examples and then $\forall h \in \mathcal{V}_{spe}, \delta_{\mathcal{S}} \leq \mu_h$. In the same way, since \mathcal{V}_{gen} contains the most general hypotheses that cover all the positive examples and no negative ones, these hypotheses describe the situations that are *not impossible* w. r. t. the set of examples and then $\forall h \in \mathcal{V}_{gen}, \pi_{\mathcal{S}} \geq \mu_h$.

We now consider a generalized version space learning framework where the examples are associated with weights corresponding to possibility degrees in a linearly ordered scale, here $[0, 1]$ for simplicity. These possibility values have a different meaning according as the examples are positive or negative. If the example is positive, the weight corresponds to a guaranteed possibility degree. If the example is negative, the weight refers to the possibility distribution π that describes what is not impossible. The examples are then described by $\mathcal{S}_g = \{(x, u, \alpha)\}$ with $\alpha = \delta(x)$ if $u = 1$ and $\alpha = \pi(x)$ if $u = 0$. The hypothesis we want to learn describes a possibility distribution that associates a possibility degree to each example, which corresponds to the degree of compatibility of this example w. r. t. the target concept. The hypothesis is then a mapping from \mathcal{X} to $[0, 1]$. We now consider a graded hypothesis h_g that associates a possibility degree to each example (i.e. a mapping from \mathcal{X} to $[0, 1]$). The generality ordering on graded hypotheses corresponds to the partial preorder on possibility distributions, and thus $h_g^1 \leq h_g^2$ iff $\mu_{h_g^1} \leq \mu_{h_g^2}$.

A graded hypothesis is sound w. r. t. an example if it does not underestimate the possibility of a positive example and does not overestimate the possibility degree of a negative example.

Definition 6. h_g is sound w. r. t. \mathcal{S}_g iff $\forall (x, u, \alpha) \in \mathcal{S}_g, h_g(x) \geq \alpha$ if $u = 1$ and $h_g(x) \leq \alpha$ if $u = 0$.

A totally positive (guaranteed possible) example corresponds to $(x, 1, 1)$, and a totally negative (impossible) example to $(x, 0, 0)$. On the contrary, the examples of the form $(x, 1, 0)$ and $(x, 0, 1)$ have no influence on the learning algorithm according to the definition of soundness. Indeed, an example of the form $(x, 1, 0)$ means just that x is not guaranteed to be representative of the concept and $(x, 0, 1)$ means that it is completely possible (but not certain) for x to be representative of the concept.

Proposition 1. Let $\delta_{\mathcal{S}_g}$ and $\pi_{\mathcal{S}_g}$ denote the possibility distributions obtained by revising the initial bipolar distributions ($\forall x \in \mathcal{X}, \delta_{\mathcal{S}_g}(x) = 0, \forall x \in \mathcal{X}, \pi_{\mathcal{S}_g}(x) = 1$) by $\pi_{new}(x) = \alpha$ if $u = 0$ or $\delta_{new}(x) = \alpha$ if $u = 1$ for all the new pieces of information $(x, u, \alpha) \in \mathcal{S}_g$. Given $\mathcal{V}_g = \langle \mathcal{V}_{spe\ g}, \mathcal{V}_{gen\ g} \rangle$ the continuous version space from the set of examples \mathcal{S}_g , we have $\forall h \in \mathcal{V}_{spe\ g}, \delta_{\mathcal{S}_g} \leq \mu_h$ and $\forall h \in \mathcal{V}_{gen\ g}, \pi_{\mathcal{S}_g} \geq \mu_h$ where μ_h is the possibility distribution associated with h .

This proposition shows that we can use the version space learning settings for describing bounds of possibility distributions according to a set of examples of that are not impossible or guaranteed to be possible. Due to the limited description capabilities of the hypothesis language, the possibility distributions associated with the bounds of the version space may lie in between the distributions obtained by bipolar revision.

5 Learning Sugeno Integrals in a Continuous Version Space

At first glance, we can notice the similarity of the version space lattice with the one that describes a set of Sugeno integrals compatible with a data set \mathcal{D} . Starting from this remark, we will show that the construction of the lattice of Sugeno integrals compatible with a data set leads to a particular form of bipolar version space.

Let us encode the elicitation problem of Sugeno integrals as a bipolar version space problem.

- \mathcal{X} is the state space defined by the n -tuples of values taken by $C = \{C_1, \dots, C_n\}$
- \mathcal{S}_g is computed from \mathcal{D} as follows: A n -tuple $a = (a_1, \dots, a_n)$ associated with a global evaluation α is considered both as a positive and a negative example with a possibility level equal to α .
- The hypothesis space \mathcal{H} is the space of the Sugeno integrals defined on $C = \{C_1, \dots, C_n\}$.

Such an interpretation is compatible with the bipolar view of the version space. Indeed, a pair (a, α) is now viewed as two triples $(a, 0, \alpha)$ and $(a, 1, \alpha)$. Thus, from the evaluation context point of view, a is now both guaranteed to be possible and is not impossible at level α . The global rating α estimates how much the situation a is close to a reference situation that corresponds to the extreme value $\alpha = 1$. In this scope, due to Proposition 1, the example $(a, 1)$ both has a totally guaranteed possibility and is totally not impossible with respect to the reference situation. In the same way, $(a, 0)$ will be both an impossible example and an example that is not at all guaranteed to be possible. Then, we have the two following properties.

Proposition 2. *Let $\langle \check{v}, \hat{v} \rangle$ be the bounds of the lattice of Sugeno integrals compatible with a data set. Then, $\langle \check{v}, \hat{v} \rangle$ is the graded version space of the Sugeno integrals defined on $C = \{C_1, \dots, C_n\}$ where a n -tuple $a = (a_1, \dots, a_n)$ associated with a global evaluation α is considered both as a positive and a negative example with a possibility level equal to α .*

Proof: $\langle \check{v}, \hat{v} \rangle$ are the bounds of the lattice of all the Sugeno integrals compatible with the data set and their compatibility definition is a special case of the version space soundness definition where $h(x) = \alpha$ for $u = 0$ and $u = 1$ in the sense of section 4.

Proposition 3. *Let $\delta_{\mathcal{S}_g}$ and $\pi_{\mathcal{S}_g}$ denote the possibility distributions obtained by revising the initial bipolar distributions (i.e. $\forall x \in \mathcal{X}, \delta(x) = 0, \forall x \in \mathcal{X}, \pi(x) = 1$) by all the new pieces of information (a, α) present in the data set, each being understood as $\pi_{new}(a) = \alpha$ and $\delta_{new}(a) = \alpha$. Given $\langle \check{v}, \hat{v} \rangle$ the bounds of the lattice of Sugeno integrals consistent with a data set, we have $\delta_{\mathcal{S}_g} = \delta_{\check{v}}$ and $\pi_{\mathcal{S}_g} = \pi_{\hat{v}}$ for the examples in \mathcal{S}_g computed from \mathcal{D} .*

Thus in the particular case described at the beginning of the section (where each example appears twice with their respective positive and negative readings), the bounds of the lattice of Sugeno integrals compatible with a data set yield exactly the bipolar distributions obtained after the revision by each piece of data in the data set.

The constraints induced by a pair (a, α) may be felt too strong in practice, especially from a learning perspective. In order to relax these constraints we propose to consider a weakened definition of compatibility.

Definition 7. *A Sugeno integral is weakly compatible with a data pair (a, α) if its value has the same position as α with respect to the a_i 's in $a = (a_1, \dots, a_n)$.*

More formally, let us consider a pair (a, α) of the DIF type (the above definition only makes sense for this type). We define \check{a} and \hat{a} such that $\check{a} = \max\{a_i \in a_1, \dots, a_n \mid a_i \leq \alpha\}$ and $\hat{a} = \min\{a_i \in a_1, \dots, a_n \mid a_i \geq \alpha\}$. The bounds of the family of Sugeno integrals are now $\check{v}_{a, \check{a}, EQU}$ and $\hat{v}_{a, \hat{a}, EQU}$. This kind of compatibility constraint is more adapted to the subjective nature of the values as often in multiple criteria evaluation. We can now describe a new graded bipolar version space problem:

- \mathcal{X} is the state space defined by the n -tuples of values taken by $C = \{C_1, \dots, C_n\}$
- \mathcal{S}_g is computed from \mathcal{D} as follows: A n -tuple $a = (a_1, \dots, a_n)$ associated with a global evaluation α is considered both as being positive with a possibility level equal \check{a} and being negative with a possibility level equal to \hat{a} .
- The hypothesis space \mathcal{H} is the space of the Sugeno integrals defined on $C = \{C_1, \dots, C_n\}$.

We now obtain a version space problem where each example induces both a positive and a negative example with the constraint that the level associated with the positive example is lower than the level associated to the negative one. The two previous propositions can be then extended.

Proposition 4. *Let $\langle \check{v}, \hat{v} \rangle$ be the bounds of the lattice of Sugeno integrals that are weakly compatible with a data set. Then, $\langle \check{v}, \hat{v} \rangle$ is the version space of the Sugeno integrals defined on $C = \{C_1, \dots, C_n\}$ where a n -tuple $a = (a_1, \dots, a_n)$ associated with a global evaluation α is considered both as a positive example with a level equal to \check{a} and a negative example with a level equal to \hat{a} .*

Proposition 5. *Let $\delta_{\mathcal{S}_g}$ and $\pi_{\mathcal{S}_g}$ denote the possibility distributions obtained by revising the initial bipolar distributions (i.e. $\forall x \in \mathcal{X}, \delta(x) = 0, \forall x \in \mathcal{X}, \pi(x) = 1$) by all the new pieces of information (a, α) in the data set, each of them understood as $\pi_{new}(a) = \hat{a}$ and $\delta_{new}(a) = \check{a}$. Given $\langle \check{v}, \hat{v} \rangle$ the bounds of the lattice of capacity weakly compatible with a data set, we have $\delta_{\mathcal{S}_g}(x) = \delta_{\check{v}}(x)$ and $\pi_{\mathcal{S}_g} = \pi_{\hat{v}}$ for the examples in \mathcal{S}_g computed from \mathcal{D} .*

These results are interesting both from a Sugeno integral and a continuous version space point of view. The global rating can be viewed as the closeness measure with respect to a referential situation. This interpretation agrees with the way

the data are collected from human assessments. The link with the bipolar settings helps us to give an interpretation of the bounds of the results obtained by using a family of Sugeno integrals. Then the lower and the upper bounds correspond respectively to the guaranteed possibility and the non impossibility that the situation is close to the referential one. This also agrees with the idea that the global evaluation may be imprecise due to its subjective nature, which makes its precise assessment difficult.

The results also show that Sugeno integrals represent another form of hypotheses for graded version space learning, which differs from the stratified set of classical hypotheses used in [7]. This may be thought as not too surprising if we remember that Sugeno integrals can be represented as layered set of rules [3]. Moreover, the size of the bounds of the graded version space, using stratified sets of classical hypothesis as described in [7], may grow exponentially with the number of examples. In the case of a family of compatible Sugeno integrals both the lower and the upper bound are unique, which makes things much simpler, and thus computationally tractable. Lastly, the results suggest that the initial setting of the graded version space where an example may be associated only with a guaranteed possibility level, or only with a non impossibility level, is too general. Considering that each example is associated with the two levels, as for the graded version space view of the elicitation of Sugeno integrals, may be more realistic and tractable.

The link between the elicitation of Sugeno integrals and bipolar version space learning points out that the elicitation of Sugeno integrals from data is indeed a form of learning problem. In this scope our approach suffers from the impossibility of handling exceptions and noise, as it is already the case for the version space setting. One solution, explored in [6], is to consider the computation of the smallest partition of the data where each subset is compatible with a family of Sugeno integrals.

6 Conclusion

In this paper, we have provided a general approach to the elicitation of a family Sugeno integrals compatible with a data set when it exists. We have shown that the process of this elicitation is a form of graded version space learning. Moreover, we have proposed a reading of Sugeno integrals elicitation as a bipolar revision process. This views gives an interpretation, in terms of guaranteed possibility and non impossibility degrees, for the lower and upper bounds of the ratings when considering weak compatibility.

This opens some perspectives for the learning of Sugeno integrals. In the future, we will focus on the handling of noise and incompatibilities in the elicitation of Sugeno integrals. For doing that, we will need to introduce a graded compatibility measure for estimating to what extent a Sugeno integral is compatible with the data. This turns the problem of the identification of the family of Sugeno integrals compatible with the data into the finding of the most compatible family of Sugeno integrals with respect to the introduced measure.

References

1. Dubois, D., Marichal, J.-L., Prade, H., Roubens, M., Sabbadin, R.: The use of the discrete Sugeno integral in decision making: A survey. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 9, 539–561 (2001)
2. Dubois, D., Prade, H., Smets, P.: Not impossible vs. guaranteed possible in fusion and revision. In: *Proc. of the 6th European Conference (ESCQARU 2001)*, Toulouse, France, September 19-21, pp. 522–531. Springer, Heidelberg (2001)
3. Greco, S., Matarazzo, B., Slowinski, R.: Axiomatic characterization of a general utility function and its particular cases in terms of conjoint measurement and rough-set decision rules. *European Journal of Operational Research* 158(2), 271–292 (2004)
4. Kandel, A., Byatt, W.J.: Fuzzy sets, fuzzy algebra, and fuzzy statistics. *Proceedings of IEEE* 66, 1619–1639 (1978)
5. Mitchell, T.: Generalization as search. *Artificial intelligence* 18, 203–226 (1982)
6. Prade, H., Rico, A., Serrurier, M., Raufaste, E.: Eliciting sugeno integrals: Methodology and a case study. In: *Proc. of the 10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU 2009* (to appear, 2009)
7. Prade, H., Serrurier, M.: Bipolar version space learning. *International Journal of Intelligent Systems, Bipolar Representations of Information and Preference (Part 2: reasoning and learning)* 23(10), 1135–1152 (2008)
8. Rico, A., Labreuche, C., Grabisch, M., Chateauneuf, A.: Preference modeling on totally ordered sets by the Sugeno integral. *Discrete Applied Mathematics* 147 (2005)
9. Sugeno, M.: Theory of fuzzy integrals and its applications. PhD thesis, Tokyo Institute of technology (1974)
10. Sugeno, M.: Fuzzy measures and fuzzy integrals: A survey. In: Gupta, M.M., Saridis, G.N., Gaines, B.R. (eds.) *Fuzzy Automa and Decision Processes*, pp. 89–102. North-Holland, Amsterdam (1977)

Efficient MAP Inference for Statistical Relational Models through Hybrid Metaheuristics

Marenglen Biba, Stefano Ferilli, and Floriana Esposito

Department of Computer Science, University of Bari, Italy
biba@di.uniba.it, ferilli@di.uniba.it, esposito@di.uniba.it

Abstract. Statistical Relational Models are state-of-the-art representation formalisms at the intersection of logical and statistical machine learning. One of the most promising models is Markov Logic (ML) which combines Markov networks (MNs) and first-order logic by attaching weights to first-order formulas and using these as templates for features of MNs. MAP inference in ML is the task of finding the most likely state of a set of output variables given the state of the input variables and this problem is NP-hard. In this paper we present an algorithm for this inference task based on the Iterated Local Search (ILS) and Robust Tabu Search (RoTS) metaheuristics. The algorithm performs a biased sampling of the set of local optima by using RoTS as a local search procedure and repetitively jumping in the search space through a perturbation operator, focusing the search not on the full space of solutions but on a smaller subspace defined by the solutions that are locally optimal for the optimization engine. We show through extensive experiments in real-world domains that it improves over the state-of-the-art algorithm in terms of solution quality and inference time.

1 Introduction

Many real-world applications of AI require both probability and first-order logic (FOL) to deal with uncertainty and structural complexity. Traditionally, AI research has followed two separate directions: one that is based on logical representations, and one on statistical ones. Logical AI approaches like logic programming, description logics, classical planning, symbolic parsing, rule induction, etc, tend to emphasize handling complexity. Statistical AI approaches like Bayesian networks, hidden Markov models, neural networks, etc, tend to emphasize handling uncertainty. However, intelligent agents must be able to deal with both for applications in the real world. The first attempts to integrate logic and probability in AI date back to the works in [1,2,3]. Later, several authors began using logic programs to compactly specify Bayesian networks, an approach known as knowledge-based model construction [4].

Recently, in the field of Statistical Relational Learning [5], several approaches for combining logic and probability have been proposed. All these approaches combine probabilistic graphical models with subsets of FOL (e.g., Horn Clauses).

In this paper we focus on Markov Logic (ML) [6], a powerful representation language that has finite FOL and probabilistic graphical models as special cases. It extends FOL by attaching weights to formulas providing the full expressiveness of graphical models and FOL in finite domains and remaining well defined in many infinite domains [6]. Weighted formulas are viewed as templates for constructing MNs and in the infinite-weight limit, ML reduces to standard FOL. In ML it is avoided the assumption of i.i.d. (independent and identically distributed) data made by most of statistical learning approaches by using the power of FOL to compactly represent dependencies among objects and relations.

Inference is the process of responding to queries once the model has been learned. Efficient and effective inference is important to evaluate and compare the learned models. On the other side, inference is often a subroutine when learning statistical models of relational domains. These models often contain hundreds of thousands of variables or more, making efficient inference crucial to their learnability. Moreover, in on-line learning and inference, often used by agents, decisions are based on the output of the inference process, thus fast and accurate algorithms are strongly needed for this task. Maximum *a posteriori* (MAP) inference in MNs means finding the most likely state of a set of output variables given the state of the input variables and this problem is NP-hard. For ML, the MAP state is the state that maximizes the sum of the weights of the satisfied ground clauses. The authors in [7] use MaxWalkSAT [8] to find the MAP state. This paper presents a novel algorithm that exploits Iterated Local Search (ILS) [9] and Robust Tabu Search (RoTS) [10] metaheuristics. Experiments in real-world domains show that it outperforms the state-of-the-art algorithm for MAP inference in ML, in terms of solutions quality and inference running times.

The paper is organized as follows: Section 2 introduces MNs and MLNs, Section 3 introduces the ILS and RoTS metaheuristics and describes the IRoTS algorithm, Section 4 presents the experiments and Section 5 concludes.

2 Markov Networks and Markov Logic Networks

A MN is a model for the joint distribution of a set of variables $X = (X_1, \dots, X_n) \in \chi$ [11]. It is composed of an undirected graph G and a set of potential functions. The graph has a node for each variable, and the model has a potential function ϕ_k for each clique in the graph. A potential function is a non-negative real-valued function of the state of the corresponding clique. The joint distribution represented by a MN is given by: $P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}})$ where $x_{\{k\}}$ is the state of the k th clique. Z , known as the *partition function*, is given by: $Z = \sum_{x \in \chi} \prod_k \phi_k(x_{\{k\}})$. A Markov Logic Network [6] L is a set of pairs $(F_i; w_i)$, where F_i is a formula in FOL and w_i is a real number. Together with a finite set of constants $C = \{c_1, c_2, \dots, c_p\}$ it defines a MN $M_{L;C}$ as follows: 1. $M_{L;C}$ contains one binary node for each possible grounding of each predicate appearing in L . The value of the node is 1 if the ground predicate is true, and 0 otherwise. 2. $M_{L;C}$ contains one feature for each possible grounding of each formula F_i in L . The value of this feature is 1 if the ground formula is true, and 0 otherwise. The weight

of the feature is the w_i associated with F_i in L . Thus there is an edge between two nodes of $M_{L,C}$ iff the corresponding ground predicates appear together in at least one grounding of one formula in L . An MLN can be viewed as a template for constructing MNs. The probability distribution over possible worlds x specified by the ground MN $M_{L,C}$ is given by $P(X = x) = \frac{1}{Z} \exp(\sum_{i=1}^F w_i n_i(x))$ where F is the number of formulas in the MLN and $n_i(x)$ is the number of true groundings of F_i in x .

Propositionalization represents the task of replacing a first-order KB by a propositional one which is equivalent to it. In finite domains, this is done by replacing each universally (existentially) quantified formula with a conjunction (disjunction) of all its groundings. A first-order KB is satisfiable iff the equivalent propositional KB is satisfiable too. In this way, inference over a first-order KB (together with a set of constants) can be performed by propositionalization followed by test of satisfiability. In recent years, Stochastic Local Search [12] methods have made much progress in solving these hard combinatorial problems. In [13], finding the most likely state of a Bayesian network was solved by reduction to weighted satisfiability. For MLNs, MAP inference reduces to finding a truth assignment that maximizes the sum of weights of all satisfied clauses. The state-of-the-art algorithm for MAP inference in ML proposed in [7], is based on MaxWalkSat which extends WalkSAT [14] to the weighted satisfiability problem. In this case, each clause has a weight and the goal is maximizing the sum of the weights of satisfied clauses.

3 MAP Inference in MLNs through Metaheuristics

3.1 Iterated Local Search

Many widely known and high-performance local search algorithms make use of randomized choice in generating or selecting candidate solutions for a given combinatorial problem instance. These algorithms are called Stochastic Local Search (SLS) algorithms [12] and represent one of the most successful and widely used approaches for solving hard combinatorial problems. Many “simple” SLS methods come from other search methods by just randomizing the selection of the candidates during search, such as Randomized Iterative Improvement (RII), Uniformed Random Walk, etc. Many other SLS methods combine “simple” SLS methods to exploit the abilities of each of these during search. These are known as Hybrid SLS methods [12]. ILS is one of these metaheuristics because it can be easily combined with other SLS methods.

One of the simplest and most intuitive ideas for addressing the fundamental issue of escaping local optima is to use two types of SLS steps: one for reaching local optima as efficiently as possible, and the other for effectively escaping local optima. ILS methods [12,9] exploit this key idea, and essentially use two types of search steps alternately to perform a walk in the space of local optima w.r.t. the given evaluation function. The algorithm works as follows: search starts from a randomly selected element of the search space. From this initial candidate, a

locally optimal solution is obtained by applying a subsidiary local search procedure. Then each iteration consists of three steps: first a perturbation method is applied to the current candidate s ; this yields a modified candidate s' from which in the next step a subsidiary local search is performed until a local optimum s'' is obtained. In the last step, an acceptance criterion is used to decide from which of the two local optima s or s'' the search process is continued. The algorithm can terminate after some steps have not produced improvement or simply after a certain number of steps.

3.2 Robust Tabu Search

Robust Tabu Search (RoTS) [10] is a special case of Tabu Search [15]. In each search step, the RoTS algorithm (Algorithm 1) for MAX-SAT flips a non-tabu variable that achieves a maximal improvement in the total weight of the unsatisfied clauses and declares it tabu for the next tt steps. The parameter tt is called the tabu tenure. An exception to this “tabu” rule is made if a more recently flipped variable achieves an improvement over the best solution seen so far (this mechanism is called aspiration). Furthermore, whenever a variable has not been flipped within a certain number of search steps, it is forced to be flipped. This implements a form of long-term memory and helps prevent stagnation of the search process. The tabu status of variables is determined by comparing the number of search steps that have been performed since the most recent flip of a given variable with the current tabu tenure. Finally, instead of using a fixed tabu tenure, every n iterations the parameter tt is randomly chosen from an interval $[tt_{min}, tt_{max}]$ according to a uniform distribution. The RoTS algorithm is closely related to MaxWalkSAT-Tabu where in each search step one of the non-tabu variables that achieves a maximal improvement in the total weight of the unsatisfied clauses is flipped and declared tabu for the next tt steps. However, different from MaxWalkSAT, RoTS additionally to the aspiration criteria, forces a variable to be flipped if it has not been flipped for a certain number of steps.

3.3 The IRoTS Algorithm

The original version of IRoTS for MAX-SAT was proposed in [16]. Algorithm 2 starts by randomly initializing the truth values of all atoms. Then it tries to efficiently reach a local optimum CL_S by using RoTS. At this point, a perturbation method based again on RoTS is applied leading to the neighbor CL'_C of CL_S and then again a local search based on RoTS is applied to CL'_C to reach another local optimum CL'_S . The *accept* function decides whether the search must continue from the previous local optimum or from the last found local optimum CL'_S .

Careful choice of the various components of Algorithm 2 is important to achieve high performance. For the tabu tenure we refer to the parameters used in [16] that have proven to be highly performant across many domains. At the

Algorithm 1. The Robust Tabu Search algorithm

```

RoTS(F: weighted CNF formula,  $tt_{min}$ : minimum tabu tenure,  $tt_{max}$ : maximum tabu tenure,
maxNoImprov: maximum number of steps without improvement)
num_atoms = number of variables in F;
 $\hat{A}$  = randomly chosen assignment of the variables in F;
Cost( $\hat{A}$ ) = sum of weights of unsatisfied formulas; A =  $\hat{A}$ ; k = 0;
repeat
  if k mod n = 0 then
    tt = random( $[tt_{min}, tt_{max}]$ );
  end if
  Atom = randomly selected variable whose flip results in a maximal improvement in Cost;
  if Score(A with Atom flipped) < Score(A) then
    A = A with Atom flipped;
  else
    if  $\exists$  variable A that has not been flipped for  $\geq 10 * n$  steps then
      A = A with Atom flipped;
    else
      Atom = randomly selected non-tabu variable whose flip results in the maximal improvement
      in Cost;
      A = A with Atom flipped;
    end if
  end if
  if Score(A) < Score( $\hat{A}$ ) then
     $\hat{A}$  = A;
  end if
  k = k + 1;
until no improvement in  $\hat{A}$  for maxNoImprov steps
return  $\hat{A}$ 

```

beginning of each iteration, all variables are declared non-tabu. The clause perturbation operator (flipping the atoms truth value) has the goal to jump in a different region of the search space where search should start with the next iteration. It is not easy to decide whether strong or weak perturbations should be performed. Weak perturbations may cause the subsidiary local search procedure *LocalSearch_{RoTS}* to fall again in the same local optimum, but very strong perturbations may lead *LocalSearch_{RoTS}* to a very long search, far away from the next good solution. As perturbation heuristics in our algorithm, we decided to use a fixed number of RoTS steps $9n/10$ with tabu tenure $n/2$ where n is the number of atoms (in future work we intend to dynamically adapt the nature of the perturbation). Regarding the procedure *LocalSearch_{RoTS}*, it performs RoTS steps until no improvement is achieved for n^2/d steps (we call d threshold ratio) with a tabu tenure $n/10 + 4$. The *accept* function always accepts the best solution found so far. The difference of our algorithm with that in [16] is that we do not dynamically adapt the tabu tenure and do not use a probabilistic choice in *accept*.

4 Experiments

Regarding IRoTS we want to answer these questions:

(Q1) Does IRoTS improve over the state-of-the-art algorithm for MLNs in terms of solutions quality?

(Q2) Does IRoTS performance depend on the particular configuration of clauses' weights?

Algorithm 2. The Iterated Robust Tabu Search algorithm

```

Input:  $C$ : set of weighted clauses in CNF, BestScore: current best score)
 $CL_C$  = Random initialization of truth values for atoms in  $C$ ;
 $CL_S$  = LocalSearchRoTS( $CL_C$ );
BestAssignment =  $CL_S$ ; BestScore = Score( $CL_S$ );
repeat
   $CL'_C$  = PerturbRoTS(BestAssignment);
   $CL'_S$  = LocalSearchRoTS( $CL'_C$ );
  if Score( $CL'_S$ )  $\geq$  BestScore then
    BestScore = Score( $CL'_S$ )
  end if
  BestAssignment = accept(BestAssignment,  $CL'_S$ );
until  $k$  consecutive steps have not produced improvement
Return BestAssignment

```

(Q3) Does IRoTS performance depend on particular features of the dataset, i.e., number of ground clauses and predicates?

(Q4) In case IRoTS finds better solutions than the state-of-the-art algorithm, what is the performance in terms of running time?

(Q5) What is the performance of IRoTS for huge relational domains with hundreds of thousands of ground predicates and clauses?

To perform MAP inference we need MLN models and evidence data. MLNs can be hand-coded or learned from data. Since the goal here is to perform inference for complex models where it is hard to find the MAP state given evidence, we decided to generate complex models from real-world data and test IRoTS against the current state-of-the-art algorithm of [7]. We used the UW-CSE dataset [6] and the MLN hand-coded model that comes together with it. For the first experiment we learned weights using the algorithm PSCG [17] giving advisedBy as non-evidence predicate. We trained the algorithm for 500 iterations on each area of the dataset. After learning the MLNs, we performed MAP inference with query predicate advisedBy. We commented on the test set also the student and professor predicates together with predicate advisedBy.

In order to equally compare IRoTS with MaxWalkSAT, we decided to compare our algorithm IRoTS with the tabu version of MaxWalkSAT by using the same number of search steps for both algorithms. For IRoTS, parameters $d = 1$ and $k = 3$. We observed that on the language and theory folds the iterations were very fast and 3 steps without improvement were too few. Thus we used $k = 10$ only for these areas, and $k = 3$ for the rest. Anyway, for IRoTS we counted the overall number of flips and we used the same number for MaxWalkSAT with tabu (MWS-T). The tabu tenure for MWS-T was set to the default of Alchemy [18], i.e., equal to 5. The results are reported in Table 1 where for each algorithm we report the cost of false clauses of the final solution and running times in minutes.

Since IRoTS uses the perturbation procedure to escape local optima, it would be fair to compare IRoTS with a version of MWS-T that uses a similar mechanism to jump in a different region of the search space. For this reason we compared IRoTS also with MWS-TR (Tabu&Restarts) with a number of ten restarts and with a number of flips for each iteration equal to 1/10th of the overall number of flips. In this way, the equality of the comparison is maintained

Table 1. Inference results for predicate advisedBy with MLNs learned with 500 iterations of PSCG

fold	IROTS		MWS-T		MWS-TR		preds	clauses
	Cost	Time	Cost	Time	Cost	Time		
ai	93103.7	56.65	92393.5	62.98	92512.9	60.27	4760	185849
gra	72221.8	26.91	72245.1	28.88	71659.8	46.30	3843	136392
lang	32398.1	1.03	32668.2	1.08	32380.1	1.06	840	15762
sys	117144.0	125.71	118416.0	134.35	118629.0	192.16	5328	218918
theo	71726.1	17.85	71727.9	19.30	71873.1	34.03	2499	73600
avg	77318.7	45.63	77490.1	49.32	77411.0	66.76	-	-

Table 2. Inference results for predicate advisedBy with MLNs learned by running PSCG for 10 hours

fold	IROTS		MWS-T		MWS-TR		preds	clauses
	Cost	Time	Cost	Time	Cost	Time		
ai	98513.5	71.34	99737.8	78.31	99876.4	74.53	4760	185849
grap	28007.9	26.3	28074.8	28.02	28005.2	43.06	3843	136392
lang	10985.8	1.48	11070.8	1.57	10711.3	1.52	840	15762
sys	73154.6	55.06	73471.8	59.92	73642.9	57.98	5328	218918
theo	90979.1	25.53	89517.9	26.73	89462.7	49.06	2499	73600
Avg	60328.2	35.94	60374.6	38.91	60339.7	45.23	-	-

in order to perform the same number of flips for all algorithms. Moreover, for MWS-T we used the default tabu tenure of Alchemy that is five, but it would be more interesting to compare IROTS with MWS-TR with the same tabu tenure as IROTS, i.e., $n/10 + 4$. Thus we used this tabu tenure for MWS-TR.

As can be seen, IROTS on average finds solutions of higher quality than the other algorithms and is also faster even though the number of search steps is the same. Thus, questions (Q1) and (Q4) can be answered affirmatively. However, we want to be sure that the performance of IROTS towards the other algorithms does not depend on the particular weights of the model. For this reason we decided to generate other MLNs on the same dataset but with different weights from the first ones. We did this by using again PSCG and running it for 10 hours instead of 500 iterations for each training set. This will guarantee that the MLNs generated will be different in terms of the clauses' weights. Inference results are reported in Table 2 and again IROTS performs better than the other algorithms, thus question (Q2) can be answered affirmatively, since for the same number of ground clauses and predicates but with different clauses' weights, IROTS finds better solutions.

An important question is whether the performance of IROTS towards the other algorithms depends on the number of ground clauses and predicates. It is important to maintain the same performance for any number of groundings. Thus we decided to consider an additional query predicate in order to change the number of groundings. We learned weights using PSCG (with 50 iterations per fold) but this time considering as non-evidence predicates both the predicate advisedBy and tempAdvisedBy. The final MLNs learned should be able to predict the probability for all the groundings of both predicates. We report experiments for each predicate in turn and finally for an inference task where both predicates are specified as query predicates. In this way we will have a different

Table 3. Inference results for query predicate advisedBy on a model learned with both advisedBy and tempAdvisedBy as non-evidence predicates

fold	IROTS		MWS-T		MWS-TR		preds	clauses
	Cost	Time	Cost	Time	Cost	Time		
ai	50.85	2.15	50.85	13.55	50.85	10.30	4760	185762
gra	62.10	2.10	62.10	8.40	62.10	6.30	3843	136297
lang	9.75	0.12	9.75	0.32	9.75	0.27	840	15711
sys	52.96	11.20	52.96	80.71	52.96	60.07	5328	218820
theo	57.23	0.15	57.23	2.08	57.23	1.63	2499	73540
avg	46.58	3.14	46.58	21.01	46.58	15.71	-	-

Table 4. Inference results for query predicate tempAdvisedBy using a model learned with both advisedBy and tempAdvisedBy as non-evidence predicates

fold	IROTS		MWS-T		MWS-TR		preds	clauses
	Cost	Time	Cost	Time	Cost	Time		
ai	16112.70	93.89	16669.30	101.98	16309.00	143.29	4760	185672
gra	12872.90	30.63	13153.10	32.81	12765.50	33.03	3843	136244
lang	2238.57	0.73	2196.23	0.75	2024.31	0.68	840	15706
sys	19722.50	61.07	20352.70	65.65	19938.30	95.62	5328	218727
theo	7388.90	47.85	7668.17	55.60	7600.41	34.50	4892	261078
avg	11667.11	46.83	12007.90	51.36	11727.50	61.42	-	-

number of ground predicates and clauses compared to the previous experiments. As it can be seen (Table 3), in this case the algorithms find the same solution but IROTS is much faster than the other algorithms. While for the predicate tempAdvisedBy (Table 4), IROTS performs much better than MWS-T and is more accurate than MWS-TR. Regarding running times IROTS is clearly faster than the other algorithms.

Finally, with the last generated MLNs by declaring as non-evidence predicates both advisedBy and tempAdvisedBy, we performed joint inference with both predicates in a single step. IROTS is clearly superior against the other algorithms (Table 5). The difference in solutions quality is higher towards MWS-T with an improvement of approximately 12% in the solutions quality. MWS-TR is competitive with IROTS but loses on average 7% in terms of solutions quality towards IROTS. Regarding running time IROTS is slightly slower than the MWS-T and slightly faster than MWS-TR. However, the differences are not significant compared to the overall running times.

The results clearly answer questions (Q2) and (Q3). We generated different MLNs with different weights, but the better performance of IROTS towards the other algorithms was not affected by clauses' weights. Moreover, with the last three experiments we generated different MLNs that together with the evidence data produce a different number of ground predicates and clauses during inference. The results show that IROTS is superior in terms of solutions quality and performance does not change with the number of ground predicates and clauses. Regarding question (Q4), IROTS is in general faster. In only one case, IROTS is slightly slower than MWS-T but finds much better solutions. Thus question (Q4) can be answered stating that even though it finds better solutions, IROTS does not spend more time and in general is faster than both MWS-T and MWS-TR.

Table 5. Inference results for query predicates advisedBy and tempAdvisedBy in a single inference task using a model learned with both advisedBy and tempAdvisedBy as non-evidence predicates

fold	IROTS		MWS-T		MWS-TR		preds	clauses
	Cost	Time	Cost	Time	Cost	Time		
ai	13367.10	294.01	15659.60	278.3	14610.00	331.79	9384	680351
grap	11511.40	166.78	12622.90	160.78	11994.40	150.17	7564	495227
lang	1996.73	7.1	2054.05	8.71	2004.55	8.1	1624	52491
sys	16823.70	362.78	19283.50	339.2	18484.60	337.44	10512	804425
theo	6845.18	80.65	7545.73	104.3	7112.81	103.05	4900	261890
avg	10108.82	182.26	11433.16	178.26	10841.27	186.11	-	-

Table 6. Inference results for query predicates taughtBy, advisedBy and tempAdvisedBy in a single inference task

fold	IROTS		MWS-T		MWS-TR		preds	clauses
	Cost	Time	Cost	Time	Cost	Time		
ai	509738	52.97	536670	49.22	538508	50.52	23664	1894428
grap	338790	40.33	338554	35.58	339004	32.62	23485	1510794
lang	37034	1.18	43756.2	1.17	42886.7	1.14	5152	157944
sys	494128	48.58	619380	38.58	604742	36.62	26136	2045461
theo	175668	15.33	214252	11.5	210758	12.62	14504	794484
avg	311071.6	31.68	350522.44	27.21	347179.74	26.7	-	-

The results in Table 5 answer question (Q5): inference consists in thousands of ground predicates and a really huge number of clauses. However, to fully answer question (Q5), we generated MLNs with an additional query predicate such that the number of ground predicates and clauses could be very high. We chose from the predicates of UW-CSE the taughtBy predicate that has three arguments, thus a huge ground MN is to be solved for MAP inference. We learned the MLNs with taughtBy as an additional non-evidence predicate and running PSCG for 50 iterations. We then performed inference with all query predicates. IRoTS again performs better than the other algorithms (Table 6). The number of ground clauses is very high and in one fold it reaches nearly 2 million. This is common for relational domains where grounding of FOL clauses causes a combinatorial explosion in the number of ground clauses. Results however show that IRoTS is slower, even though it finds much better solutions. Thus question (Q5) can be answered affirmatively in that for tasks with a huge number of groundings, IRoTS is superior in terms of solutions quality.

5 Conclusion

Statistical Relational Models are state-of-the-art formalisms at the intersection of logical and statistical machine learning. Markov Logic combines Markov Networks and first-order logic by attaching weights to first-order formulas and using these as templates for features of MNs. In this paper we present a high performing algorithm for MAP inference in Markov Logic, based on the ILS and RoTS metaheuristics. The algorithm performs a biased sampling of the set of local

optima focusing the search not on the full space of solutions but on a smaller subspace defined by the solutions that are locally optimal for the optimization engine. Extensive experiments show that it improves over the state-of-the-art algorithm in terms of solution quality and inference time.

References

1. Bacchus, F.: Representing and Reasoning with Probabilistic Knowledge. MIT Press, Cambridge (1990)
2. Halpern, J.: An analysis of first-order logics of probability. *Artificial Intelligence* 46, 311–350 (1990)
3. Nilsson, N.: Probabilistic logic. *Artificial Intelligence* 28, 71–87 (1986)
4. Wellman, M., Breese, J.S., Goldman, R.P.: From knowledge bases to decision models. *Knowledge Engineering Review* 7 (1992)
5. Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning. MIT, Cambridge (2007)
6. Richardson, M., Domingos, P.: Markov logic networks. *Machine Learning* 62, 107–236 (2006)
7. Singla, P., Domingos, P.: Discriminative training of markov logic networks. In: Proc. 20th Nat'l Conf. on AI (AAAI), pp. 868–873. AAAI Press, Menlo Park (2005)
8. Selman, B., Kautz, H., Cohen, B.: Local search strategies for satisfiability testing. In: *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, pp. 521–532. American Mathematical Society, Providence (1996)
9. Loureno, H., Martin, O., Stutzle, T.: Iterated local search. In: Glover, F., Kochenberger, G. (eds.) *Handbook of Metaheuristics*, pp. 321–353. Kluwer, USA (2002)
10. Taillard, E.: Robust taboo search for the quadratic assignment problem. *Parallel Computing* 17, 443–455 (1991)
11. Della Pietra, S., Pietra, V.D., Laferty, J.: Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 380–392 (1997)
12. Hoos, H.H., Stutzle, T.: *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann, San Francisco (2005)
13. Park, J.D.: Using weighted max-sat engines to solve mpe. In: Proc. of AAAI, pp. 682–687 (2005)
14. Kautz, H., Selman, B., Jiang, Y.: A general stochastic approach to solving problems with hard and soft constraints. In: *The Satisfiability Problem: Theory and Applications*. AMS (1997)
15. Glover, F., Laguna, M.: *Tabu Search*. Kluwer, Boston (1997)
16. Smyth, K., Hoos, H., Stützle, T.: Iterated robust taboo search for max-sat. In: *Canadian Conference on AI*, pp. 129–144 (2003)
17. Lowd, D., Domingos, P.: Efficient weight learning for markov logic networks. In: Proc. of the 11th PKDD, pp. 200–211. Springer, Heidelberg (2007)
18. Kok, S., Singla, P., Richardson, M., Domingos, P.: The alchemy system for statistical relational ai. Technical report, Dep. CSE-UW, Seattle, WA (2005)

Combining Time and Space Similarity for Small Size Learning under Concept Drift

Indrė Žliobaitė

Vilnius University, Faculty of Mathematics and Informatics,
Naugarduko st. 24, LT-03225 Vilnius, Lithuania
`indre.zliobaite@mif.vu.lt`

Abstract. We present concept drift responsive method for classifier training for sequential data. Relevant instance selection for training is based on similarity to the target observation. Similarity in space and in time is combined. The algorithm determines an optimal training set size. It can be used plugging in different base classifiers. The proposed algorithm shows the best accuracy in the peer group. The algorithm complexity is reasonable for the field applications.

1 Introduction

Classification of sequential data is often a non stationary problem. Changes in underlying distribution are referred to as *concept drift*. Concept drift occurs when one data source S_1 (underlying probability distribution) gets replaced by S_2 . For example, in spam categorization new ways of spam are invented, personal interpretation, what is spam, might change.

Concept drift is assumed to be unpredictable with certainty, but can be expected depending on the data domain. An optimal classifier *after* the drift would model S_2 . The data from the source S_2 is scarce after the drift.

If concept drift is observed, heuristics suggests dropping the old data (originating from S_1) out of the training sample, leaving a *training window* of the N most recent sequential instances. Another approach is to relearn selecting only relevant instances into the training set. We introduce **a method which builds the classifier using the nearest neighbors (in time and space) of the observation in question**. The method is expected to demonstrate a competitive advantage under gradual non uniform drift scenarios in small and moderate size data sequences.

In Section 2 we outline related work and map the proposed method. In Section 3 the proposed method is presented in detail. Section 4 gives experimental setup and the results. Sections 5 and 6 conclude.

2 Related Work

Concept drift is assumed being not predictable with confidence, classification methods need to have adaptation mechanisms. One group of concept drift responsive methods is based on change detection [22,6,16,49]. After the drift is

detected, a portion of the old data is dropped out of the training sample. These methods are usually based on sudden drift scenario.

Another group of methods build multiple classifiers and then try to replicate the distribution of concepts via fusing or selecting the classifier outputs. This group includes heuristic window search algorithms [11,17] as well as classifier ensembles [18,21,12,19]. The training data is formed including instances, which are sequential in time. Unselective use of the old data might lead to gain in accuracy by chance, depending on consistency between the old and the new concept. When concept drift is gradual several data sources might be active at a time interval, systematic training data selection is needed.

Systematic data selection issue has been brought up in [7,5,15,19,3]. Ganti et al. [7] describe generic state of the art framework for systematic training data selection without real plug-and-play algorithm. Lazarescu et al. in [15] extend their previous algorithm with relevance (similarity to current concept) based forgetting. Tsymbal et al. [19] test the ensemble classifiers on the nearest neighbors of the target observation. They select the classifier, which will make the final decision, based on this test. They do not use instance selection for *building* the classifiers. Fan in [5] proposes decision tree specific plug-and-play method. They build multiple classifiers and include systematic training set selection. [3] organize training data into prototype clusters, referred to as case bases. In contrast to the above works, they exclude the instances which are too similar to the ones already present in the training set.

In this study we propose an algorithm FISH, which systematically selects training instances. Similarity in time and space is addressed. The method can be used with different base classifiers. In FISH method the size of a training set is prefixed and set in advance, while the extension FISH2 operates using variable training set size, which is reset in each step of the training process.

FISH *builds* a classifier based on similarity to target instance. Other multiple classifier methods [19] employ similarity aspect but only in the classifier selection phase. However, the needed classifier might not be present among the ensemble members. Our approach is similar to lazy learning [2], but the main difference is that the latter does not construct explicit generalization and makes classification based on direct comparison of the target and training instances.

3 The FISH Methods

3.1 Scenario Set Up

Consider a streaming data for classification. One data point $\mathbf{x} \in \mathbb{R}^p$ is received at a time. At time $t + 1$ the task is to predict the class label for the target observation \mathbf{x}_{t+1} . Any selected or all the historical labeled data $\mathbf{x}_1, \dots, \mathbf{x}_t$ can be used to build a classifier. At time $t + 2$ after the classification decision and receiving the true label, we could add \mathbf{x}_{t+1} to the training data. At time $t + 1$ one data source is active, but the data instance can come from several sources with a probability. Assume the prior probabilities of the classes are equal.

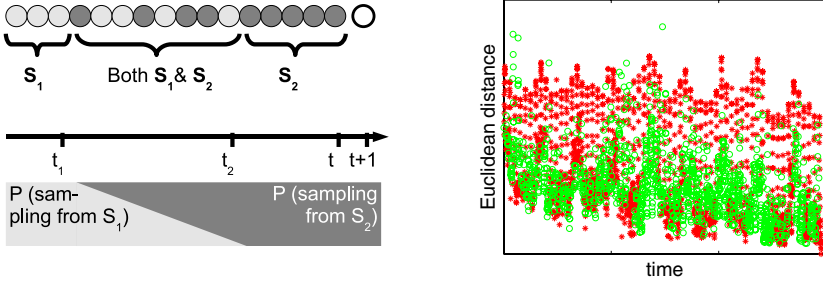


Fig. 1. (a) Gradual drift scenario. (b) Electricity data example, \circ and $*$ mark classes.

Concept drift means that i.i.d. may not hold for the data $\mathbf{x}_1, \dots, \mathbf{x}_t$. For example consider a gradual drift scenario, illustrated in Figure 1. Up to time t_1 data generating source S_1 is active. In time interval $t_1 + 1, \dots, t_2$ both sources are active and an instance comes from either one or the other source with a probability. The probability of sampling from S_2 increases with time. A designer does not know when the sources switch. We aim to select a training set, consisting of the instances, which are similar to the target observation. We can find, how similar \mathbf{x}_{t+1} is to the training instances available, even though the label of \mathbf{x}_{t+1} is not known.

3.2 Similarity in Time and Space

Similarity between two instances is usually defined as a function of distance in space. If the problem is non stationary, similarity in time is relevant as well.

Consider a rotating hyperplane example in Figure 2. A binary classification problem is represented by black and gray dots. There are three data generating sources S_1, S_2 and S_3 . In (a) the instances which coming from the source S_1 are depicted. Later in (b) the source S_2 becomes active, the discriminant line for these new instances has rotated 45° . Finally, in (c) the source S_3 is active, corresponding to another rotation of 45° . A circle in each of the subfigures defines neighborhood (similarity) in space. In (a) only instances from class 2 are within the circle, in (b) there is a mix of both classes, while in (c) only class 1 instances are within the circle. The circle is fixed in space, but the nearest neighbors within the circle would depend on which data source was active at that time.

Let D_{ij} be a similarity measure between two instances \mathbf{x}_i and \mathbf{x}_j . We combine similarities in space and time as a sum of the distances:

$$D_{ij} = a_1 d_{ij}^{(s)} + a_2 d_{ij}^{(t)}, \tag{1}$$

where $d^{(s)}$ indicates distance in space, $d^{(t)}$ indicates distance in time, a_1, a_2 are the weight coefficients, for simplicity $A = \frac{a_2}{a_1}$ can be used, then $D_{ij}^* = d_{ij}^{(s)} + A d_{ij}^{(t)}$.

In order to manage the balance between the time and space distances, $d^{(s)}$ and $d^{(t)}$ need to be normalized. We suggest scaling the values of each feature

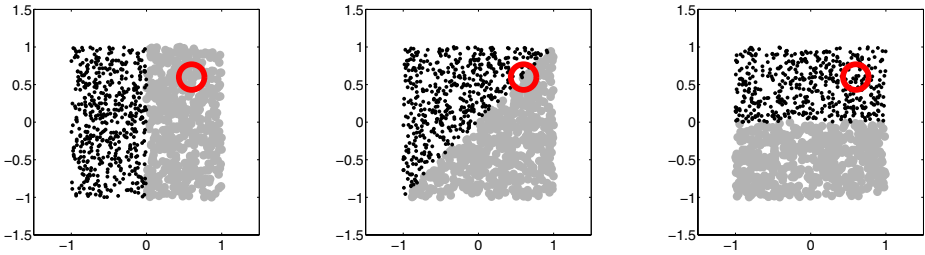


Fig. 2. Rotating hyperplane example: (a) initial source S_1 , (b) source S_2 after 45° rotation, (c) source S_3 after 90° rotation. Black and grey dots represent the two classes.

to form an interval $[0, 1]$. Similarity in time between the instances \mathbf{x}_i and \mathbf{x}_t is defined here as $d_{ij}^{(t)} = |i - j|$. We scale the time distances so that $d_{ij}^{(t)} \in [0, 1]$.

We justify this simple setting by the following property. If only similarity in space is addressed in Equation (11) $a_2 = 0$, the measure turns to *instance selection*. If only similarity in time is addressed $a_1 = 0$, we end up with the commonly used *training window* approach.

3.3 The FISH Algorithm

FISH selects training instances based on similarity to the target observation. We present two versions of the algorithm, FISH requires training sample size as an input, while an extension FISH2 allows variable training sample size. To implement a variable sample size we incorporate the principles from two windowing methods (11) (KLI) and (19) (TSY).

In the presentation we focus on more advanced FISH2, more details on FISH can be found in the technical report (20). FISH2 is presented in Figure 3.

First we calculate the similarities in time and space between the target observation and the historical instances (Equation (11)) and sort them from minimum to maximum. We use the Euclidean distance: $d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^p |x_i - y_i|^2}$ as a basic measure. The effect of different similarity measures is a subject of further investigation. The features are scaled to the interval $[0, 1]$.

Next, the closest k instances to the target observation are selected as a validation set. The training sets are formed using the closest s instances. The method works similarly to windowing in (11) (KLI). They use sequential instances in time to form the windows. We use combined time and space similarity. We select the training size L , which has given the best accuracy on the validation set.

Cross validation needs to be employed when testing on the validation set. That means we take validation instances for testing one by one. At testing time the one is excluded from the training set. Without cross validation the training set of size k is likely to give the best accuracy, because in that case training set would be equal to the validation set.

The outcome of the algorithm is a set of L training indices $\{i_1, \dots, i_L\}$. For the final classification decision regarding the observation \mathbf{x}_{t+1} , a set of L original

THE INSTANCE SELECTION ALGORITHM (FISH2)

Input: labeled observations in \mathbb{R}^p : $\mathbf{x}_1, \dots, \mathbf{x}_t$, target observation \mathbf{x}_{t+1} with unknown label, neighborhood size k , time/space similarity weight A (Equation (II)).

1. Calculate distances $D_j = d(\mathbf{x}_j, \mathbf{x}_{t+1})$ for $j = 1, \dots, t$ (Equation (II)).
2. Sort the distances from minimum to maximum: $D_{z_1} < D_{z_2} < \dots < D_{z_t}$.
3. For $s = k : \text{step} : t$
 - (a) select s instances having the smallest distance D ,
 - (b) using cross-validation^a build a classifier C_s using the instances indexed $\{i_1, \dots, i_s\}$ and test it on the k nearest neighbors indexed $\{i_1, \dots, i_k\}$, record testing error e_s .
4. Find the minimum error classifier C_L , where $L = \arg \min_{L=k}^t (e_L)$.
5. Output the indices $\{i_1, \dots, i_L\}$.

Output: indices $\{i_1, \dots, i_L\}$ to form a training set using $\mathbf{x} \in \mathbb{R}^p$ observations.

^a when test on the instance i_k , this instance is excluded from the validation set

Fig. 3. The Instance selection algorithm (FISH2)

training instances are used $\{\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_L}\}$. The method can be used plugging in various base classifiers.

FISH2 differs from FISH in two main aspects. FISH uses prefixed training size N , while training set size in FISH2 is variable. FISH forms training set handling the instances coming from different classes separately, while FISH2 collects training sample from all the closest instances.

4 Experimental Evaluation

To support the viability of FISH2/FISH, we test them using real datasets along with the peer group of algorithms: Klinkenberg et al. [11] (KLI) and Tsymbal et al. [19] (TSY). Klinkenberg algorithm tries out a set of different training windows and selects the one which shows the best accuracy on the most recent training data. Tsymbal algorithm builds a number of classifiers on different consecutive training subsets used and uses similarity to target observation to select the final classifier. Both use windowing to form the base classifiers. In contrast, FISH 2 builds the multiple classifiers using systematic instance selection.

The motivation for choosing this peer group is to highlight the effect of systematic instance selection which is done by FISH and FISH2. The chosen algorithms use similar framework, based on multiple classifiers, they use no explicit change detection and are base classifier independent.

We include all the history method (ALL) as a benchmark in testing. Every time step the classifier is retrained using all the past data. If the data happens to be stationary, ALL should be the most accurate.

We test the methods using five base classifiers. Parametric Nearest Mean classifier (NMC), non-parametric k Nearest Neighbors classifier (kNN) (for which we take $k = 7$), Parzen Window classifier (PWC) and not pruned decision tree (tree).

4.1 Datasets

We use three real datasets with expected concept drift, three real dataset with artificial drift and one synthetic dataset for illustration of concept.

We made *Luxembourg data* (LU) using European Social Survey¹ [10] 2002-2007. The task is to classify a subject with respect to the internet usage ‘high’ or ‘low’. We use 20 features (31 after transformation of the categorical variables), which were selected as general demographic representation. The set is balanced $977 + 924$.

Ozone level detection [1] (Ozone) represents local ozone peak prediction, that is based on eight hours measurement. Data size 24×2534 . The set is highly imbalanced $160 + 2374$, with only 160 ozone peaks.

Electricity market data (Elec), first described by Harries [8]. We use the time period with no missing values comprised of 6×2956 instances collected from May 11 to July 11, 1997. Labels ‘up’ or ‘down’ indicate the change of the price. The set is moderately balanced $1673 + 1283$.

German credit approval (Cred) [1] classifies customers as having good or bad credit risks. Following [13], a gradual concept change was introduced artificially as a hidden context. We sort the data using one of the features (feature ‘age’ was chosen) and then eliminate this feature from the dataset. Data size 23×1000 . The set is imbalanced $700 + 300$.

Vote data [1] (Vote) represents 1984 United States Congressional Voting Records. The instances represent 435 congressmen (267 democrats, 168 republicans). There are 16 features (votings). The data is categorical, we coded ‘no’ as -1 , ‘yes’ as 1, missing value as 0.

Ionosphere data [1] (Iono) represents Johns Hopkins University Ionosphere data. Binary classification task of the radar returns into ‘good’ and ‘bad’ signals. Data size 43×351 . The set is moderately balanced $136 + 215$.

In Iono and Vote datasets the concept drift is artificial, since we assume the data is presented in a time order and use sequential training-testing procedure. How can we know that there is a concept drift in these datasets? If using this procedure concept drift responsive methods outperform ALL the history method, this can be treated as the drift evidence (non stationarity of the sequential data).

We generate *Hyperplane data* (Hyp) rotating the decision boundary, as shown in Section 3.2. We use 30° rotation for each concept. We have five concepts in total $0^\circ, \dots, 90^\circ$. We generate 10 instances from each concept and 40 instances from each mix of neighboring concept with linearly increasing sampling probability. The size of the data is $65 + 95$.

¹ Norwegian Social Science Data Services (NSD) acts as the data archive and distributor of the ESS data.

4.2 Implementation Details

For FISH and TSY we use training set size $N = 40$, FISH2 and KLI has adaptable set size. KLI and TSY operate in batch mode, we use batch size 15 for both. For TSY we use the following parameters: maximum ensemble size = 7, number of the nearest neighbors = 7 for error estimation. The weights used for time and space similarity for FISH/FISH2 were $a_1 : a_2 = 1 : 1$ for all the data.

For Ozone and Elec data backward search for FISH, FISH2, KLI and TSY was limited to 1100 instances to reduce the complexity of the experiment. For testing with the decision tree using Elec and Ozone data we subsampled taking every 5th instance to speed up the experiments.

4.3 Algorithm Evaluation

We evaluate FISH performance based on the *testing error* and *complexity*.

To evaluate the accuracy, we calculated the ranks of the peer methods. The best method for a given data set was ranked 1, the worst method was ranked 5. The ranks for each data set sum to 15. An average rank over all the datasets was calculated for each classifier and used as performance measure. We exclude the synthetic dataset (hyperplane) from ranking.

To evaluate the applicability, we calculated the worst case and the average complexity of all five peer methods. We counted the number of data passes required to make a classification decision for one observation at time t . The results (approximations) are presented in Table 1. We also present the hyperparameters that needed to be preset for each algorithm.

Table 1. Algorithm complexities. b - batch size; M - ensemble size; k - testing neighborhood size; N - training set size; A - time/space weight; t - time since the start of the sequence.

Method	Worst case	Average	Hyperparameters
ALL	t	the same	–
KLI	$\frac{t^2}{2} + \frac{tb}{2}$	$\frac{t^2}{2b} + \frac{t}{2}$	b
TSY	$t(k+1) + N$	$t(k+1) + \frac{N}{b}$	b, M, k, N
FISH	$t(N+2) + \frac{N(2-N)}{4}$	the same	A, N
FISH2	$\frac{t^2k}{2} + \frac{t(k+2)}{2}$	the same	A, k

The run time is reasonable for sequential data, for all five algorithms it takes up to 1 min for NMC, kNN and PWC and for the decision tree it is ~ 5 times longer to cast a classification decision for *one time observation* on a 1.46 GHz PC, 1GB RAM. For implementation MATLAB 7.5 was used.

Finance, biomedical applications are the domains where the data is scarce, imbalanced, while concept drift is very relevant. For example, in bankruptcy prediction an observation might be received once per day or even per week, while the model needs to be constantly updated and economic cycles imply

Table 2. Testing errors. The best accuracy for each column is underlined. Symbol ‘•’ indicates that the method is significantly worse than FISH2, ‘◦’ indicates that the method is significantly better than FISH2, and ‘–’ indicates no difference (at $\alpha = 0.05$).

method	base	Hyp	LU	Ozone	Elec	Cred	Vote	Iono	rank
FISH2	NMC	13.21	<u>8.05</u>	20.77	17.60	29.03	7.60	<u>16.29</u>	1.67
FISH		<u>11.95</u>	10.37•	<u>8.49</u> ◦	17.66–	<u>28.83</u> –	<u>7.14</u> –	16.57 –	1.67
KLI		15.09	29.63 •	32.04 •	19.22 •	36.24 •	11.29 •	21.71 •	3.25
TSY		13.84	35.89 •	56.45 •	<u>15.47</u> ◦	40.64 •	11.29 •	23.43 •	3.58
ALL		14.47	39.68 •	88.24 •	24.84 •	37.84 •	11.52 •	31.71 •	4.83
FISH2		kNN	13.21	13.58	<u>6.83</u>	17.70	29.03	8.53	<u>21.71</u>
FISH	<u>12.58</u>		16.47 •	<u>6.83</u> –	17.70 –	28.93 –	<u>8.06</u> –	<u>21.71</u> –	2.25
KLI	16.35		16.26 •	7.11 –	17.56 –	30.03 –	9.86 –	22.86 –	3.67
TSY	14.47		28.79 •	7.03 –	<u>13.16</u> ◦	31.43 ◦	10.60 –	23.14 –	4.17
ALL	15.72		<u>11.84</u> ◦	6.99 –	19.86 •	<u>28.83</u> –	8.29 –	22.29 –	2.5
FISH2	PWC		13.21	<u>11.63</u>	77.99	<u>41.18</u>	<u>34.33</u>	<u>8.53</u>	12.86
FISH		<u>12.58</u>	15.16 •	86.89 •	43.62 –	34.43 –	8.76 –	<u>12.57</u> –	3.08
KLI		16.35	14.16 •	<u>65.08</u> ◦	44.53 •	34.63 –	10.37 –	15.14 •	3.67
TSY		15.72	26.42 •	73.33 ◦	43.62 –	36.54 –	9.68 –	19.71 •	4.00
ALL		15.72	11.68 –	86.42 •	43.62 –	34.43 –	<u>8.53</u> –	12.86 •	2.58
FISH2		tree	<u>15.09</u>	0.37	<u>13.24</u>	22.00	<u>31.33</u>	<u>7.14</u>	<u>15.43</u>
FISH	15.72		0.37 –	17.57 •	23.01 –	32.13 –	<u>7.14</u> –	21.43 •	3.58
KLI	16.98		0.37 –	15.12 –	21.66 –	36.34 •	9.68 •	20.86 •	3.67
TSY	15.72		0.37 –	14.82 –	21.15 –	37.04 •	10.14 •	20.57 •	3.33
ALL	17.61		0.37 –	<u>13.24</u> •	21.32 –	32.83 –	7.83 –	18.57 –	2.42

concept drift. In supermarket stock management, stock quantity needs to be predicted once per week, thus only 52 observations are received per year. In such application cases even one hour of the algorithm run for the decision would not be an issue.

5 Results and Discussion

The experimental results of the five algorithms using four alternatives base classifiers with seven datasets ($5 \times 4 \times 7 = 140$ experiments) are provided in Table 2. In order to estimate a statistical significance of the differences between the error rates of two methods, we used the McNemar [14] paired test, which does not require assumption about i.i.d. origin of the data.

The five methods were ranked as presented in Section 4.3 with respect to each data set, and the ranks were then averaged (last column in Table 2). FISH2 has the best rank by a large margin with PWC and tree classifiers, in kNN FISH prevails and for NMC FISH and FISH2 perform equally. The final scores averaged over all three base classifiers are: 1.94 for FISH2, 2.65 for FISH, 3.56 for KLI, 3.77 for TSY and 3.08 for ALL.

Using kNN, PWC and tree as a base classifier, ALL method outperform TSY and KLI according to the rank score. It implies, that under this setting there would be little point in employing those concept drift responsive methods and increasing complexity, as simple retraining (ALL) would do well. FISH is outperformed by ALL using a tree. The results are the worst for Ozone and Iono datasets which are highly unbalanced. FISH handles class imbalance explicitly, but tree classifier itself handles class imbalance well. The results in favor of FISH2 are mostly significant using PWC and tree as base classifiers. Some of the results indicate no statistical difference, the datasets are not large.

FISH2 method is designed to work where concept drift is not clearly expressed. These are the situations of gradual drift, reoccurring contexts. ALL method outperforms all the drift responsive methods but not FISH2 with kNN as a base classifier.

Ozone data is highly unbalanced, FISH method selects training samples from the two classes separately, it significantly outperforms other methods when using parametric base classifier (NMC). Ozone classification according to the highest prior would give 6.74% missclassification rate, however, this would make no sense, as no ozone peaks would be identified at all this way.

Windowing methods work well on Elec data, because the drifts in this data are more sudden. Elec data shows the biggest need for concept drift adaptive methods, because for these datasets ALL method performs relatively the worst from the peer group.

The credits for FISH2 performance in the peer group shall be given to similarity based training set selection. KLI addresses only similarity in time (training window). TSY uses only similarity in time for classifier training, but then they use similarity in space for classifier selection. We employ a combination of time and space similarity already in classifier building phase.

6 Conclusion

We present concept drift responsive method for classifier training based on selecting similar instances to form a training set.

An integration of time and space similarity is a conceptual contribution to concept drifting data mining. Giving 0 weight to the distance in space gives sliding window approach, while 0 weight to time gives instance selection approach, used in multiple classifier systems for stationary cases. In this study 1 : 1 balance is used, weight selection is subject to further investigations. In the future the role of similarity measure in the FISH algorithm family will be addressed as well.

The FISH2 method shows the best accuracy in the peer group on the datasets exhibiting gradual drifts and a mixture of several concepts. The algorithm complexity is reasonable for field applications.

In this pilot study we did not focus on the distance measure. The study can be extended looking at what similarity measure is particularly suitable for the drifting data.

References

1. Asuncion, A., Newman, D.: (UCI) Machine learning repository. University of California, Irvine, School of Information and Computer Sciences (2007), <http://www.ics.uci.edu/~mllearn/MLRepository.html>
2. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally weighted learning. *Artificial Intelligence Review* 11(1-5), 11–73 (1997)
3. Beringer, J., Hüllermeier, E.: Efficient instance-based learning on data streams. *Intell. Data Anal.* 11(6), 627–650 (2007)
4. Bifet, A., Gavaldà, R.: Learning from time-changing data with adaptive windowing. In: *SDM*, pp. 443–448. SIAM, Philadelphia (2007)
5. Fan, W.: Systematic data selection to mine concept-drifting data streams. In: *Proc. 10th ACM SIGKDD*, pp. 128–137. ACM, New York (2004)
6. Gama, J., Medas, P., Castillo, G., Rodrigues, P.P.: Learning with drift detection. In: *17th Brazilian Symposium on AI. LNCS*, pp. 286–295. Springer, Heidelberg (2004)
7. Ganti, V., Gehrke, J., Ramakrishnan, R.: Mining data streams under block evolution. *SIGKDD Explor. Newsl.* 3(2), 1–10 (2002)
8. Harries, M.: Splice-2 comparative evaluation: Electricity pricing. Technical report, The University of South Wales (1999)
9. Hulthen, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: *KDD 2001: Proc. of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 97–106. ACM, New York (2001)
10. Jowell, R., The Central Co-ordinating Team.: European social survey 2002/2003; 2004/2005; 2006/2007. Technical Reports, London: Centre for Comparative Social Surveys, City University, 2003, 2005, 2007
11. Klinkenberg, R., Joachims, T.: Detecting concept drift with support vector machines. In: *Proc. 17th ICML*, pp. 487–494. Morgan Kaufmann, San Francisco (2000)
12. Kolter, J.Z., Maloof, M.A.: Dynamic weighted majority: An ensemble method for drifting concepts. *J. of Machine Learning Research* 8, 2755–2790 (2007)
13. Koychev, I.: Gradual forgetting for adaptation to concept drift. In: *ECAI 2000 Workshop on Current Issues in Spatio-Temporal Reasoning*, pp. 101–106 (2000)
14. Kuncheva, L.I.: *Combining Pattern Classifiers: Methods and Algorithms*. Wiley Interscience, Hoboken (2004)
15. Lazarescu, M., Venkatesh, S.: Using selective memory to track concept effectively. In: *The Proc. of the Int. Conf. on Intell. Sys. and Control*, pp. 14–20 (2003)
16. Nishida, K., Yamauchi, K.: Detecting concept drift using statistical testing. In: Corruble, V., Takeda, M., Suzuki, E. (eds.) *DS 2007. LNCS (LNAI)*, vol. 4755, pp. 264–269. Springer, Heidelberg (2007)
17. Raudys, S., Mitasiunas, A.: Multi-agent system approach to react to sudden environmental changes. In: *Proc. 5th MLDM. LNCS*, pp. 810–823 (2007)
18. Street, W.N., Kim, Y.: A streaming ensemble algorithm (sea) for large-scale classification. In: *Proc. 7th ACM SIGKDD*, pp. 377–382. ACM, New York (2001)
19. Tsybmal, A., Pechenizkiy, M., Cunningham, P., Puuronen, S.: Dynamic integration of classifiers for handling concept drift. *Inf. Fusion* 9(1), 56–68 (2008)
20. Žliobaitė, I.: Instance selection method (fish) for classifier training under concept drift. Technical report, Vilnius University 2009-01 (2009)
21. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: *Proc. 9th ACM SIGKDD*, pp. 226–235. ACM, New York (2003)
22. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. *Machine Learning* 23(1), 69–101 (1996)

Similarity and Kernel Matrix Evaluation Based on Spatial Autocorrelation Analysis

Vincent Pisetta¹ and Djamel A. Zighed²

¹ Rithme, 59 bd Vivier-Merle, 69003 Lyon, France

² ERIC Laboratory, 5 Av. Pierre-Mendès France, 69500 Bron, France
vpisetta@rithme.eu, abdelkader.zighed@univ-lyon2.fr

Abstract. We extend the framework of spatial autocorrelation analysis on Reproducing Kernel Hilbert Space (RKHS). Our results are based on the fact that some geometrical neighborhood structures vary when samples are mapped into a RKHS, while other neighborhood structures do not. These results allow us to design a new measure for measuring the goodness of a kernel and more generally a similarity matrix. Experiments on UCI datasets show the relevance of our methodology.

Keywords: Kernel matrix evaluation, spatial autocorrelation, similarity learning.

1 Introduction

Recently, kernel methods such as Support Vector Machines [1], Kernel K-NN, etc. have delivered extremely high performances in a wide variety of supervised and unsupervised learning tasks. The key to success is that kernel methods proceeds in two steps : the first is to map data into a (usually higher dimensional) feature space ; and the second is to use a linear separator in the feature space, which is efficient and has theoretical guarantees. A large spectrum of linear algorithms can be kernelized, allowing the introduction of nonlinearity implicitly by kernel maps. Such mappings are defined by kernel function:

$$\varphi : X \rightarrow H$$

X is the original data space and H is the feature space, which is usually chosen to be a reproducing kernel Hilbert space (RKHS). Operating directly H is usually impossible because H may be of very high dimension. Kernel methods take advantage of the kernel trick to operate solely on the kernel matrix induced from data:

$$K = \{ \langle \varphi(x_i), \varphi(x_j) \rangle \}_{i=1, \dots, n, j=1, \dots, n}$$

The quality of these methods is really dependent on the choice of a good kernel and features – ones that are typically carried out before the learning process itself. The goodness of a kernel function can only be assessed from the goodness of the kernel matrix K ; therefore, measuring the goodness of K is of primary interest in various contexts.

The most commonly used efficient kernel goodness measure is kernel target alignment (KTA) [5]. Due to its simplicity and efficiency, KTA has been used in many works for two central issues in kernel methods: designing kernels and learning kernels from data. However, it was pointed out in [6] that KTA suffers from several drawbacks, especially because it is only a sufficient condition (and not a necessary condition) to be able to learn well. To overcome this problem, the authors have proposed a new measure of kernel goodness called FSM corresponding to the ratio of the within-class variance in the direction between-class centers to the distance between the class centers in the feature space. Both KTA and FSM have a computational complexity in $O(n^2)$. This complexity does not involve the kernel matrix computation which has a computational cost of $O(n^2)$.

In this paper, we extend the framework introduced in [7] based on spatial autocorrelation statistics in order to assess the goodness of a similarity matrix. Consequently, the work presented here aims at evaluating general similarity matrix and not only kernel matrix, so we do not restrict the user to employ semi-positive definite kernels, but any kind of similarity matrix. We first briefly introduce (section 2) the concept of cut edge weight statistic which is a spatial autocorrelation indicator giving an *a priori* idea of the learning ability in a given space. We analyze the effect of the neighborhood structure used in the statistic computation in RKHS. In section 3, we study the topology induced by different neighborhood graph structures and we give some toy experiments. Section 4 gives experiments of the proposed method for kernel/similarity matrix selection. Finally, in section 5 we conclude.

2 Spatial Autocorrelation and Goodness of Representation Space

2.1 Spatial Autocorrelation Framework

We consider the classical supervised learning scheme. Given a sample set S composed of n training instances x_1, \dots, x_n described in a d -dimensional space R^d and the corresponding vector of responses $y = \{y_1 | \dots | y_n\}$, $y_i \in \{k_1, \dots, k_p\}$, where $\{k_1, \dots, k_p\}$ are categorical labels, we aim at finding a function f such that a loss function $L(y, f(x))$ is minimized. In this context, the goodness of the result (small error rate) is strongly related to the quality of the representation space, i.e., the overall discrimination power of the d features (X_1, \dots, X_d) characterizing an instance x_i . Consequently, defining *a priori* the quality of the representation space is of first importance in order to: (1) avoid wasting time in looking for a discrimination function if the discrimination power of the representation space is low, (2) eventually modifying the representation space by feature selection or feature construction in order to seek better representation independently of the learning algorithm.

The problem of characterizing *a priori* the quality of a representation is not new and has been firstly studied in statistics by Rao[8]. The author considers a situation where the classes are normally distributed and measures their learning degree through a homogeneity test. Similarly, Kruskal and Wallis have defined a distribution-free test based on the hypothesis of scale parameters equality. More recently, in [7], the

authors have proposed a statistical test based on the relative weight of edges connecting points from distinct labels in a neighborhood graph. This test is based on statistical autocorrelation principles and works as follows:

- First a geometrical neighborhood graph based on the representation (X_1, \dots, X_d) is constructed;
- Then, they calculate the sum of the weights of edges connecting points from distinct classes;
- Finally, they compare this sum of weights with the sum of weights that would have been found in a situation of random repartition of classes in the space.

The first step (neighborhood graph construction) aims at defining for each point x_i a set of neighbors. The principle of the test is that if a large majority of instances have neighbors of the same class, then the representation space will be good. Several neighborhood graphs can be used. After, the neighborhood graph computation, they introduce two statistics : the cut edge weight statistic and the uncut edge weight statistic respectively defined as $J = \sum_{i=1}^n \sum_{j=1}^n w_{ij} Z_{ij} E_{ij}$ and $I = \sum_{i=1}^n \sum_{j=1}^n w_{ij} (1 - Z_{ij}) E_{ij}$ where $Z_{ij} = 1$ if x_i and x_j have the same label, 0 otherwise, $E_{ij} = 1$ if x_i and x_j are neighbors, 0 otherwise and w_{ij} is the similarity between x_i and x_j . Finally, the relative cut edge weight statistic is defined as : $RCEW = J/(I + J)$. Actually, $RCEW$ is simply the ratio of the sum of similarity between neighbors of distinct classes by the overall similarity of neighbors whatever their labels. The lower the value of this statistic is, the better is the representation space. In their experiments, the authors have found a very strong correlation between $RCEW$ and the error rate using instance based learning classifiers.

2.2 Spatial Autocorrelation in RKHS

We propose to use the $RCEW$ statistic in order to evaluate the quality of a kernel/similarity matrix. We propose to evaluate the goodness of a kernel matrix (and more generally a similarity matrix) by calculating the statistic $J/(I + J)$ for a given kernel/similarity. To achieve this goal, we simply build a neighborhood graph based on Euclidean distances in the feature space induced by the kernel. These distances are easily calculated in the following manner : using a kernel K , the Euclidean distance between two points x and y in the feature space is defined as : $\partial_F(x, y) = \sqrt{K(x, x) + K(y, y) - 2K(x, y)}$. In this context, the results of spatial autocorrelation analysis will be highly influenced by the employed graph structure. Indeed, in RKHS, the dimension of the induced space can be potentially infinite and will probably lead to drastic changes according to the graph structure employed.

In [7], the authors have used in their experiments the relative neighborhood graph [9], however they have suggested that any connected graphs such as: Gabriel Graph, Minimum Spanning Tree and Delaunay Triangulation could be used. They motivated their choice arguing that the relative neighborhood graph is a good compromise between the number of neighbors and the computational complexity. We give hereafter two definitions of connected graphs commonly used in computational geometry.

Definition 1 (Gabriel Graph). A Gabriel Graph (GG) is a connected graph in which two samples x and y are connected if and only if the following property is verified:

$$\partial^2(x, y) \leq \partial^2(x, z) + \partial^2(y, z), \forall z \in S$$

Where $\partial^2(a, b)$ is the squared Euclidean distance between points a and b . This definition implies that the hypersphere of center $(x + y)/2$ and radius $\partial(x, y)/2$ is empty (i.e, it does not contain any point of S).

Definition 2 (Minimum Spanning Tree). A minimum spanning tree (MST) is a connected graph such that the sum of its connection weights is minimal.

Generally, the weight given to a connection between two samples x and y is equal to the distance between x and y .

Hereafter, we study theoretically the impact of using GG or MST in RKHS. The following theorems prove that under RBF and polynomial kernels (the most commonly used), GG structure (i.e, the set of connected points) will change while the MST structure will stay identical, indicating that GG should be preferred for evaluating kernel.

Theorem 1 (kNN Invariance for RBF kernels and polynomial kernels). Let $kNN_I(x)$ be the set of k nearest neighbors of a pattern x in the input space I , and $kNN_\varphi(x)$ be that of the pattern $\varphi(x)$ in the feature space φ . If the mapping function $\varphi(x)$ is defined such that:

$$\varphi(x) \cdot \varphi(y) = K(x, y) = \exp(-\gamma \|x - y\|^2), \quad \forall \gamma > 0$$

or

$$\varphi(x) \cdot \varphi(y) = K(x, y) = (x \cdot y + 1)^p, \quad \forall p \in \mathbb{N}_+, p \neq 0$$

Then $kNN_I(x) = kNN_\varphi(x)$.

Proof: The proof is given in [10].

Corollary 1 (MST Invariance for RBF and polynomial kernels). Let $MST_I(x)$ be the set of MST neighbors of a pattern x in the input space I , and $MST_\varphi(x)$ be that of the pattern $\varphi(x)$ in the feature space φ . If the mapping function $\varphi(x)$ is defined such that:

$$\varphi(x) \cdot \varphi(y) = K(x, y) = \exp(-\gamma \|x - y\|^2)$$

or

$$\varphi(x) \cdot \varphi(y) = K(x, y) = (x \cdot y + 1)^p, \quad \forall p \in \mathbb{N}_+, p \neq 0$$

Then $MST_I(x) = MST_\varphi(x)$.

Proof: We start by introducing Prim’s algorithm which is one of the well-known procedure allowing the construction of a MST.

MST construction : Prim’s algorithm

Input : The fully weighted connected graph (i.e, each sample is connected to all others) with the sample set V and the set of edges E . Typically, weights are equal to the distances between points.

Initialization : $V_{new} = \{x\}$ where x is a sample selected at random from V , $E_{new} = \emptyset$

Repeat until $V_{new} = V$:

Choose the edge (u, v) from E having the lowest weight (the smallest distance) such that u is in V_{new} et v not in V_{new}

Add v to V_{new} and (u, v) to E_{new}

Output : V_{new} and E_{new} form a MST

Prim’s algorithm selects at random a point from the set S and looks for its nearest neighbor before discarding it from the starting set . By theorem 1, we know that the nearest neighbor of a sample x in the input space is also its nearest neighbor in the feature space under polynomial or RBF kernel. Consequently, Prim’s algorithm will lead to the same connections (i.e, the same set of edges) in the input or feature space. We insist on the fact that even if the connected points are identical in both spaces, distances between them is able to change.

Theorem 2 (GG Variance for RBF kernels and polynomial kernels). Let $GG_I(x)$ be the set of GG neighbors of a pattern x in the input space I , and $GG_\varphi(x)$ be that of the pattern $\varphi(x)$ in the feature space φ . If the mapping function $\varphi(x)$ is defined such that:

$$\varphi(x). \varphi(y) = K(x, y) = \exp(-\gamma\|x - y\|^2)$$

or

$$\varphi(x). \varphi(y) = K(x, y) = (x. y + 1)^p, \forall p \in N_+, p \neq 0$$

Then $GG_I(x) \neq GG_\varphi(x)$.

Proof: The proof is trivial and consists in replacing in the equation of Definition 1 the Euclidean distances in input space by their kernelized values. Hereafter, we give the proof for the RBF kernel. We remind the reader that the squared Euclidean distance in RKHS between two samples x and y is $\partial^2_F(x, y) = K(x, x) + K(y, y) - 2K(x, y)$. x and y are neighbors in input space iff $\partial^2(x, y) \leq \partial^2(x, z) + \partial^2(y, z)$. Imagine we use a RBF network with an hyperparameter $\gamma = 1$, and we have a squared distance between x and y of 12, of 1 between x and z and of 6 between y and z . In the input space, x and y are not neighbors because $12 > 6 + 1$. However, in the feature space induced by the RBF kernel, the squared Euclidean distance between x and y becomes $2 - 2\exp^{-12} = 1.999987$. The distance between x and z becomes 1.2642 and the distance between y and z becomes 1.99504. Then x and y are neighbors in the feature space. The proof is identical with polynomial kernels.

Results of theorem 1 and 2 suggest to use the analysis based on GG in for evaluating the quality of a kernel since not only the pairwise distances but also the graph structure will change. Finally, we point out that as similarity value w_{ij} , we will consider normalized kernel/similarity value defined as: $w_{ij} = K(x_i, x_j) / \sqrt{K(x_i, x_i)K(x_j, x_j)}$. This normalization step is useful for scaling similarities when distinct types of kernels are used.

3 Toy Experiments

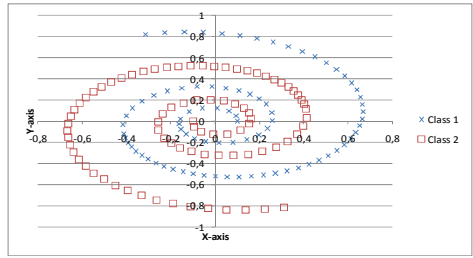
In order to visualize the variance or invariance in feature space of graph structures, we propose to study the topology induced by the graph geometry respectively in the input space and in the feature space. Concretely, we propose to use the ISOMAP algorithm [11] based on Euclidean representations in these spaces. ISOMAP is a manifold learning algorithm which works in the following manner : first, a neighborhood graph (usually symmetric K-NN graph) is constructed over the set S . Then, pairwise Euclidean distances are replaced by geodesic distances, i.e, the distance between two samples x and y becomes the length of the shortest path between x and y along the graph. After these computations, a classical multidimensional scaling [12] is performed on geodesic distances matrix in order to build new features which allow to recover the real dimensionality of data (the structure of the manifold). Figure 1 shows the results of ISOMAP on a toy example based on MST and GG in input and feature spaces. We have also reported the results of the statistic $J/(I + J)$, the SVM classification accuracy (SVM was performed using the LIBSVM library[13]) and the number of edges of the corresponding graph. The values of $J/(I + J)$ and SVM accuracy have been calculated by averaging the results of these two quantities in a 10-fold cross-validation.

Fig. 1 clearly show that the topology induced by the GG drastically changes when datas are mapped into higher dimensional spaces. At the opposite, the topology induced by a MST is really less sensitive to such mappings. The main change only appears in the scales of the new generated features due to the modification of distances in feature space. We also observe that the number of connections using a GG incredibly increases when we map data in high dimensional spaces. This observation was however predictable since the expected number of connections of the GG increases with the dimension of the space.

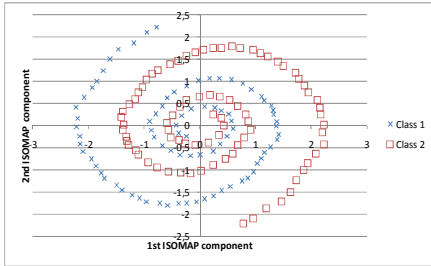
The spatial autocorrelation statistic stays approximately the same using a MST, while it completely changes with a GG when we map data in very high dimensional spaces (RBF kernel). Consequently, the statistic based on GG seems more relevant for evaluating the goodness of a kernel matrix. Moreover, the weighted statistic seems to be related to the error rate of the SVM while it is not very affected using a MST.

4 Experiments

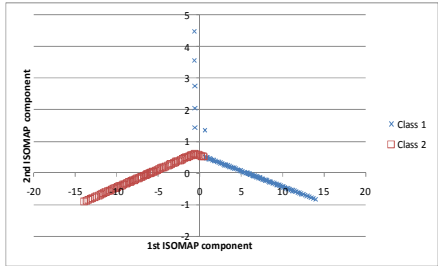
Experiments have been realized on eight datasets from UCI machine learning repository. In order to evaluate the relevance of our measure of goodness, we have calculated the values of KTA, FSM and RCEW based on GG and MST, and SVM error rates (5-fold cross-validation) using three different kernels. The three kernels used are



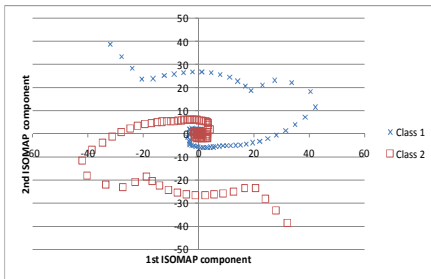
(a) Original dataset



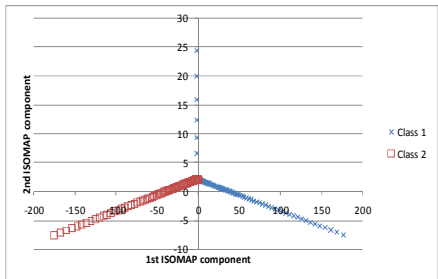
(b) GG (Linear Kernel)



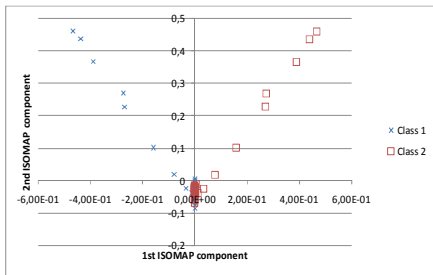
(c) MST (Linear Kernel)



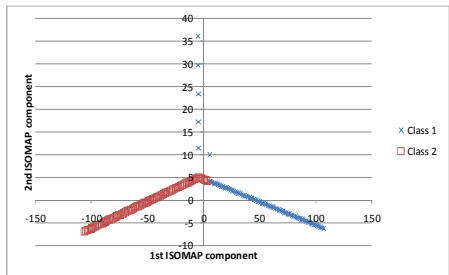
(d) GG (Polynomial Kernel of deg 4)



(e) MST (polynomial Kernel of deg 4)



(f) GG (RBF Kernel with $\gamma = 100$)



(g) MST (RBF Kernel with $\gamma = 100$)

Fig. 1. Results of ISOMAP on MST and GG on a toy dataset with different kernels. (b) : svmacc=0.5 stat=0.5664 nedges=381 ; (c) : stat=0.012 nedges=163. (d) : svmacc=0.49 stat=0.5189 nedges=886 ; (e) : stat=0.0256 nedges=163. (f) : svmacc=0.96 stat=0.042 nedges=13284 ; (g) : stat=0.023 nedges=163.

linear kernel, polynomial kernel of degree 4 and RBF kernel with LIBSVM's default setting. For each of the four measure of goodness, we have ranked kernels from 1 to 3 giving the value of 1 if the corresponding kernel is evaluated as the best and 3 if evaluated as the worst. Before analyzing the results, we insist on the fact that neighborhood graph computation has been realized through an incremental version of the algorithm [14]. This incremental version decreases the computational complexity from $O(n^3)$ to approximately $O(n^2)$ making our approach scalable and in the same time complexity than KTA or FSM.

Separately, we have ranked kernels according to the error rate of the SVM and we have compared the ranking lists. For example, if the best kernel according to error rate is also considered as the best according to the measure of goodness, we report the value of 1. If the best kernel according to SVM error rate is considered as the second best kernel according to the goodness measure, we have reported the value of 2, etc. Table 1 shows the results of the rankings. We can see that the RCEWGG has the best average ranking ex-aequo with FSM, followed by KTA and RCEWMST. RCEWGG and FSM ranks have found the best kernel in half of the cases, while KTA have found the best kernel only for two datasets. RCEWMST is the worst measure retrieving the best kernel only for one dataset.

Table 1. Ranking the best model (in terms of cross-validation error rates) using surrogate measures

Data	RCEWGG	FSM	RCEWMST	KTA
ionosphere	1	1	3	3
heart	1	1	2	3
diabetes	2	1	1	3
german	2	2	2	2
mushrooms	1	1	2	1
vehicle	1	2	3	2
breast-cancer	2	2	2	2
australian	3	3	3	1
Average	1,625	1,625	2,25	2,125

5 Conclusion

We have introduced a new measure of goodness of a kernel matrix based on spatial autocorrelation analysis. To achieve this goal, we first compute a neighborhood graph on the feature space induced by the kernel, and we calculate a statistic relating the relative weights of cut edges of the neighborhood graph. Results on eight UCI datasets show that our methodology of kernel evaluation is the best compared to the well-known KTA and FSM. We have used two types of neighborhood graphs for which we have proven stability or variance in RKHS compared to input space. Our future work

will be concentrated on two distinct aspects. First, we aim at analyzing theoretically variance and invariance of other neighborhood structures (such as the relative neighborhood graph) in RKHS and variance and invariance of neighborhood structures for more complex kernels. The second type of extension is based on searching for optimization methods in order to automatically design an optimal kernel (through convex combination of a set of kernels) for a given learning task. Finally, we insist on the fact that even if this paper had for goal the analysis of kernel matrix, our methodology also works for any kind of similarity matrix (not requiring a semi-positive definite matrix). In this case however, we can think that spatial structure such as MST could be really useful.

References

1. Vapnik, N.V.: *The Nature of Statistical Learning Theory*. Springer, New York (2000)
2. Schölkopf, B., Smola, A.J.: *Learning with kernels*. MIT Press, Cambridge (2002)
3. Fine, S., Scheinberg, K.: Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, Res. 2, 243–264 (2002)
4. Ong, C.S., Smola, A.J., Williamson, R.C.: Learning the kernel with hyperkernels. *Journal of Machine Learning Research*. Res. 6, 1043–1071 (2005)
5. Cristianini, N., Shawe-Taylor, J., Elisseeff, A., Kandola, J.: On kernel-target alignment. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) *Advances in Neural Information Processing Systems*. MIT Press, Cambridge (2001)
6. Nguyen, C., Ho, T.B.: An efficient kernel matrix evaluation measure. *Pattern Recognition* 41(11), 3366–3372 (2008)
7. Zighed, D.A., Lallich, S., Muhlenbach, F.: Separability Index in Supervised Learning. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) *PKDD 2002*. LNCS (LNAI), vol. 2431, pp. 475–487. Springer, Heidelberg (2002)
8. Rao, C.: *Linear statistical inference and its applications*. Wiley, New York (1965)
9. Toussaint, G.: The relative neighborhood graph of finite planar set. *Pattern recognition* 12, 261–268 (1980)
10. Shin, H., Cho, S.: Invariance of neighborhood relation under input space to feature space mapping. *Pattern Recognition Letters* 26, 707–718 (2005)
11. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geomatic framework for nonlinear dimensionality reduction. In: *Advances in Neural Information Processing Systems*, vol. 10, pp. 682–687. MIT Press, Cambridge (2000)
12. Kruskal, J.B., Wish, M.: *Multidimensional Scaling*. Sage University Paper series on Quantitative Application in the Social Sciences, 07-011. Sage Publications, Beverly Hills (1978)
13. Chang, C.C., Lin, C.J.: *LIBSVM: a library for support vector machines* (2001)
14. Hacid, H., Zighed, D.A.: An Effective Method for Locally Neighborhood Graphs Updating. *Database and Expert Systems Applications*. In: 16th International Conference, DEXA, Copenhagen, Denmark (2005)

Job Offer Management: How Improve the Ranking of Candidates

Rémy Kessler¹, Nicolas Béchet², Juan-Manuel Torres-Moreno¹,
Mathieu Roche², and Marc El-Bèze¹

¹ LIA / Université d'Avignon, 339 chemin des Meinajariès, 84911 Avignon
² LIRMM - UMR 5506, CNRS / Université Montpellier 2, France
{remy.kessler,juan-manuel.torres,marc.elbeze}@univ-avignon.fr,
{nicolas.bechet,mathieu.roche}@lirmm.fr

Abstract. The market of online job search sites grows exponentially. This implies volumes of information (mostly in the form of free text) become manually impossible to process. An analysis and assisted categorization seems relevant to address this issue. We present E-Gen, a system which aims to perform assisted analysis and categorization of job offers and of the responses of candidates. This paper presents several strategies based on vectorial and probabilistic models to solve the problem of profiling applications according to a specific job offer. Our objective is a system capable of reproducing the judgement of the recruitment consultant. We have evaluated a range of measures of similarity to rank candidatures by using ROC curves. Relevance feedback approach allows to surpass our previous results on this task, difficult, diverse and highly subjective.

1 Introduction

The exponential growth of Internet allowed the development of a market for online job-search [1,2]. Over last few year it is in a significant expansion (August 2003: 177 000 job offers, May 2008: 500 000 job offers) [4]. The Internet has become essential in this process because it allows a better flow of information, either through job search sites or by e-mail exchanges. The answers of candidates confer a lot of information that cannot be managed efficiently by companies [3]. Even though a browser has become a universal and easy tool for the users, frequent need to enter data into Web forms from paper sources, "copy and paste" data between different applications, is symptomatic of the problems of data integration. Therefore it is essential to process this information by an automatic or assisted way. We developed the E-Gen system to resolve this problem.

It is composed of three main modules:

1. The first module extracting the information from a corpus of e-mails of job offers from Aktor's database [5].

¹ www.keljob.com

² Aktor Intéactive (www.aktor.fr).

2. The second module analysing the candidate answers (splitting e-mails into Cover Letter (CL) and Curriculum Vitae (CV)).
3. The third module analysing and computing a relevance ranking of the candidate answers.

Our previous works present the first module [4] the identification of different parts of a job offer and the extraction of relevant information (contract, salary, localization etc.). The second module analyses the content of a candidate's e-mail, using a combination of rules and machine learning methods (Support Vector Machines, SVM). Furthermore, it separates the distinct parts of CV and CL with a Precision of 0.98 and a Recall 0.96 [5]. Reading a large number of candidate answers for a job is a very time consuming task for a recruiting consultant. In order to facilitate this task, we propose a system capable of providing an initial evaluation of candidate answers according to various criteria. In this paper, we present the last module of E-Gen. Some related works are briefly discussed in section 2. Section 3 shows a general system overview. In section 4, we describe the pre-processing task and strategy used to rank the candidate answers. In section 5, we present statistics about the textual corpus, experimental protocol and results.

2 Related Work

Many approaches have been proposed in literature to reduce the costly and tedious task of managing the Human Resources. Candidate answers to a job-offers are particular and ad hoc documents, it allows to develop semantic approaches to analyse than. [6] propose an indexing method based on the *BONOM* system [7]. Their method consists of using distributional attributes of documents to locate each part to finally index the document. A semantic-based method to select candidate answers and to discuss the economical impacts in the German government was proposed by [8]. Limitations of actual systems of automatic selection of candidate answers are presented in [2]. They propose a system based on collaborative filters (*ACF*) to automatically select profiles of candidate answers in the *JobFinder* Website. [9] discuss the relevance of a common ontology (*HR ontology*) to working efficiently with this kind of documents. [3] describe an ability model and a management tool used for the candidate-answers selection. Using the same model, [10] outline a *HR-XML* based prototype dedicated to the job search task. The prototype selects and favors relevant information (pay-check, topic, abilities, etc.) from many job-service Websites, such as *Jobs.net*, *aftercollege.com*, *Directjobs.com* etc.

The study of the more relevant document – the CV – to use it automatically has been a subject of many researches. [11] propose a data mining approach. Their aim is to build automates which recognize CV topologies and candidate/job-offers profiles. A first step differentiates the CV of executive employed from other CV employed. They make a specific term extraction to obtain a categorization with the C4.5 decision tree algorithm [12]. This method focuses on the specificity of selected terms or concepts, as education level or relevant abilities, to build a classifier. The method results are yet poor (an accuracy between

0.5-0.6 of correctly categorized CV). [13,14] have made a terminology study of corpus composed by CV (of the Vedior Bis company (<http://www.vediorbis.com>)). Their approach allows to extract collocations from CV corpus based on syntactic patterns as Noun-Noun, Adjective-Noun, etc. Then these collocations are ranked by relevance to build a specialized ontology. In this paper, we present an approach to the candidatures ranking by using a combination of similarity measures and Relevance Feedback.

3 System Overview

Nowadays technology proposes new ways of on-line employment market. We propose a system which answers as fast and judiciously as possible to this challenge. An e-mail-box receives messages containing the offer. Firstly, the job offer language is identified by using n -grams. Then, E-Gen parses the e-mail, splits the offer into segments, and retrieves relevant information (contract, salary, location, etc.). Subsequently a filtering and lemmatisation process is applied to text and it will be represented in a vector space model (VSM). A categorization of text segments (Preamble, Skills, Contacts,...) is obtained by means of Support Vector Machines. This preliminary classification is afterwards transmitted to a “corrective” post-process which improves the quality of the solution (Task 1, described in [4]). During the publication of a job offer, Aktor generates an e-mail address for applying to the job. Each e-mail is redirected to a Human Resources software, (Gestmax³) to be read by a recruiting consultant. At this step, E-Gen analyses the candidate’s answers to identify each part of the candidacy and extracts the text from e-mail and attached files (by using wvWare⁴ and pdftotext⁵). After a pre-processing task, we use a combination of rules and machine learning methods to separate each distinct part (CV or CL). The process (task 2) is described in [5]. Once CL and CV are identified, the system performs an automated profiling of this candidature by using measures of similarity and a small number of candidatures previously validated as relevant candidatures by a recruitment consultant (Task 3). The whole of the chain of E-Gen System is represented in figure 1.

4 Ranking of Candidatures

4.1 Corpus Pre-processing

A classical pre-processing is applied to Textual information (CV and CL). We remove information such as names of candidates, addresses, e-mails, names of cities. Accents are deleted and capital letters are normalised. In order to avoid the introduction of noise into the models⁶, the following items are also deleted:

³ <http://www.gestmax.fr>

⁴ <http://wwware.sourceforge.net>

⁵ http://www.bluem.net/downloads/pdftotext_en

⁶ These pre-processing are not applied in the n -grams representation.

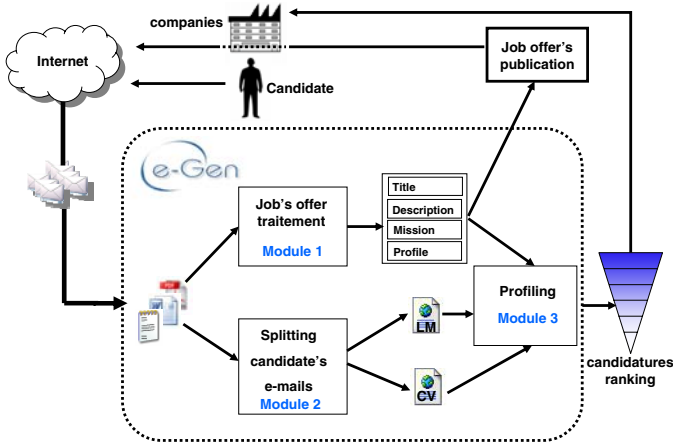


Fig. 1. System overview

verbs and functional words (to be, to have, to need,...), common expressions with a stop words list⁷(for example, that is, each of,...), numbers (in numeric and/or textual format), symbols such as “\$”, “#”, “*”. Finally, lemmatisation⁸ is performed to significantly reduce size of the lexicon. All these processes allow us to represent the collection of documents through the bag-of-words paradigm (a matrix of frequencies of terms (columns) for each candidate answer (rows)).

4.2 Comparison between Candidatures and Job Offer Using Similarity Measure

Each document is transformed into a vector with weights characterizing the frequency of terms Tf and $Tf-idf$ [15].

We have established a strategy using measures of similarity, to rank all candidatures in relation to a job offer. We combined different similarity measures between the candidate answers (CV and CL) and the associated job offer. We also tested several similarity measures as defined in [16]: *cosine* (1), which calculates the angle between job offer and each candidate answer, Minkowski distances (2) ($p = 1$ for Manhattan, $p = 2$ for euclid). The last measure used is Okapis (3) [17]. Based on okapi [18] formula, this measure is often used in Information Retrieval.

$$sim_{cosine}(j, d) = \frac{\sum_{i=1}^n j_i \cdot d_i}{\sqrt{\sum_{i=1}^n j_i^2 \cdot \sum_{i=1}^n d_i^2}} \tag{1}$$

$$sim_{Minkowski}(j, d) = \frac{1}{1 + (\sum_{i=1}^n |j_i - d_i|^p)^{\frac{1}{p}}} \tag{2}$$

⁷ <http://sites.univ-provence.fr/~veronis/donnees/index.html>

⁸ Lemmatisation finds the root of verbs and transforms plural and/or feminine words to masculine singular form. So we conflate terms *sing*, *sang*, *sung*, *will sing* into *sing*.

$$\text{Okabis}(d, j) = \sum_{i \in d \cap j} \frac{\sum_{i=1}^n j_i \cdot d_i}{\sum_{i=1}^n j_i \cdot d_i + \frac{\sqrt{|d|}}{M_d}} \quad (3)$$

where j is a job offer, d is a candidate answer, i a term, j_i and d_i occurrence of i respectively in j and d , and M_d their average size.

Several other similarity measures (Overlap, Enertex, Needleman-Wunsch, Jaro-Winkler) have been tested but they are not retained in this study, because the results obtained are disappointing. All measures used and their combinations are described in [19].

4.3 Extraction of Features

In the following sections, we describe a number of features that will be used to represent the documents. These features are based on grammatical information, n -grams of characters and semantic information.

Filtering and Weighting of Words According to their Grammatical Label. To improve the performance of similarity measures (section 4.2), we performed an extraction of grammatical information in the corpus with TreeTagger⁹ [20]. We found that CV are short documents (usually not exceeding one page) and syntactically poor: few subjects and verbs in sentences, sentences in summary form, many lists of nouns and adjectives, etc [13]. The words respecting specific grammatical labels can thus be more or less interesting. We propose to extract the following terms : **N** (Noun) **A**(adjective) **V**(Verb). These terms alone will be selected as the basis of the vector representation of documents. We tested different combinations and weights.

Character n -Grams. Mainly used in speech recognition, n -grams of characters have been used in text analysis [21]. Research shows the effectiveness of n -grams as a method of text representation [22,23]. An n -gram is like a moving window over a text, where n is the number of character in the window. An n -gram is a sequence of n consecutive characters. The move is processed by steps, one step related to one character. Then the frequencies of n -grams found are computed. For example, the sentence "developer php mysql" is represented with tri-grams [dev, eve, vel, elo, lop, ope, per, er_, r_p, _ph, php, hp_, p_m, _my, mys, ysq, sql]. We represent the space in the n -grams by using the "_". This representation automatically captures the most stem of words, avoiding lexical root research. The second interest of this representation is their tolerance to spelling mistakes and typographical errors often found in CV and CL¹⁰. We tested different n -size windows ($3/4/5/6$ -grams).

⁹ TreeTagger is a tool for annotating text with part-of-speech and lemma information. It is downloadable at

<http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>

¹⁰ For example, a words system will have difficulty recognizing the word "Developer" misspelled (with two p).

Semantic Enrichment of the Job Offer. Observation of terms with the most influence when computing the similarity measure, led us to consider enhancing the content of the job offer with an ontology derived from the base ROME^[11] from ANPE^[12]. We enriched each job with skills and educational levels expected^[13].

Relevance Feedback. We changed the system to incorporate a process of Relevance Feedback [24]. Relevance feedback is a classical method used particularly for manual query reformulation. For example, the user carefully checks the answer set resulting from an initial query, and then reformulates the query. Rocchio algorithm [25] and variations have found wide usage in information retrieval and related areas such as text categorisation [26]. Relevance Feedback has been proposed [27] to help the user to find a job with server logs from the site JobFinder^[14].

In our system, Relevance Feedback takes into account the recruiting consultant choice during a first evaluation of few CVs. Our goal is not a system capable of finding the best candidate, but a system capable of reproducing the judgement of the recruitment consultant. It is critical for recruiters not to miss a good candidate that they may have unfortunately rejected. The goal of this Relevance Feedback approach is to help them to avoid this kind of error. This approach exploits documents returned in response to a first request to improve the search results [28]. In this case, we randomly take few candidate answers (one to six in our experiments) amongst all relevant candidate answers. These are added to the job offer. So we use manual relevance feedback to reflect the user judgements in the resulting ranking. We increase the vector representation with the terms from the candidates considered relevant by a recruitment consultant. System will recompute similarity between the candidate's answer that we evaluate and job offer enriched with relevant candidates.

5 Experiments

We have selected a data subset from Aktor's database. This subset is called *Corpus Mission*. It contains a set of job offers with various thematics (jobs in accountancy, business enterprise, computer science, etc.) and their candidates. As described in [19], each document is segmented to keep relevant parts (we remove the description of the company for the job offer and the last third of CV and CL). Each candidate is tagged **relevant** or **irrelevant**. A **relevant** value

¹¹ *Répertoire Opérationnel des Métiers et des Emplois*, Operational List of Jobs and Skills.

¹² *Agence National Pour l'Emploi*, French National Agency for Employment <http://www.anpe.fr/espacecandidat/romeligne/RliIndex.do>

¹³ Example: 32321/developer/**Bac+2** à **Bac+4** in **computing CFPA, BTS, DUT;development and maintenance of computing applications, functional analysis, engineering design, coding, development and documentation of programs** etc.

¹⁴ JobFinder (<http://www.jobfinder.com>).

corresponds to a potential candidate for a given job chosen by the recruiting consultant. A **irrelevant** value is associated to an unsuitable candidate for the job (this is a decision if the human resources of the company). Our study was conducted on french job offers because the french market represents Aktor's main activity. Table 1 shows a few statistics about the *Corpus Mission*.

Table 1. Corpus statistics

Number	Job's Title	Number of candidate answers	Number of	
			relevant	irrelevant
34861	sales engineer	40	14	26
31702	accountant, Department suppliers	55	23	32
33633	sales engineer	65	18	47
34865	accountant assistant	67	10	57
34783	accountant assistant	108	9	99
33746	3 chefs	116	60	56
33553	Trade Commissioner	117	17	100
33725	urban sales consultant	118	43	75
31022	recruitment assistant	221	28	193
31274	accountant assistant junior	224	26	198
34119	sales assistant	257	10	247
31767	accountant assistant junior	437	51	386
Total		1917	323	1594

5.1 Experimental Protocol

We want to measure the similarity between a job offer and its candidate's answers. *Corpus Mission* is composed of 12 job offers associated with at least 9 candidates identified as **relevant** for each one. These measures (section 4.2) rank the candidate answers by computing a similarity between a job offer and their associated candidate answers.

We use the ROC curves to evaluate the quality ranking obtained. ROC curves [29] come from the field of signal processing. They are used in medicine to evaluate the validity of diagnostic tests. In our case, ROC curves show the rate of irrelevant candidate answers on X-axis and the rate of relevant candidate answers on Y-axis. The *Area Under the Curve (AUC)* can be interpreted as the effectiveness of a measurement of interest. In the case of candidate answers ranking, a perfect ROC curve corresponds to obtain all relevant candidate answers at the beginning of the list and all irrelevant at the end. This situation corresponds to $AUC = 1$. The diagonal line corresponds to the performance of a random system, progress of the rate of relevant candidate being accompanied by an equivalent degradation of the rate of irrelevant candidate. This situation corresponds to $AUC = 0.5$. An effective measurement of interest to order candidate answers consists in obtaining the highest AUC value. This is strictly equivalent to minimizing the sum of the ranks of the relevant candidate's answers. ROC curves are resistant to imbalance (for example, an imbalance in number of positive and negative examples) [13]. For each job offer, we evaluated the quality of ranking obtained by this method. Candidate answers considered are only those composed of CV and CL.

5.2 Results

Table 2 shows the best results obtained for each method. Each test is carried out 100 times with a random distribution of relevant candidatures for Relevance Feedback. Then we compute an average of AUC scores obtained (the curve shows the average for each size). The *TF* corresponds to the results obtained with the frequency of each term as unit. *TF-IDF* uses the product of terms frequency and inverse document frequency. *TF* and *TF-IDF* representations give globally similar results with *AUC* score at 0.64. Small size of corpus used can explain these results. Using combination and weighting of grammatical classes representation (*Grammatical Labels*) gives also close results. *N*-grams results are obtained with 5-grams. With *AUC* score at 0.6, *n*-grams results are poor. We plan, in order to improve the *n*-grams results, to find and remove frequent and insignificant strings. *Job offer enriched* corresponds to the results obtained with semantic enrichment of job offer. With *AUC* score at 0.62, semantic expansion does not improve referent results. Additional information about job offer are not required and it seems degrade performance of the system but additional tests are necessary.

Figure 2 presents results obtained with different sizes of relevance feedback (RF1 corresponds to one candidature added to the job offer, RF2 two, etc.). We use actually *residual ranking* [30]: documents that are used for relevance feedback are removed from the collection before ranking with the reformulated query. We observe that Relevance Feedback allows to improve the results more significantly.

Table 2. Comparison of *AUC* score for each method

	<i>N</i> -grams	<i>Job offer enriched</i>	<i>TF</i>	<i>TF-IDF</i>	<i>Grammatical Labels</i>	<i>Relevance Feedback</i>
Job offer/CV and CL	0.60	0.62	0.64	0.64	0.64	0.66

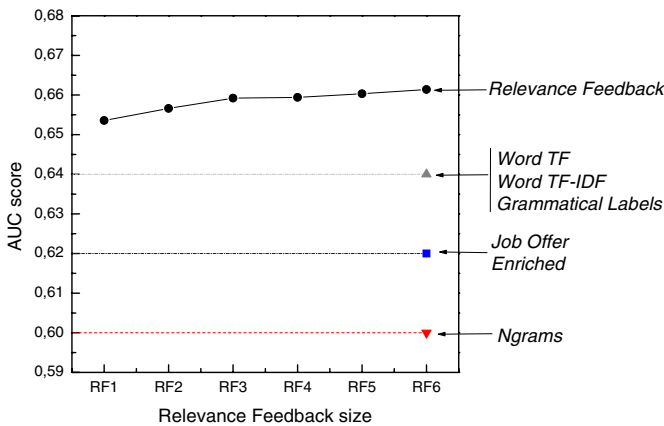


Fig. 2. Comparison of *AUC* score for each size of Relevance Feedback

RF1 gives an average *AUC* score at 0.65 and RF6 at 0.66. Currently, we study results for each mission, but they are quite disparate. For example, mission 33725 shows a good increase between each size of relevance feedback (*TF*: 0.595, RF1: 0.685, RF6:0.716) while for others the increase was less obvious (mission 33633 *TF*: 0.561, RF1: 0.555, RF6:0.579). The study of results shows that some missions has some empty candidate with label **relevant**. This leads the system to degrade performance when they are selected. Note that it is impossible to experiment RF n with $n > 6$ because of the number of candidates too small for some job offers (see table [II](#)).

6 Conclusion and Future Work

The processing of a job offer is a difficult and highly subjective task. The information we use in this kind of process is not well formatted in natural language, but follows a conventional structure. In this paper, we present the third module of the E-Gen project, a system for processing of a job-offer. The system allows to assist an employer in a recruitment task. The third module we presented in this paper focuses on candidate-answers to job offers. We rank the candidate answers by using different similarity measures and different document representations in vector space model. We choose to evaluate the quality of our approaches by computing *Area Under the Curve*. *AUC* obtained with our relevance-feedback-based-approach shows an improvement of result. As future work, we plan to apply other treatments, such as finding discriminant features of irrelevant candidatures to use Rocchio algorithm [\[25\]](#), weighting the different segments of a mission, etc. to improve results. We also plan to take into account other parameters such as vocabulary used and spelling. Thus we will perform a better analysis of the cover letters. Actually, CL are not really used by an employer in a decision process. Finally we propose to measure the CV quality by building an evaluation in a Internet portal. Our aim with this evaluation is to present to a job-finder a list of relevant job-offers in agreement with this profile.

Acknowledgement

Authors thank to Piotr Więcek.

References

1. Bizer, R.H., Rainer, E.: Impact of Semantic web on the job recruitment Process. In: International Conference Wirtschaftsinformatik (2005)
2. Rafer, R., Bradley, K., Smyt, B.: Automated Collaborative Filtering Applications for Online Recruitment Services. In: Brusilovsky, P., Stock, O., Strapparava, C. (eds.) AH 2000. LNCS, vol. 1892, pp. 363–368. Springer, Heidelberg (2000)
3. Bourse, M., Leclère, M., Morin, E., Trichet, F.: Human resource management and semantic web technologies. In: ICTTA, pp. 641–642 (2004)

4. Kessler, R., Torres-Moreno, J.M., El-Bèze, M.: E-Gen: Automatic Job Offer Processing system for Human Ressources. In: MICAI 2007, Agusalientes, Mexique, pp. 985–995 (2007)
5. Kessler, R., Torres-Moreno, J.M., El-Bèze, M.: E-Gen: Profilage automatique de candidatures. In: TALN 2008, Avignon, France, pp. 370–379 (2008)
6. Morin, E., Leclère, M., Trichet, F.: The semantic web in e-recruitment (2004). In: The First European Symposium of Semantic Web, ESWS 2004 (2004)
7. Cazalens, S., Lamarre, P.: An organization of internet agents based on a hierarchy of information domains. In: Proceedings MAAMAW (2001)
8. Tolksdorf, R., Mocho, M., Heese, R., Oldakowski, R., Christian, B.: Semantic-Web-Technologien im Arbeitsvermittlungsprozess. In: International Conference Wirtschaftsinformatik, pp. 17–26 (2006)
9. Mocho, M., Paslaru, E., Simperl, B.: Practical Guidelines for Building Semantic eRecruitment Applications. In: I-Know 2006 Special track on Advanced Semantic Technologies (2006)
10. Dorn, J., Naz, T.: Meta-search in human resource management. In: Proceedings of 4th International Conference on Knowledge Systems ICKS 2007, Bangkok, Thailand, pp. 105–110 (2007)
11. Clech, J., Zighed, D.A.: Data mining et analyse des cv: une expérience et des perspectives. In: EGC 2003, pp. 189–200 (2003)
12. Quilan, J.: C4.5: Programs for machine learning. Kaufmann, San Mateo (1993)
13. Roche, M., Kodratoff, Y.: Pruning Terminology Extracted from a Specialized Corpus for CV Ontology Acquisition. In: OTM 2006, Montpellier, France, pp. 1107–1116 (2006)
14. Roche, M., Prince, V.: Evaluation et détermination de la pertinence pour des syntagmes candidats à la collocation. In: JADT 2008, pp. 1009–1020 (2008)
15. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York (1986)
16. Bernstein, A., Kaufmann, E., Kiefer, C., Bürki, C.: Simpack: A generic java library for similarity measures in ontologies. Technical report, University of Zurich (2005)
17. Bellot, P., El-Bèze, M.: Classification et segmentation de textes par arbres de décision. In: TSI, vol. 20, pp. 107–134. Hermès (2001)
18. Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M.: Okapi at trec-3. NIST Special Publication 500-225: TREC-3, pp. 109–126 (1994)
19. Kessler, R., Béchet, N., Roche, M., El-Bèze, M., Torres-Moreno, J.M.: Automatic profiling system for ranking candidates answers in human resources. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 625–634. Springer, Heidelberg (2008)
20. Schmid, G.: Treetagger - a language independent part-of-speech tagger. In: Proceedings of EACL-SIGDAT 1995, Dublin, Ireland, pp. 44–49 (1994)
21. Damashek, M.: Gauging similarity with n-grams: Language-independent categorization of text. *Science* 1995 267, 843–848 (1995)
22. Mayfield, J., Mcnamee, P.: Indexing using both n-grams and words, pp. 500–242. NIST Special Publication (1998)
23. Hurault-Plantet, M., Jardino, M., Illouz, G.: Modèles de langage n-grammes et segmentation thématique. Actes TALN & RECITAL 2, 135–144 (2005)
24. Spärck Jones, K.: Some thoughts on classification for retrieval. *Journal of Documentation*, 89–101 (1970)
25. Rocchio, J.: Relevance Feedback in Information Retrieval, pp. 313–323 (1971)
26. Thorsten, J.: A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In: ICML 1997, pp. 143–151 (1997)

27. Smyth, B., Bradley, K.: Personalized Information Ordering: A Case-Study in Online Recruitment. *Journal of Knowledge-Based Systems*, 269–275 (2003)
28. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 288–297 (1990)
29. Ferri, C., Flach, P., Hernandez-Orallo, J.: Learning decision trees using the area under the ROC curve. In: *Proceedings of ICML 2002*, pp. 139–146 (2002)
30. Billerbeck, B., Zobel, J.: Efficient query expansion with auxiliary data structures. *Inf. Syst.* (7), 573–584 (2006)

Discovering Structured Event Logs from Unstructured Audit Trails for Workflow Mining

Liqiang Geng¹, Scott Buffett¹, Bruce Hamilton¹, Xin Wang², Larry Korba¹,
Hongyu Liu¹, and Yunli Wang¹

¹ IIT, National Research Council of Canada, Fredericton, Canada, E3B 9W4

² Department of Geomatics Engineering, University of Calgary, Canada, T2N 1N4

{liqiang.geng, scott.buffett, bruce.hamilton}@nrc.gc.ca,

xcwang@ucalgary.ca,

{larry.korba, hongyu.liu, yunlin.wang}@nrc.gc.ca

Abstract. Workflow mining aims to find graph-based process models based on activities, emails, and various event logs recorded in computer systems. Current workflow mining techniques mainly deal with well-structured and -symbolized event logs. In most real applications where workflow management software tools are not installed, these structured and symbolized logs are not available. Instead, the artifacts of daily computer operations may be readily available. In this paper, we propose a method to map these artifacts and content-based logs to structured logs so as to bridge the gap between the unstructured logs of real life situations and the status quo of workflow mining techniques. Our method consists of two tasks: discovering workflow instances and activity types. We use a clustering method to tackle the first task and a classification method to tackle the second. We propose a method to combine these two tasks to improve the performance of two as a whole. Experimental results on simulated data show the effectiveness of our method.

1 Introduction

Workflow mining refers to the task that automatically finds business process models within an enterprise by analyzing the computer operations, usually in the form of event logs, by a group of people involved in the process. These process models are usually represented in graphs and can be used to reengineer the work process and to ensure that the employees comply with the standard procedures.

Currently, most of the techniques for workflow mining require that structured (symbolized) event logs are available from certain software tools, such as Enterprise Resource Planning (ERP) systems, Customer Relationship Management (CRM) software, or Workflow Management systems [1]. However, in most real situations, these systems may not be installed in the enterprises. Therefore, structured event logs are not readily available. Instead, what we can have in most situations is the unstructured logs related to activities, such as emails sent and received, web pages accessed, documents edited, and applications executed. This kind of information can be easily obtained from web, email, and application servers. These unstructured logs do not record the purpose of the operations, i.e., activity labels for the operations, and the labels for the workflow instances that these operations belong to. For example, an

email message recorded in an unstructured log may contain keywords “trip”, “application”, “July”, “London”, “Richard”, and “Smith”. We neither know the corresponding activity type for this email, nor do we know the process instance to which the email belongs from the email itself. Unfortunately, current workflow mining techniques based on the structured logs cannot be used in such situations [1, 2]. In this paper, we propose a method to identify the activity label and process instance for each event in the unstructured logs based on the keywords and named entities identified from the content of the event, and people involved in the operation. In this way the unstructured logs can be converted to structured logs, and hence can be used as the input to the workflow mining algorithms. For example, the task of the email message described above may be labeled with activity label *Trip-Application* and process instance label *Smith-London-Trip*.

The past decade has seen much work conducted in the field of workflow mining. The most investigated problem is to create graph-based workflow models from structured logs. A number of different graphic representations have been used for workflow mining. These include directed acyclic graph [3], finite state machine [6], variation of Bayesian network [10], workflow schema [7], and Petri Net [1, 2]. Based on these representations, various algorithms have been proposed. However, all of the above-mentioned work is based on structured logs where both activity types and process instances are recorded.

Recently some researchers have started to pay attention to the content-based methods which utilize data mining techniques to decide whether to label two events as the same activity type or as part of the same workflow instance. Kushmerick and Lau used data mining methods to identify activities and transitions of activities from email messages [9]. However, that work is focused on solving a very specialized problem in E-commerce transactions. Also, they treat the task of indentifying activities and that of identifying workflow instances separately. Khoussainov and Kushmerick [8] reported a method to identify links and relations between email messages and the email speech acts [5]. However, their claim that integration of link identification and speech act identification will improve the performance for each other is based on two small data sets. In our paper, we performed systematic experiments on synthetic data sets and found that the combination of the two tasks of our problem does not necessarily improve the performance for each other.

One issue arising from analyzing the content of the artifacts obtained from computer operations is the privacy of employees in organizations. Sometimes, there are conflicts between an organization’s confidentiality and its employees’ privacy, i.e., analyzing workflow and ensuring compliance in an organization may result in a violation of employees’ privacy. However, in this paper, we will ignore the privacy and confidentiality issues and focus on the technical aspects for workflow mining.

The contributions of this paper include the following:

- (1) We proposed a novel method that uses the *transition matrix* and *preceding matrix* to combine the results of the activity identification and workflow instance identification. This method takes into account the keywords, named entities, as well as the sequence information embodied in the unstructured logs.

- (2) Our work is the first in this field based on the systematic experiments on synthetic data sets. In this way, we can see how data itself can affect the performance.

(3) We obtained observations that combining the instance identification and activity identification would not necessarily improve the results for each other, which is contrary to the claims made in [8]. We found that the quality of the data and the results of initial identification of the activities and workflow instances play a key role in the final results.

In section 2, we introduce basic concepts and state the problem we will tackle. In Section 3, we present our method to combine activity identification and instance identification tasks. In Section 4, we present the design of experiments and the experimental results. Section 5 concludes the paper.

2 Preliminaries

In this paper, we refer to the unstructured logs as *audit trails* and the structured logs as *event logs*. Audit trails and event logs are defined as follows.

An *audit trail entry* ate is a 5-tuple $(Op, SO, Rec, Cont, TS)$, where Op refers to the operation type, SO refers to the operators, Rec refers to the recipients if applicable, $Cont$ refers to the content of the artifacts related to the operation, and TS refers to the time stamp. An *audit trail* AT is a set of audit trail entries ranked by timestamp in an ascending order.

Given a set of activity types A , an *event* e is a 3-tuple (Ins, Act, TS) , where Ins is an integer referring to the workflow instance label. $Act \in A$ is an activity type. TS is the time stamp. An *event log* EL is a set of events ranked by the timestamp in an ascending order.

In some literature, *activity*, *task*, and *event* are used interchangeably. In this paper, *activity* refers to the type of an *event*, while an *event* refers to a step in a workflow. Therefore, different events may have same activity type. We will avoid using *task* to eliminate the ambiguity.

Table 1 shows an example of an audit trail. In our work, we take into account three types of user operations: document editing, email sending, and web form submission. In the table, each row represents an audit trail entry. The first event in the audit trail says that Zhang sent an email to Johnson to inform him the acceptance of their paper. The second event says that Johnson was drafting a document to apply for a travel for a conference. The last event says that Bergman was booking an air ticket on an airline web site. The audit trail did not record explicit semantic labels for these entries, nor did it show which events are correlated for a workflow instance.

Table 1. An example of audit trail

Operation	Time	Sender / Operator	Recipients	Content
email	09/02/03 00:00:00	Zhang	Johnson	Hi Mike, our paper has been accepted ...
doc	09/03/04 01:01:01	Johnson		The purpose of this travel is to learn the latest development in ...
email	09/03/05 09:09:09	Johnson	Bergman	Hi Sarah, In August, I will have a trip to Boston for a conference ...
web	09/03/06 09:09:09	Bergman		Air Canada ticket center...

Table 2 illustrates an example of an event log. In this table, we have four activity types *Apply*, *Approve*, *Decline*, and *Reimburse*, and two workflow instances: Instance 1 consists of three activities {*Apply*, *Approve*, *Reimburse*} and Instance 2 consists of two activities {*Apply*, *Decline*}. This table is a standard input for workflow mining algorithms that construct a graphic workflow model.

Table 2. An example of event log

Instance No.	Activity Types	Time
1	Apply	09/01/03 09:25:08
2	Apply	09/01/05 12:39:02
1	Approve	09/02/01 10:22:50
2	Decline	09/02/02 09:07:45
1	Reimburse	09/03/02 13:34:23

Problem Statement: Given an audit trail *AT*, convert it to an event log *EL*.

3 Converting Audit Trails to Event Logs

The mapping from an audit trail to an event log consists of two tasks: grouping related events for the same workflow instance and identifying the activity type for each event. Our integrated method includes the following steps: (1) Use a clustering method to identify the initial workflow instances. (2) Use a Naïve Bayesian classifier to identify the initial activity types. (3) Generate a *transition matrix* and *preceding matrix* based on the results from previous two steps. (4) Recluster the workflow instances with the transition matrix. (5) Reclassify activity types with the preceding matrix. The mining process is shown in Figure 1.

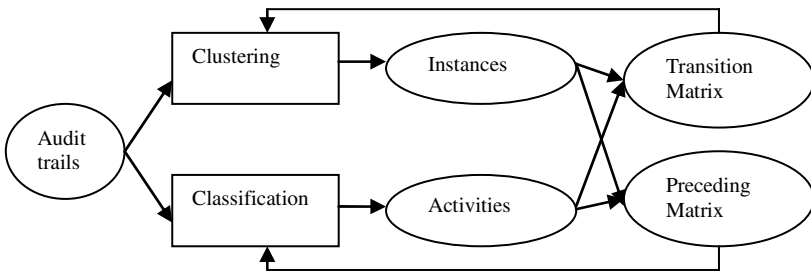


Fig. 1. Process for converting audit trails to event logs

3.1 Discovering Workflow Instances

In this section, we address the problem of discovering process instances. A process instance corresponds to a single execution of a business process. Each process instance may consist of tens or hundreds of events, depending on the granularity level of the events. Unlike in the case of topic detection and tracking, where keywords are used to determine the similarity of two events [4], in workflow mining, we take the

approach of viewing the similarity between events within a workflow instance being determined by the named entities contained in the artifacts of the events. For example, in the case of a travel application, every artifact (emails, documents, webpage forms) involved may include the name(s) of the applicant(s), the destination(s), and the dates for departure and return. Therefore, we use these named entities in the events to group events into instances. We treat the named entities as symbolic values; therefore the

similarity of the named entities can be defined as
$$sim(e_1, e_2) = \frac{1}{K} \sum_{i=1}^K \frac{|E_{1i} \cap E_{2i}|}{|E_{1i} \cup E_{2i}|}$$
,

where e_1 and e_2 are two events, K is the number of the entity types considered, E_{1i} and E_{2i} are the named entities for the two artifacts for the entity type i , which are sets of words. For example, suppose we have two documents, each of which contains two types of entities: locations and persons' names. The first document contains destinations $\{Paris, France, Toronto\}$ and the persons' names $\{John, Smith, Mary, Bergman\}$. The second one contains destinations $\{Paris, Toronto\}$ and persons' names $\{John, Mike, Mary\}$. The similarity between the two documents in terms of the named entities is $sim(d_1, d_2) = 1/2 * (2/3 + 2/5) = 0.53$.

The similarity between an event e and an instance ins , which consists of a set of events, is defined as
$$sim(e, ins) = \frac{1}{K} \sum_{i=1}^K \frac{|E_e \cap (\bigcup_{e' \in ins} E_{e'})|}{|E_e \cup (\bigcup_{e' \in ins} E_{e'})|}$$
.

Here we did not try other specialized similarity measures because specifying the similarity measure too much to improve accuracy is not our purpose. Instead, we would like to see how integration of the instance discovery and activity discovery affects each other in a more general situation.

The clustering algorithm for discovering workflow instance is presented in Figure 2. Ins refers to the set of instances to be identified. ins refers to an instance which is composed of a set of events. The algorithm processes the events in the chronological order. It first finds the instances identified up to the present that is most similar to the current

Function InstanceClustering
Rank the events in the ascending order of timestamp
$Ins = \{\}$ // Initialize the set of instances
for each event e in the log do
$sim = \max_{ins \in Ins} (sim(e, ins))$
if $sim > threshold$ //The instance for the event is identified
$ins_select = \arg \max_{ins \in Ins} (sim(e, ins))$
$ins_select = ins_select \cup \{e\}$
else //A new instance for the event is created
$ins_new = \{e\}$
$Ins = Ins \cup \{ins_new\}$
endif

Fig. 2. Clustering algorithm for workflow instance identification

event. If the similarity value between the event and the most similar instance is greater than a threshold, it assigns the event to the instance. Otherwise a new instance is created with this event as the first event in the instance.

Greater similarity threshold values result in more workflow instances to be discovered, each of which contains fewer events, while smaller threshold values result in fewer instances, each of which has more events.

3.2 Discovering the Activities

For a specific workflow, the number of the activity types is fixed. Classification algorithms can be used to train a classification model based on the keywords contained in the artifacts to classify the events into activity types. We used a Naïve Bayesian classifier to identify activities due to its efficiency and ease of incorporating new features.

According to the Naïve Bayesian classifier, given a set of keywords w_1, w_2, \dots, w_k associated with an event e , the probability of e being an activity A can be defined as

$$P(A | w_1, \dots, w_k) \propto P(A) \prod_{i=1}^k P(w_i | A).$$

In our implementation, Laplace smoothing is used to avoid zero values for $P(w_i | A)$.

We assign the event to the activity with the maximum posterior probability

$$A = \arg \max_i P(A_i | w_1, \dots, w_k).$$

3.3 Constructing the Transition Matrix and Preceding Matrix

In some cases, the named entities alone in the audit trail may not be enough to discover the workflow instances. For example, suppose there are two instances intervening together. Instance one is about John's trip to London, Ontario and Instance two is about his trip to London, UK. If he were to write an email about an expense claim for his trip, but only included his name and London as named entities, it would be difficult to say which process instance this event belongs to. However, if the current stages in the process of the two instances are known, it may help make the decision. Suppose the current stage of process instance one is *applying for trip* and that of instance two is *booking hotel*. It might be safe to say that this new event should belong to instance two because it is more likely that the reimbursement is done after booking a hotel room and/or flight. Similarly, combining the keywords and the sequence information can also help classify activities.

After identifying the workflow instances and activities in the first round as described in Sections 3.1 and 3.2, we can generate the initial structured event log. Based on the initial event log, we can construct an n times n transition matrix, where n denotes the number of activity types. The transition matrix indicates the probability that each activity is followed by each other activity in a particular process instance. Specifically, the entry of row i and column j records the probability that activity i is followed by activity j , denoted as $Follow(a_i, a_j) = P(a_j | a_i)$. We also construct an n times n preceding matrix, where each entry $Preceding(a_i, a_j) = P(a_i | a_j)$ represents the probability that activity a_j is preceded by a_i .

3.4 Using the Transition Matrix for Reclustering Workflow Instances

The reclustering algorithm is identical to the initial clustering algorithm as shown in Figure 2 except that we replace Line 5 with $sim = \max_{ins \in Ins} (Follow(a(last(ins)), a(e)) * sim(e, ins))$ and replace Line 7 with $ins_select = \arg \max_{ins \in Ins} (Follow(a(last(ins)), a(e)) * sim(e, ins))$, where $a(e)$ denotes the mapping from event e to activity type, and $last(ins)$ denotes the last event that has been grouped in the current instance ins up to present. It should be noted that in the second clustering process, the optimal similarity threshold may be different from that for the initial clustering due to the introduction of the factor *Follow*.

3.5 Reclassification for Activity Discovery

With the event log obtained from the initial clustering and classification, we can reclassify the events to new activity labels with the adjusted probability estimation $P(A | w_1, \dots, w_k, A_p) \propto proceed(A_p, A) * P(A) \prod_{i=1}^k P(w_i | A)$, where $P(w_i | A)$ and $P(A)$ are the same items as in the first classification, $A_p = a(last(inst))$ denotes the activity type of the last event grouped in the instance $inst$ up to present, and $preceding(A_p, A)$ refers to the probability that activity A is preceded by activity A_p . It can be seen that in training the second classification model, we can use the items in the initial classification models and only need to obtain the preceding matrix from the initial event log, which makes the reclassification process very efficient. This reclassification for activity types combines both the keywords of the documents and the sequence patterns of the events. Here we assume the Markov property of the sequence, i.e., the current activity is only dependent on the preceding one.

4 Experiments

We conducted extensive experiments on simulated data to evaluate the performance of our method. The experiments were implemented in Java and were performed on a Core 2 1.83GHz PC with 4GB memory, running on Windows XP.

4.1 Experiment Design

As pointed out by [8], the real email messages containing workflow are difficult to obtain due to privacy concerns, let alone the real data representing workflow which also contains other computer operations. Therefore we used simulated data for our experiments. First we investigated the common process of approving employee travels in an organization. We represented the simplified workflow model in a Petri Net as shown in Figure 3.

The simulated data sets were generated in two steps. First, structured event logs were generated from the workflow model. Then, operators, timestamps, named entities, and the keywords associated with each event were generated. Types of named entities we considered include traveler's names, destinations, and dates of departure

and return. Named entities and keywords can contain noise. We generated nine audit trail data sets with noise levels ranging from 10% to 90% with an increment of 10%. We consider three types of noise: insertion of a random word (or named entities) from a dictionary, deletion of keywords (or named entities), and replacement of keywords (or named entities) with other words in the dictionary (or other named entities of the same type). The three types of noise were added with the same probability. Each audit trail data set contains 100 instances with around 1400 events.

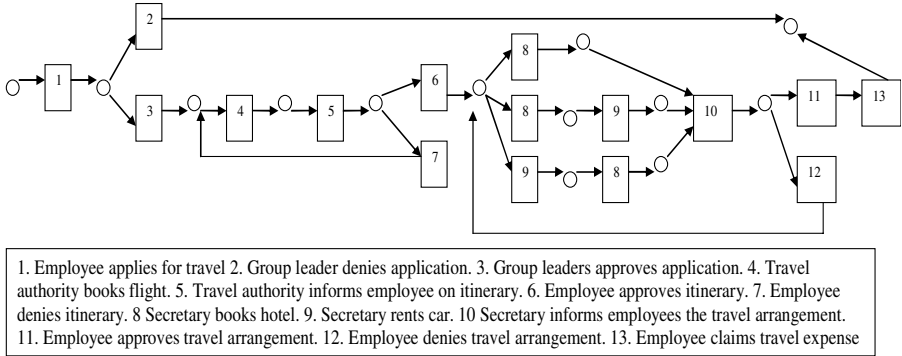


Fig. 3. Travel application workflow

We used the F measure to evaluate the instance clustering results. The F measure consists of two factors, *precision* and *recall*. In the scenario of clustering, *recall* represents how many object pairs that should be in same cluster are in the same cluster in the clustering results. The *precision* represents how many object pairs that are discovered in the same cluster are correct. The F measure is the harmonic mean of recall and precision.

We used *accuracy* to evaluate the classification of the activities. *Accuracy* is the ratio between the number of correctly classified objects to the number of all objects.

4.2 Experimental Results

Table 3 shows the F measure for the initial clustering results on the simulated data sets. Each row in the table represents a similarity threshold and each column represents a data set with different levels of noise. The best results for each data set are shown in bold face. Two observations can be obtained from the table. First, when we increase the noise level, the best derivable F measure decreases. This coincides with our intuition. Secondly, for the data set with higher level of noise, the best F measure values are obtained from smaller similarity thresholds. This is because when the noise level increases, the similarity values between events that belong to the same workflow instance decrease.

A second clustering is conducted on the best results of the initial clustering and the initial activity classification for each data set. Similarly, we vary the similarity threshold to obtain the best results for the second clustering. Figure 4 compares the results of initial clustering and second clustering. The X-axis denotes the level of noise for the data sets and the Y axis denotes the best F values obtained. If the activity labels

are obtained from the initial activity classification, which is not perfect, the second clustering results are better when the noise level is below 30%. This means that if the quality of data is reasonably good, and accordingly the results of the initial clustering and classification are reasonably good, the second clustering will improve the results from the initial clustering. Otherwise, the second clustering will deteriorate the results. By intuition if the initial results are poor, and provide false information to the second clustering, it only makes things worse. We also compared these results with the second clustering when the activity labels are perfect. It can be seen that the second clustering results based on perfect activity labels are better than those of the initial clustering when the noise level is below 50%. Another observation is that second clustering with perfect activity labels almost always obtains better results than the second clustering with imperfect activity labels obtained from initial classification.

Table 3. Initial clustering results for instance identification

Noise \ T	10%	20%	30%	40%	50%	60%	70%	80%	90%
0.1	0.407	0.473	0.507	0.546	0.549	0.577	0.600	0.513	0.480
0.2	0.721	0.874	0.942	0.907	0.746	0.579	0.513	0.509	0.514
0.3	0.913	0.910	0.719	0.583	0.561	0.485	0.315	0.168	0.060
0.4	0.970	0.672	0.500	0.403	0.274	0.150	0.067	0.026	0.013
0.5	0.774	0.501	0.339	0.221	0.103	0.037	0.014	0.008	0.010
0.6	0.606	0.359	0.198	0.093	0.026	0.006	0.003	0.001	0.002
0.7	0.482	0.232	0.087	0.023	0.004	0.000	0.001	0.001	0.001
0.8	0.351	0.106	0.020	0.002	0.000	0.000	0.000	0.000	0.000
0.9	0.182	0.024	0.002	0.000	0.000	0.000	0.000	0.000	0.000

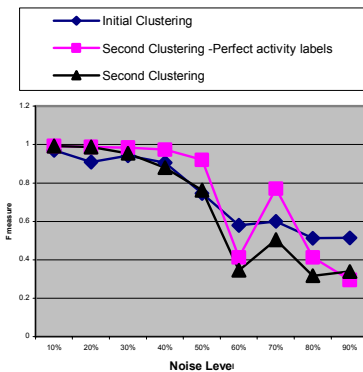


Fig. 4. Comparison of initial clustering and second clustering for instance identification

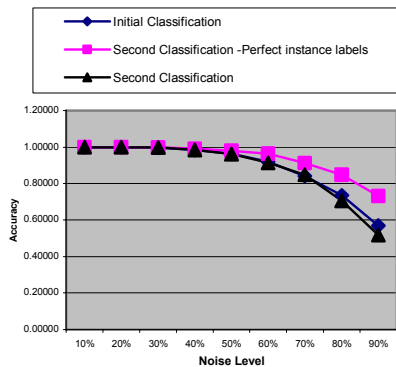


Fig. 5. Comparison of the initial and second classifications for the activity identification

Figure 5 presents the results for activity classification. It can be seen that when the noise level is below 70%, there is no significant difference between the initial classification and the second classification based on the imperfect instance labels obtained

from the clustering process. When the noise level is above 70%, the second classification obtained worse accuracy than the initial classification. This is because the preceding activity identified is inaccurate such that it provides wrong information for the second classification. We also compared the initial classification and the second classification which is based on the perfect instance labels. It shows that the information for perfect labels for instances did improve the performance for the second classification at all noise levels. The experiments show that if the clustering method obtains instances with sufficiently good results, it improves the activity classification results. Otherwise, it could deteriorate the results.

5 Conclusions

We worked on the problem of identifying instances and activities of workflows from unstructured data, and showed that integration of the two tasks has the potential to improve performance for each other, when they provide sufficiently accurate information to each other. Experimental results show that the integration of activity identification and instance identification is a double-edged sword. When the initial classification and clustering results are good enough, the second clustering and classification will obtain better results. Otherwise, performance deteriorates. Answering the question about how to define a “sufficiently good” situation is our future work.

In our experiments on simulated data, we made some assumptions and simplifications. For example, we assume that all the events in the logs are related to the target work process. In reality, most of the events recorded in the audit trail may not be related to the target work process. So filtering out these unrelated events may need to be conducted in a preprocessing step. Another simplification involves named entity detection. In our experiments, the named entities in the synthetic data sets are generated rather than detected, which makes our task easier. However, we inserted different levels of noises for named entities, which could be thought of as simulating falsely-identified named entities. Thirdly, we ignored the semantics of the terms in the simulated data set, such as the synonyms, homonyms, and equivalences between terms. In the future, we will generate more complicated simulated data to address these issues. Finally, we ultimately plan to apply this method to real data.

References

- [1] van der Aalst, W.M.P., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.M.M.: Workflow mining: A survey of issues and approaches. *Data and Knowledge Engineering* 47(2), 237–267 (2003)
- [2] van der Aalst, W.M.P., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. *IEEE Trans. Knowl. Data Eng* 16(9), 1128–1142 (2004)
- [3] Agrawal, R., Gunopulos, D., Leymann, F.: Mining process models from workflow logs. In: Schek, H.-J., Saltor, F., Ramos, I., Alonso, G. (eds.) *EDBT 1998*. LNCS, vol. 1377, pp. 469–483. Springer, Heidelberg (1998)
- [4] Allan, J. (ed.): *Topic Detection and Tracking: Event-Based Information Organization*. Kluwer Academic Publishers, Dordrecht (2002)

- [5] de Carvalho, V.R., Cohen, W.W.: On the collective classification of email “speech acts”. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005), pp. 345–352 (2005)
- [6] Cook, J.E., Wolf, A.L.: Software process validation: Quantitatively measuring the correspondence of a process to a model. *ACM Trans. Softw. Eng. Methodol.* 8(2), 147–176 (1999)
- [7] Greco, G., Guzzo, A., Manco, G., Saccà, D.: Mining frequent instances on workflows. In: Proceedings of the 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 209–221 (2003)
- [8] Khoussainov, R., Kushmerick, N.: Email Task Management: An Iterative Relational Learning Approach. In: Second Conference on Email and Anti-Spam (CEAS), Stanford University, California, USA (2005)
- [9] Kushmerick, N., Lau, T.A.: Automated email activity management: An unsupervised learning approach. In: Proceedings of the 2005 International Conference on Intelligent User Interfaces, pp. 67–74 (2005)
- [10] Silva, R., Zhang, J., Shanahan, J.G.: Probabilistic workflow mining. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 275–284 (2005)

GIS-FLSolution: A Spatial Analysis Platform for Static and Transportation Facility Location Allocation Problem

Wei Gu¹, Xin Wang¹, and Liqiang Geng²

¹ Department of Geomatics Engineering, University of Calgary, Calgary, AB Canada

² NRC Institute for Information Technology, Fredericton, NB Canada

{wgu,xcwang}@ucalgary.ca, liqiang.geng@nrc-cnrc.gc.ca

Abstract. Static and transportation facility location allocation problem is a new problem in facility location research. It aims to find out optimal locations of static and transportation facilities to serve an objective area with minimum costs. The problem is challenging because two types of facilities are involved and locations of transportation facilities are dependent on locations of static facilities and demand objects. This paper proposes a new stand-alone GIS platform, GIS-FLSolution, to solve the problem. Combined with a customized algorithm called STFLS, the platform is built on MapObjects and can successfully provide results with a friendly graphical user interface. Preliminary experiments have been conducted to demonstrate the efficiency and practicality of the platform.

Keywords: Facility location problem, Static facility, Transportation facility, Geographical Information Systems (GIS).

1 Introduction

Facility location problem is an important research topic in spatial data analysis which solves problems of matching the supply and demand by using sets of objectives and constraints [1]. The objective is to determine a set of locations for the supply so as to minimize the total supply and assignment cost. For instance, city planners may have a question about how to allocate facilities, such as hospitals and fire stations for new residence area. The decision will be made based on the local populations and the capability of the limited resources. For another example, a local school board may provide accessible service to a population while minimizing the total distance travelled by the students. These examples are involved with single type of facilities such as hospitals, fire stations or schools. Various methods for the single type of facility location problem have been proposed for the above applications [1, 2, 3, 4, 5].

In reality, we often face two types of facilities location problem when the number of the single type of facilities within a service area is inefficient. For example, for emergency medical services, we can locate the hospital locations in such a way that it achieves full coverage of a service with the minimum total travelling distance. This usually ends up with the hospital locations being close to the dense community.

However, for the residents in the sparse and remote area, since the number of hospitals is limited, in order to offer fast response, the ambulance should be located to shorten the time to access medical services. In this application, two types of facilities need to be located in a region, static facilities (e.g. hospitals) and transportation facilities (e.g. ambulances). The service is supplied to the customers by the cooperation of these two types of facilities. In addition, dependency relations usually exist between different types of facilities. Specifically, the locations of transportation facilities are dependent on the locations of static facilities and demand objects. For example, the locations of ambulances will be determined by both residence locations and hospital locations. However, none of the current algorithms can be applied to the two types of facilities location problem directly.

During the past thirty years, Geographical Information Systems (GIS) has evolved into a mature research and application area involving a number of academic areas including Geography, Civil Engineering, Computer Science, Land Use Planning and Environmental Science [6]. By supporting a wide range of spatial queries and analyses, GIS is playing a significant role in location model development and application. GISs have been developed in use in municipalities, states, utilities, and governmental agencies, which range from simple, limited systems to large, complex software systems. However, none of them include spatial analysis function which can be used for the two types of facilities location problem directly.

In this paper, we propose a novel heuristic algorithm to solve the two types of facilities location problem. Instead of only minimizing the average travelling distance between the static facilities and demand objects, the new algorithm also considers the constraint between the transportation facilities and static facilities. Based on the new algorithm, we also present a simple GIS platform, called GIS-FLSolution, which was designed for geographers and civil engineers to facilitate their applications.

The rest of the paper is organized as follows. Section 2 summarizes related work on the location allocation problem and GIS. In section 3, the GIS-FLSolution platform is presented. We first give a formal definition of the static and transportation facility location problem. Then, a customized algorithm named STFLS is introduced to solve the problem. Finally, the structure of the platform is proposed. Experimental studies on synthetic and real datasets are presented in Section 4. Finally, Section 5 concludes the paper with future research directions.

2 Related Work

Facility location decisions are critical element in strategic planning for a wide range of private and public firms [7]. Three common objectives on the single level facility location problem have been studied:

The *P*-Median Problem—seeking to locate a given *P* number of facilities that minimizes the total distance travelled from all demand points to their nearest serving facilities. When applied to a general network, the *P*-median problem is hard to be solved optimally (this type of problems is NP-hard). However, by limiting potential facility locations on the network nodes, the *P*-median problem can be solved in polynomial time. Several integer programming techniques and efficient heuristics have been developed for solving this problem [8]. Recently, Zhang et al. [9] proposed the

min-dist optimal-location query in spatial database to find the optimal location for one single facility. They proved a theorem that the number of candidate locations is finite and a min-dist optimal location exists at some intersection points of horizontal and vertical lines passing through demand objects. The theorem limits the number of candidate locations without losing the power to find the exact answers. However, this method can only allocate one facility using the Manhattan distance.

The Center Problem—locating a given number of facilities that minimizes the maximum distance while asking for coverage of all demand points. If facility locations are restricted to the nodes of the network, the problem is a *vertex center problem*. Center problems which allow facilities to be located anywhere on the network are *absolute center problems* [8].

The Covering Problem— finding the locations of a fixed number of facilities that maximizes the total demand covered by them within a maximum acceptable distance. In [7], Pacheco et al. proposed a method to solve the problem under probabilistic condition. They adapted three metaheuristic strategies—scatter search, tabu search, and variable neighborhood search—to find the best locations in Spain’s Burgos province to place health resources for treating people in diabetic comas. Similar to the P -median problem above, the covering problem is NP-complete for general networks. Covering problem model is widely used to determining the deployment of Emergency Medical Service System (EMS) vehicles in environments [10, 11].

While the earlier approaches consider that each static facility candidate can satisfy an unlimited number of demand points, recent approaches tend to give each candidate facility limited capacity, i.e. a maximum number of demand points that it can satisfy. To identify the assignment with the optimal overall quality, Ghoseiri and Ghannadpour [12] proposed an urgencies capability constraint facility allocation method that gives an assignment between a static facility and a demand point through urgencies. An *urgency* is a method to define a precedence relationship between points, which can also be viewed as a priority.

GIS is an ideal and sometimes indispensable tool for analyzing location allocation problem. The central element of a GIS is the use of a location referencing system so that data about a specific location can be analyzed in its relationship to other locations. Church [6] provided a review of GIS and location modeling and pointed out there is an inextricable link between GIS and Location Science. Gerrard et al. [13] presented an application using the ARC/INFO [14] location model capabilities applied to biological reserve site selection. Ruggles and Church [15] have used a GIS to generate and maintain many possible scenarios for an application of a location model to historical settlements in the Basin of Mexico. However, none of these GIS platforms can handle static and transportation facility location allocation problem at the same time.

3 Static and Transportation Facility Location Platform

In this section, we present a new GIS platform called GIS-FLSolution to solve static and transportation facility location problem. In Section 3.1, we give the formal definitions of the problem. In Section 3.2, the searching algorithm used in the platform is discussed. Section 3.3 describes the implementation of GIS-FLSolution.

3.1 Static and Transportation Facility Location Problem

Two types of distance used in this paper are defined as follow.

Definition 1. Given a set of demand objects D and a set of static facilities S , the *average travelling distance (ATD)* is defined as:

$$ATD = \frac{\sum_{d_j \in D} dist(d_j, s_i) * d_j.w}{\sum_{d_j \in D} d_j.w}, \text{ where } s_i \in S, d_j \in D \text{ and } s_i \text{ is } d_j\text{'s assigned static}$$

facility. $d_j.w$ is a positive number representing the demand of the demand object d_j .

Definition 2. Given a set of demand objects D , a set of static facilities S and a set of transportation facilities T , the *transportation travelling distance (TTD)* of the demand object d_j is defined as:

$TTD(d_j) = dist(d_j, s_i \parallel t_k) + dist(d_j, s_i)$, where $s_i \in S$, $d_j \in D$, $t_k \in T$ and $dist(d_j, s_i \parallel t_k)$ is the distance from a location of a demand object d_j to its assigned static facility location s_i or the closest transportation facility location t_k , whichever is shorter.

Static and Transportation Facility Location (STFL) Problem is to determine locations for a set of static facilities S and a set of transportation facilities T , which satisfies the following conditions:

- (1) Minimize the value of the average weighted traveling distance function ATD
- (2) Minimize $Max\{TTD(d_j), d_j \in D\}$

In the definition, the condition (1) minimizes the average weighted travelling distance from every demand object to its assigned static facility. The condition (2) stipulates to minimize the maximum value of traveling distance for each transportation facilities. Since locations of static facilities are more important to most of demand objects, the condition (1) has higher priority.

3.2 STFLS: A Heuristic Method for STFL Problem

Static and transportation optimal facility locations problem is a NP-hard problem. In this subsection, we propose a new heuristic method called Static Transportation Facilities Location Searching Algorithm (STFLS). The algorithm contains two steps: static facility location searching and transportation facility location searching.

3.2.1 Static Facility Location Searching

In this step, we propose a heuristic method to find local optimal locations using clustering. Clustering is the process of grouping a set of objects into classes so that objects within a cluster have high similarity to one another, but are dissimilar to objects in other clusters [17]. The clustering process is used to reduce the searching area.

SearchStaticFacilityLocations (D,S)

Input: a set of demand points D , a set of static facilities S with unknown locations.

Output: locations of S .

Initialize: hasBetterStaticLocation=True, miniDist=0
averageTralDist=0

```

1  randomly choose the initial locations for static
   facilities in S
2  UrgencyCapabilityConstraintAssignment (D,S)
   /*calculate the average travelling distance for current
   arrangement of static facilities.*/
3  averageTralDist = calculateAverageTralDist(D,S)
4  while(hasBetterStaticLocation) //static facilities
   exchanging
   /* Intra-cluster searching */
5  { for all s∈S do
6      { newIntraPos=findIntraCluOptimalLoc(s,D,S)
7          newAverageDist=calAveDist(newIntraPos,D,S)
8          insert (newIntraPos, tempPosList)
9          insert (newAverageDist, tempDistList) }
   /* Inter-cluster searching. Find the smallest value in
   tempDistList and its corresponding position in
   tempPosList. Then record this distance as minDist, the
   position as newPos, the original location of static
   facility in the cluster as oldPos. */
10  newPos, oldPos, minDist ←
    findInterclusterOptimalLoc(tempDistList, tempPosList)
11  exchange(newPos, oldPos)
   /*reassign the demand points to static facilities.*/
12  urgencyCapabilityConstraintAssignment(D,S)
13  if(minDist < averageTralDist)
14      averageTralDist = minDist
15  else /* this step terminates if the last local
   altering did not result in a change */
16      hasBetterStaticLocation = False }

```

Fig. 1. Pseudo code of static facility location searching

Fig. 1 presents the pseudo-code of static facility location searching. The procedure involves three steps:

Step 1: Initialization. First, the locations of static facilities are picked randomly in line 1. Secondly, every demand object is assigned a static facility by using urgency capability constraint assignment method [12] in line 2. After the assignment, each static facility together with its assigned demand objects is considered as a cluster.

Step 2: Intra-Cluster Searching. The goal of the step is to find the local optimal location to minimize the average travelling distance from demand points to the static

facility in each cluster. Because the objects in a cluster are closer to each other than the objects from other clusters, for every static facility in a cluster, we assume that its optimal location should be in that cluster. Through separating the whole area into different clusters, the searching space for every static facility is reduced from the whole area to its cluster. First, `findIntraCluOptimalLoc` finds the local optimal location for each static facility from its cluster. The local optimal location is saved in `newIntraPos` and its corresponding minimum average travelling distance is saved in `newAverageDist` in lines 6-7. Secondly, the method inserts `newIntraPos` into `tempPosList` which stores the local optimal location in each cluster and inserts `newAverageDist` into `tempDistList` which stores the minimum average travelling distance in lines 8-9 for the inter-cluster searching.

Step 3: Inter-Cluster Searching. The goal of the step is to compare the local optimal static facilities' location in every cluster and select one which can reduce the average distance most as shown in lines 10-11. Because each cluster is defined as the locations of a static facility and its assigned demand objects, after changing the static facility's location, the clusters need to be rebuilt by assigning all demand points to new static facilities in line 12. The reason we only change one static facility to its new local optimal location is that all the local optimal locations for static facilities are determined under the same distribution of static facilities in step 2. Thus, once one static facility's location is changed, the other static facilities' local optimal locations could be changed.

Step 2 and step 3 are iterated until no change happened in step 3.

3.2.2 Transportation Facilities Location Searching

Locations of transportation facilities depend on both locations of demand objects and static facilities. To reduce the computation time, we use a greedy method in this step. The method includes 3 steps. Fig. 2 presents the pseudo-code of transportation facilities location searching.

Step 1: Initialization. Randomly choose the initial locations for all transportation facilities in line 1.

Step 2: Exchanging. The strategy is that it changes every transportation facility to the location whichever reduces the maximum transportation travelling distance most within each iteration. First, `findOptimalTransportationLoc` finds the optimal location for each transportation facility. The optimal location is stored in `tempPos` as shown in line 4. Then each transportation facility is changed to a location which reduces the maximum transportation travelling distance most in line 5 and the corresponding minimum transportation travelling distance is saved in `tempDist` in line 6.

Step 3: Calculation. Calculate the transportation facility travelling distance of every demand point and determine every transportation facility's service list of demand points under current transportation facility locations in line 6.

Step 2 and step 3 are iterated until no change is made in step 2 or the iteration times reach the predefined threshold.

SearchTransportationFacilitiesLocations(D,S,T,threshold)

Input: a set of transportation facilities T, a set of demand points D, threshold is a positive number the user input which presents the maximum iteration times.

Output: locations of T

Initialize: hasBetterTransportationLocation=True,
miniDist= ∞ , count=0;

```

1  randomly choose the initial locations for
   transportation facilities in T.
2  while((hasBetterTransportationLocation)and
       (Count < threshold))
3    { for all t $\in$ T do
4      { tempPos=findOptimalTransportationLoc(t,T,D)
5        exchange(tempPos, t)
6        tempDist=calculateTransportTravelDist(T) }
7    count = count + 1
8    if(tempDist > miniDist)
9      HasBetterTransportationLocation = False
10   else
11     miniDist = tempDist }

```

Fig. 2. Pseudo code of transportation facility location searching

3.3 GIS-FLSolution

We implemented GIS-FLSolution by using Java and MapObjects-Java standard edition [16]. MapObjects is an easy-to-use, pure Java application programming interface (API) that enables programmers to develop custom Java clients or implement stand-alone GIS mapping solutions. Through calling a set of Java mapping and non-visual components in MapObjects, GIS-FLSolution has the ability to read diverse data sources (dbf file, xls file and csv file with spatial attributes and non-spatial attributes) and to map properties in an appropriate scale. Combined with a customized facility location allocation algorithm, GIS-FLSolution can successfully solve the STFL problem.

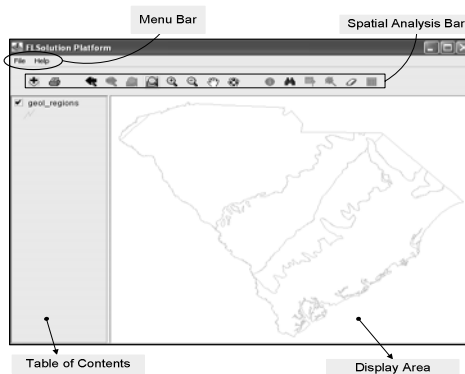


Fig. 3. Graphical user interface of GIS-FLSolution

Fig. 3 shows the GIS-FLSolution graphical user interface. The top left corner of the interface contains *Menu Bar* which includes *File* button and *help* button. By selecting *File*, users are allowed to upload their datasets, print out maps and exit the platform. In *Help*, a help file is provided to explain each functions of the platform. *Spatial Analysis Bar* contains several simple spatial analysis tools, such as *Zoom In*, *Zoom Out*, *Query Builder*, *Select Features* and *Attributes*, which allow users to select specific features and retrieve data records from spatial database. The bottom right corner of the interface is a *Display Area* showing visualized results. To the left of the *Display Area* is *Table of Contents* displaying the corresponding contents shown in the *Display Area*.

4 Experiments

4.1 Comparison between GIS-FLSolution and Optimal Solution

Synthetic datasets for demand objects were created in a 300 × 300 area. The values in the following experiments are the average of the results which are from running the algorithm six times. To compare the performance of GIS-FLSolution with the optimal solution, we generate a dataset with 100 demand objects and locate 3 static facilities and 2 transportation facilities. The weight of each demand object is 30 and the capability of each static facility is 1000. Table 1 compares the results and shows that the optimal solution has a better performance on average and maximum travelling distance than GIS-FLSolution but it is more time consuming.








Table 1. The comparison between GIS-FLSolution and the optimal solution

	GIS-FLSolution	Optimal solution
Average traveling distance	48.7	46.5
Maximum traveling distance	199.0	192.3
Execution time (s)	1	200

4.2 Experiment with Real Data

This section presents a sufficient-capacity experiment with a real data set from South Carolina, which is to locate five hospitals and three ambulances. The data set consists of 867 census tracts (Census2000), with each treated as a demand object. The population of each census tract is considered as the demand weight of each object, which varies from 197 to 16,745. The total population (or demand) is 4,212,012. Centroid of each census tract is used as the locations of the corresponding demand objects. Capabilities of the hospitals range from 800,000 to 1,400,000.

Using the platform to solve the problem should follow the steps below: First, upload a dataset to the platform by selecting *File* → *Add Layer*. Second, set up parameters such as the number of static facilities and transpiration facilities. Then the platform calls the customized algorithm to find the most suitable locations of static facilities and transportation facilities. Finally, the results are mapped to the user interface.

Fig. 4 presents the result of using GIS-FLSolution.  denotes the location of a hospital.  represents the location of ambulance. The points marked by ,  and  stand for different ambulance's service area. In Fig. 4, two ambulances are assigned close to each other but serving different areas marked by  and . Since the function of ambulances is to minimize the maximum transportation travelling distance, the ambulances are not assigned to the centroids of their service area. In this example, the average travelling distance is 56.2 km, the maximum transportation travelling distance is 260 km and the execution time is 12 seconds.

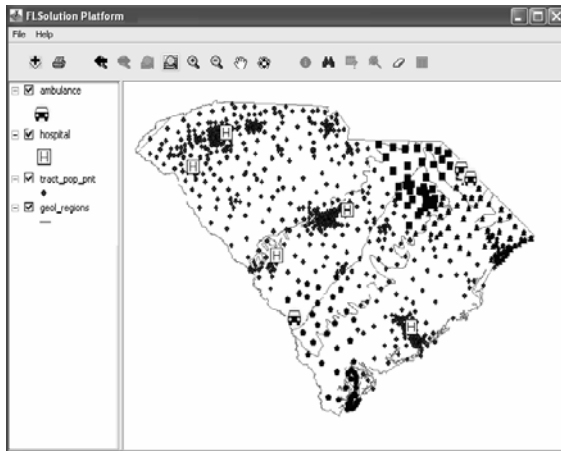


Fig. 4. Solution to South Carolina dataset using GIS-FLSolution

5 Conclusion and Future Work

In this paper, we introduce a new type of facility location problem with both static and transportation facilities and propose a novel heuristic algorithm STFLS to solve it. STFLS assigns two types of facilities to the local optimal locations based on the spatial distribution of demand objects and the dependency of the facilities. Based on STFLS, a stand-alone GIS platform, GIS-FLSolution is presented. The platform gives a friendly user interface. To our knowledge, GIS-FLSolution is the first platform to handle the location allocation problem about two types of facilities. According to the experimental results, GIS-FLSolution can accurately allocate two types of facilities into an area.

In the future, we will extend the problem and platform to handle multiple types of facilities. In addition, we will introduce a pre-processing method and spatial data index in STFLS to reduce the execution time. Finally, we will implement a new version of GIS-FLSolution using ArcEngine which supports dynamic mapping and information releasing in web environments.

References

1. Owen, S.H., Daskin, M.S.: Strategic facility location: A review. *European Journal of Operational Research* 111(3), 423–447 (1998)
2. Longley, P., Batty, M.: *Advanced Spatial Analysis: The CASA Book of GIS*. ESRI (2003)
3. Arya, V., Garg, N., Khandekar, R., Pandit, V., Meyerson, A., Mungala, K.: Local search heuristics for k-median and facility location problems. In: *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing*, pp. 21–29 (2001)
4. Jain, K., Mahdian, M., Saberi, A.: A new greedy approach for facility location Problems. In: *Proceedings of STOC* (2002)
5. Charikar, M., Khuller, S., Mount, D.M., Narasimhan, G.: Algorithms for facility location problems with outliers. In: *Proceedings of SODA* (2001)
6. Church, R.L.: Geographical information systems and location science. *Computers and Operations Research* 29(6), 541–562 (2002)
7. Pacheco, J., Casado, S., Alegre, J.F.: Heuristic Solutions for Locating Health Resources. *IEEE Intelligent Systems* 23(1), 57–63 (2008)
8. Daskin, M.S.: *Network and Discrete Location: Models Algorithms and Applications*. Wiley, Chichester (1995)
9. Zhang, D., Du, Y., Xia, T., Tao, Y.: Progressive Computation of The Min-Dist Optimal-Location Query. In: *Proceedings of International Conference on Very Large Data Bases (VLDB)*, pp. 643–654 (2006)
10. Daskin, M.S.: Application of an Expected Covering Model to Emergency Medical Service System Design. *Decision Sciences* 13(3), 416–439 (1982)
11. Jia, H., Ordonez, F., Dessouky, M.: A modeling framework for facility location of medical service for large-scale emergencies. *IIE Transactions* 39(1), 41–55 (2007)
12. Ghoseiri, K., Ghannadpour, S.F.: Solving Capacitated P-Median Problem using Genetic Algorithm. In: *Proceedings of International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 885–889 (2007)
13. Gerrard, R.A., Stoms, D.A., Church, R.L., Davis, F.W.: Using GIS models for reserve site selection. *Transactions in GIS* 1(2), 45–60 (1996)
14. ArcInfo website,
<http://www.esri.com/software/arcgis/arcinfo/index.html>
15. Ruggles, A., Church, R.L.: An analysis of late-horizon settlement patterns in the Teotihuacan-Temascalapa basins, a location-allocation and GIS approach. In: Aldenderfer, M.S., Maschner, H.D.G. (eds.) *Anthropology, space and geographic information systems*. Oxford University Press, Oxford (1997)
16. MapObjects-Java standard edition,
<http://www.esriuk.com/products/product.asp?prodid=46&>
17. Han, J., Kamber, M., Tung, A.K.H.: Spatial Clustering Methods in Data Mining: A Survey. In: Miller, H., Han, J. (eds.) *Geographic Data Mining and Knowledge Discovery*. Taylor and Francis, Abington (2001)

A CBR System for Knowing the Relationship between Flexibility and Operations Strategy

Daniel Arias-Aranda¹, Juan L. Castro², Maria Navarro², and José M. Zurita²

¹ Department of Business Management, Faculty of Business Management and Economics, Granada University, Spain

² Department of Computer Science and Artificial Intelligence, ETSI Informática, Granada University, Spain

Abstract. Changing environments are driving firms towards the development of new techniques for the decision making process in order to fit rapidly with alterations and adjustments of the market. In this context, the relationship between operations strategy and flexibility plays a fundamental role for increasing performance goals. For this reason, this paper presents a Fuzzy Probabilistic Case-based reasoning (FP-CBR) system which studies the relationship between flexibility and operations strategy in a real sample of engineering consulting firms in Spain. The objective is to develop a framework of analysis based on CBR and fuzzy logic whose accuracy is measured in order to assess scientific evidence to the conclusions. In order to help manager to make decisions about the firms.

1 Introduction

Operations strategy has been receiving increasing attention in literature especially due to the fast changes in the firm environments. Operations strategy has been studied in literature according to its different types and dimensions. The categories of decisions identified by literature [20] regarding the operations strategy are layout, push vs. pull orientation of the service delivery process, degree of standardization of the process, number of different services offered, information technology focus (cost saving vs. service upgrading), relation between back and front office activities, human resource management, degree of customer participation and new services design and development [2]. However, operations flexibility emerges as competitive priority that most firms need to confront to when adjusting to environmental variations. Different dimensions and flexibility types have been identified in the literature [12]. The general dimensions are expansion, distribution of information, routing, labour and equipment, market flexibility, services and servuction, process, programming and volume. Operations strategy dimensions and flexibility constructs configure the basis of our analysis in the present study.

The relationship between operations strategy and flexibility has become the object of several studies [18,11] which focus on different competitive priorities as moderators of this link. The understanding of the best practices to fit operations strategy and flexibility is crucial to decrease uncertainty in the decision making

process of the firms [16]. These kind of relationships have been analyzed through the use of different approaches from statistical analysis, as we can see in [3] to Artificial Intelligence (AI) techniques like expert systems [15,19], CBR [13,14], genetic algorithms [1] and so on.

The aim of this research is to study the relationship between operations strategy and flexibility with current techniques of AI. These techniques allow us to combine case-base information with other relevant aspects of the problem in order to make the best decision. We develop a CBR system, called Fuzzy Probabilistic CBR system (FP-CBR). It uses the information obtained by the probability of one successfully used solution conditioned to the attributes values and it also takes advantage of the flexibility that fuzzy logic contributes to the problem, in order to extract useful knowledge of the database. This techniques were successfully used in previous works as we can see in [5,6,7,8]. We have chosen this kind of systems because case-based reasoning is quite simple to implement in general, but it often handles complex and unstructured decision making problems very effectively. Moreover, its prediction model is maintained in an up-to-date state because the case-base is revised in real time, which is a very important feature for the real world application.

The rest of this paper is organized in four sections. Section 2 describes the information extraction technique FP-CBR in detail. Section 3 describes the case study. Section 4 presents the experimental results and the final discussion. The conclusions of our research study are given in Section 5.

2 System Description

The FP-CBR system is based on fuzzy logic (like [9,10]) and probability theory (as the following CBR-system [17]) to enhance its performance and increment the model accuracy. Along the process, the initial probabilities of each solution will be updated with the information generated by every attribute concrete value. Detailed explanations for each step are presented as follows.

Step 1. Initial Probability

Each new case is regarded as having an initial probability, associated with each possible solution S . This initial probability is calculated by using case-base information and definition 1.

Definition 1. *Let S be one of the solutions stored in memory. The Initial Probability of solution S , $P_0(S)$, is defined as*

$$P_0(S) = \frac{\text{Number of cases in memory, whose solution is } S}{\text{Number of cases in memory}}$$

Step 2. Conditional Probability

Bayes' Theorem [4], conditional probabilities and initial probability are used to calculate the probability of a solution when we know the new case, namely,

$P(S/New Case)$. Let us suppose that there are m different solutions S_1, \dots, S_m in the case base and that each case contains n attributes A_1, \dots, A_n . A new case arrives, $New Case = (A_1 = a_1, \dots, A_n = a_n)$, where a_1, \dots, a_n are the specific values of each attribute of the $New Case$. If we are interested in the i th solution (S_i), the process can be analyzed as described in the next paragraphs.

Definition (1) is used to calculate the initial probability $P_0(S_i)$. This probability is then updated until $P(S_i/New Case)$ is obtained. For this purpose, the attributes are introduced one at a time, and their respective probabilities calculated. If the probability result is approximately zero this attribute is automatically removed of the probability, we use this simplification because in this work we interpret the probability as an information about the problem, for this reason if the probability is zero the attribute does not contribute with any information and we calculate the following probability without taking this attribute into account. Let us see the process in detail.

We start with attribute 1, we calculate $P_1(S_i) = P(S_i/A_1 = a_1)$. When we know the probability result, we check if it is approximately zero. If the result is different from zero we introduce the next attribute and calculate $P_2(S_i) = P(S_i/A_1 = a_1 \cap A_2 = a_2)$. Otherwise, we remove the first attribute in the probability and we calculate the following one $P_2(S_i) = P(S_i/A_2 = a_2)$ without taking the first attribute into account, and so on, up to the n th attribute, $P_n(S_i) = P(S_i/A_1 = a_1 \cap \dots \cap A_n = a_n) = P_n(S_i/New Case)$. P_j will be

$$\begin{aligned} P_j(S_i) &= P(S_i/A_j = a_j \cap \dots \cap A_1 = a_1) = \\ &= \frac{P(A_j = a_j/S_i) \cdot P_{j-1}(S_i)}{P(A_j = a_j/S_1) \cdot P_{j-1}(S_1) + \dots + P(A_j = a_j/S_m) \cdot P_{j-1}(S_m)} = \\ &= \frac{P(A_j = a_j/S_i) \cdot P_0(S_i/A_{j-1} = a_{j-1} \cap \dots \cap A_1 = a_1)}{P(A_j = a_j/S_1) \cdot P(S_1/X) + \dots + P(A_j = a_j/S_m) \cdot P(S_m/X)} \end{aligned}$$

where $X = (A_{j-1} = a_{j-1} \cap \dots \cap A_1 = a_1)$ and where each $P(A_j = a_j/S_i)$ is calculated as follows

$$P(A_j = a_j/S_i) = \frac{\text{Number of cases in the case base whose attribute } A_j \text{ verifies D}}{\text{Number of cases in the case base whose associated solution is } S_i}$$

where $D = \{A_j = \nu \mid |\nu - a_j| \leq \alpha, \alpha \in \mathbb{R}\}$.

Step 3. Information Index

Now, we save in a vector the number of times that the conditioned probability to that attribute is not approximately zero. We call this new information about the attributes, *Information Index (R)*, where each component vector will be the *Information Index* of the i th attribute (R_i).

Step 4. Calculation of the Local Similarity

Once the *Information Index* has been calculated (Step 3), we calculate the local similarity. In this work the following local similarity measurement [1] is used.

$$sim(x_i^{Mem}, x_i^{New}) = 1 - \frac{|x_i^{Mem} - x_i^{New}|}{x_i^{Max} - x_i^{min}} \tag{1}$$

where x_i^{Mem} is the i th attribute of the case in memory, x_i^{New} is the i th attribute of the current case and, x_i^{Max} , x_i^{min} , are the maximum and minimum values between all the cases (including the target case) for the attributes, respectively.

Step 5. Assignment of Fuzzy Labels

We assign fuzzy labels to the local similarity and the information index of each attribute R_i . In our model we have used the following membership functions (Fig. 1), is not a restriction to use any other that fits the data.

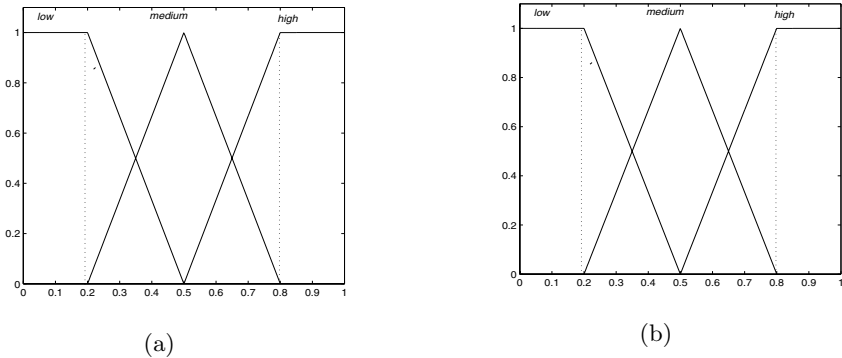


Fig. 1. (a) Local similarity (b) Information index

Step 6. Modifying the Local Similarity

In this step, we will see how to get that the *Information index* takes part inside the problem. It will be introduced into the similarity measure, using a fuzzy inference system. In order to build this system, we shall modify the formula of the overall similarity making it dependent on the Information index of each attribute (R_i) and local similarity ($sim(x_i^{Mem}, x_i^{New})$)

$$sim^{R_i}(x_i^{Mem}, x_i^{New}) = f(R_i, sim(x_i^{Mem}, x_i^{New})) \tag{2}$$

The function $f(\cdot)$ is implicitly obtained through a fuzzy inference system. This fuzzy inference system for the i th attribute contains the following 9 rules:

Rule 1. If R_i is *high* and $sim(x_i^{Mem}, x_i^{New})$ is *high*, then
 $v_i = sim(x_i^{Mem}, x_i^{New}) + 6 \cdot sim(x_i^{Mem}, x_i^{New})$.

Rule 2. If R_i is *high* and $sim(x_i^{Mem}, x_i^{New})$ is *medium*, then
 $v_i = sim(x_i^{Mem}, x_i^{New}) + 5 \cdot sim(x_i^{Mem}, x_i^{New})$.

Rule 3. If R_i is *high* and $sim(x_i^{Mem}, x_i^{New})$ is *low*, then
 $v_i = sim(x_i^{Mem}, x_i^{New}) + 2 \cdot sim(x_i^{Mem}, x_i^{New})$.

Rule 4. If R_i is *medium* and $sim(x_i^{Mem}, x_i^{New})$ is *high*, then
 $v_i = sim(x_i^{Mem}, x_i^{New}) - 0.2 \cdot sim(x_i^{Mem}, x_i^{New})$.

Rule 5. If R_i is *medium* and $sim(x_i^{Mem}, x_i^{New})$ is *medium*, then
 $v_i = sim(x_i^{Mem}, x_i^{New}) - 0.3 \cdot sim(x_i^{Mem}, x_i^{New})$.

Rule 6. If R_i is *medium* and $sim(x_i^{Mem}, x_i^{New})$ is *low*, then
 $v_i = sim(x_i^{Mem}, x_i^{New}) - 0.4 \cdot sim(x_i^{Mem}, x_i^{New})$.

Rule 7. If R_i is *low* and $sim(x_i^{Mem}, x_i^{New})$ is *high*, then
 $v_i = sim(x_i^{Mem}, x_i^{New}) - 0.3 \cdot sim(x_i^{Mem}, x_i^{New})$.

Rule 8. If R_i is *low* and $sim(x_i^{Mem}, x_i^{New})$ is *medium*, then
 $v_i = sim(x_i^{Mem}, x_i^{New}) - 0.4 \cdot sim(x_i^{Mem}, x_i^{New})$.

Rule 9. If R_i is *low* and $sim(x_i^{Mem}, x_i^{New})$ is *low*, then
 $v_i = sim(x_i^{Mem}, x_i^{New}) - 0.5 \cdot sim(x_i^{Mem}, x_i^{New})$.

Step 7. Inference System

The calculation of $sim^{R_i}(x_i^{Mem}, x_i^{New})$ is inferred in two steps. First, we calculate the firing strength g_j $j = 1, \dots, k$, where k is the number of fired rules. For instance, the firing strength of the first rule is obtained as follows:

$$g_1 = \mu_{high}(R_1) \cdot \mu_{high}(sim(x_1^{Mem}, x_1^{New})) \tag{3}$$

For all the rules, the firing strength is calculated using Equation 3. After, we calculate the $f(.)$ function

$$f(R_i, sim(x_i^{Mem}, x_i^{New})) = \frac{\sum_{j=1}^{k_i^i} v_j^i \cdot g_j^i}{\sum_{j=1}^{k_i^i} g_j^i} \tag{4}$$

where k_i^i is the number of fired rules for the i th attribute, v_j^i is the output of the j th rule fired for the i th attribute and g_j^i is the firing strength of the j th rule for the i th attribute.

3 Case Study

3.1 The Data

The expert gave us collected data about the operations strategy, level of flexibility and performance from a sample of 71 engineering consulting firms in Spain.

A questionnaire was the technique used to obtain data for the study. The questionnaire has 122 questions in 3 main groups divided in blocks of questions each one. The first one related to operations management divided in 9 blocks. A second block formed by the flexibility questions divided in 7 blocks. And the third group for the results in 2 blocks. See the complete questionnaire in *Questionnaire raised to companies involved in the study (English)* (<http://ic2.ugr.es/~jmsa/>).

Tables 1 and 2 shows the variables used in the study and the items of the questionnaire with respect to. Under the column *Variable* we have the variables names and the column *Block* indicates the questionnaire blocks where these variables appear.

Table 1. Operations Strategy

Variable	Block	Variable	Block
Layout	AI	Back and Front office activities	AVI
Push/Pull Orientation	AII	Human resource management	AVII
Standardization	AIII	Customer participation	AVIII
Different services offered	AIV	Design and development of new services	AIX
Use of information technologies	AV		

Table 2. Flexibility

Variable	Block	Variable	Block
Expansion	B1-B6	Market	B17-B18
Distribution of information	B7-B9	Services and products	B19-B20
Routing	B10-B12	Process, programming and volume	B21-B24
Equipment	B13-B16		

3.2 Model Construction, Performance Evaluation Criterion and Implementation

The model is tested making k -fold cross-validation. In each k -fold cross-validation the entire data set is divided into k mutually exclusive subsets with the same distribution. Each fold is used once to test the performance of the classifier that is generated from the combined data of the remaining $k - 1$ folds. Ten complete validations are realized, in order to have enough results at the moment of confirming the results with t -test. In this study, 5-fold cross-validation was applied. We evaluate the classification performances of models by accuracy, because our goal is to improve the accuracy in order to have enough evidence to support justification of the hypotheses obtained.

We implement FP-CBR by MATLAB 7.1. To contrast the efficiency of our method we have compared it with the following models: J4.8 which is the commercial version of well-known C5.0, Nave Bayes, Logistic (we have used the available implementations in Weka [21]), and standard CBR also implemented by us in MATLAB 7.1.

4 Results of the Study and Discussion

The goal of this paper is to develop a rather wide empirical study, so in this section we apply the specific CBR tool (FP-CBR) and we show the obtain results in order to analyze the relationship between flexibility and operations strategy in the service industry of Engineering consulting firms.

4.1 Experimental Results

In this subsection we show the results of the case study. Table 3 shows the operations strategy dimensions which exceeded the Information index. Table 4 shows a summary of the results obtained in accuracy for each flexibility dimensions.

Table 3. Selected set of attributes(operations strategy dimensions)

CV	FB1	FB2	FB3	FB4	FB5	FB6	FB7
1	AI,AIV AIX	AI,AIX	AII,AIII AIV,AVIII	AII,AIII AIV	AI,AII AV	AIX	AI
2	AI,AIV AIX	AIX	AIV,AVIII	AII,AIII AIV,AIX	AI,AII	AIX	AI,AIX
3	AI,AIV AIX	AIX	AIV,AVIII	AI,AII,AIII AIV,AV,AVI	AI,AII AIII	AIX	AI,AIX
4	AI,AIV AIX	AIX	AI,AII,AIII AIV,AVIII	AII,AIII AIV	AI,AII	AIX	AI,AIX
5	AI,AIV AIX	AV,AIX	AIV,AVIII	AII,AIII	AI,AII	AIX	AI,AII AIX
6	AI,AIV AIX	AI,AIX	AIV,AVIII	AII,AIII,AIV AVI,AVIII,AIX	AI,AII AVIII	AIX	AI,AVI
7	AI,AIV AIX	AIX	AIV,AVIII	AIII	AI,AII,AIII AVIII,AIX	AIX	AI
8	AI,AIV AIX	AV,AIX	AIV,AVIII	AI,AII,AIII AIV,AVI,AIX	AI,AII	AIX	AI,AIX
9	AI,AIV AIX	AIX	AIV,AVIII	AI,AII AIII,AIV	AI,AII	AIX	AI,AIX
10	AI,AIV AIX	AIX	AIV,AVIII	AIII	AI,AII AV	AIX	AI,AIX

As we can see the accuracy of PF-CBR is the highest among all methods in all flexibility dimensions. We perform a *t*-test to verify that the result is statistically significant. Table 5 shows the result.

Table 4. Accuracy mean of each classifier

	J4.8	Bayes	Logistic	CBR	FP-CBR
FB1	0.8663	0.8894	0.8707	0.8584	0.8923
FB2	0.8083	0.7649	0.7074	0.7074	0.8233
FB3	0.7322	0.5786	0.7353	0.6707	0.7384
FB4	0.7077	0.7231	0.7123	0.7138	0.7467
FB5	0.6653	0.6908	0.6526	0.6944	0.7017
FB6	0.7483	0.8090	0.7282	0.8108	0.7449
FB7	0.8416	0.8283	0.8433	0.7216	0.8600

Table 5. Overview of the t-test results of each pairwise classifier

	J4.8-FPCBR	Bayes-FPCBR	Log-FPCBR	CBR-FPCBR
FB1	0.023	0.749	0.055	0.016
FB2	0.041	0.000	0.000	0.000
FB3	0.493	0.000	0.846	0.003
FB4	0.047	0.05	0.166	0.086
FB5	0.138	0.415	0.010	0.020
FB6	0.000	0.891	0.000	0.000
FB7	0.435	0.004	0.053	0.000

4.2 Discussion

Finally the knowledge shown in Table 3 according to items and dimensions used in [2] can be summarized as:

1. *AI(Layout), AIV(Offered Services), AIX(New Services) directly and strongly influences FB1(Expansion)*. Strategic decisions regarding physical modifications (Layout) and changes in the number of delivered services or development of new services are directly related to the expansion dimension of flexibility. This dimension increases flexibility when the cost and time to expand capacity are moderate. However, there is a limit in the expansion capabilities. Once that limit is reached, outsourcing can be an option to increase flexibility over that limit.
2. *AIX(New Services) directly and strongly influences FB2(Distribution of information)*. The development of new services involves a redefinition of the distribution of information within the operations area of the firm. The nature of such redefinition will be directly related to the number of resources and capabilities the new service shares with the existing ones. Service firms competing on innovative services will require higher levels of flexibility on distribution of information.
3. *AIV(Offered Services), AVIII(Customer Participation) directly and strongly influences FB3(Routing)*. When the service firm augments the number of offered services as well as the customer participation in the service delivery process, routing flexibility adjusting is directly related to those strategic decisions. When offering new services, some activities are common to old and new services. However, the need to add value to new services increases the number to activities to be performed and hence, new routes of activities need to be developed. Customer participation involves some degree of variability in the way services are delivered. Many times, different ways and resources need to be offered to customers for self service. This increases the need for routing flexibility.
4. *AIII(Standardization) directly and strongly influences FB4(Equipment)*. The strategic dimension corresponding to the degree of standardization is directly related to the equipment and personnel dimension of flexibility. A high degree of standardization requires a service design based mainly on general use equipment resources. On the other hand, customization involves interchange

of information between the customer and the personnel in order to adapt the service to the customer needs and wishes.

5. *AI(Layout), AII(Push/Pull Orientation) directly and strongly influences FB5 (Market)*. The layout strategic dimensions as well as the Push vs. Pull dimension are directly related to the market flexibility dimension. Adapting to market changes requires a right layout combination as well as the correct Push or Pull orientation in order to minimize possible order delays and/or unattended services.
6. *AIX(New Services) directly and strongly influences FB6(Product and Services)*. The new services strategic dimension and the products and services flexibility dimension are deeply interrelated. The strategic decision of delivering new services entails high levels of production and services flexibility in order to minimize the costs from swapping from old to new services while increasing the total output.
7. *AI(Layout) directly and strongly influences FB7(Process, programming and volume)*. Finally, the flexibility dimension of process, programming and volume is directly linked to the layout strategic dimension. Different layout configurations allow operating at different levels of output according to the different arrangements of the process, programming and volume dimension.

5 Concluding Remarks and Future Works

The present study contributes to the field of service operations management on the base of previous studies performed on the relationship between operations strategy and flexibility. In this case, a new methodology based on CBR tools has been applied leading to interesting findings. The results obtained are consistent with previous studies [2,3]. Nonetheless, the CBR tool applied provides new insights that help to understand more accurately, the effects of the relationship between operations strategy and flexibility. For future research, techniques based on soft computing such as genetic algorithms or neurofuzzy systems will be included to adjust or learn the parameters in the fuzzy rules, belonging functions etc. These parameters are especially important in the final stages of development of the system.

References

1. Ahn, H., Kim, K.-J.: Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach. *Applied Soft Computing* 9(2), 599–607 (2009)
2. Arias, D.: Relationship between Operations Strategy and Size in Engineering Consulting Firms. *International Journal of Service Industry Management* 1(3), 263–285 (2002)
3. Arias, D.: Service Operations Strategy, Flexibility and Performance in Engineering Consulting Firms. *International Journal of Operations and Production Management* 23(11), 1401–1421 (2003)

4. Bayes, T.: An essay towards solving a problem in the doctrine of chances. *Phil. Trans. Roy. Soc.* 53, 370–418 (1783)
5. Castro, J.L., Navarro, M., Sánchez, J.M., Zurita, J.M.: Similarity local adjustment: Introducing attribute risk into the case. In: *Proceedings of the European and Mediterranean Conference on Information Systems*, Alicante, Spain (2006)
6. Castro, J.L., Navarro, M., Sánchez, J.M., Zurita, J.M.: Global risk attribute in case-based reasoning. In: *Proceedings of the 7th International Conference on Case-Based Reasoning*, Belfast, Ireland, pp. 21–30 (2007)
7. Castro, J.L., Navarro, M., Sánchez, J.M., Zurita, J.M.: An automatic method to assign local risk. In: *Proceedings of the IADIS multi conference on computer science and information systems Amsterdam, IADIS 2008*, The Netherlands, pp. 151–157 (2008)
8. Castro, J.L., Navarro, M., Sánchez, J.M., Zurita, J.M.: Loss and Gain Functions for CBR Retrieval. *Information Sciences* 179(11), 1738–1750 (2009)
9. Chang, P.-C., Liu, C.H., Lai, R.K.: A fuzzy case-based reasoning model for sales forecasting in print circuit board industries. *Expert Systems with Applications* 34(3), 2049–2058 (2008)
10. Cheng, M.-Y., Tsai, H.-C., Chiu, Y.-H.: Fuzzy case-based reasoning for coping with construction disputes. *Expert Systems with Applications* 36(2), 4106–4113 (2009)
11. Evans, J.R.: An exploratory study of performance measurement systems and relationships with performance results. *Journal of Operations Management* 22, 219–232 (2004)
12. Gupta, Y.P., Goyal, S.: Flexibility trade-offs in a random flexible manufacturing system: A simulation study. *International Journal of Production Research* 30(3), 527–557 (1992)
13. Li, S.-T., Ho, H.-F.: Predicting financial activity with evolutionary fuzzy case-based reasoning. *Expert Systems with Applications* 36(1), 411–422 (2009)
14. Lin, R.-H., Wang, Y.-T., Wu, C.-H., Chuang, C.-L.: Developing a business failure prediction model via RST, GRA and CBR. *Expert Systems with Applications* (2007)
15. Miah, S.J., Kerr, D.V., Gammack, J.G.: A methodology to allow rural extension professionals to build target-specific expert systems for Australian rural business operators. *Expert Systems with Applications* 36(1), 735–744 (2009)
16. Nieto, M., Arias, D., Minguela, R., Rodríguez, A.: The evolution of operations management contents: an analysis of the most relevant textbooks. *Industrial Management & Data Systems* 99(7-8), 345–353 (1999)
17. Park, Y.J., Kim, B.C., Chum, S.H.: New knowledge extraction technique using probability for case-based reasoning: application to medical diagnosis. *Expert Systems* 23(1), 2–20 (2006)
18. Safizadeh, M.H., Ritzman, L.P., Mallick, D.: Revisiting Alternative Theoretical Paradigms in Manufacturing Strategy. *Production and Operations Management* 9(2), 111–127 (2000)
19. Shiue, W., Li, S.-T., Chen, K.-J.: A frame knowledge system for managing financial decision knowledge. *Expert Systems with Applications* 35(3), 1068–1079 (2008)
20. Tersinea, R., Harvey, M.: Global Customerization of Markets Has Arrived! *European Management Journal* 16(1), 79–90 (1998)
21. Witten, I.H., Frank, E.: *Data Mining: Practical machine learning tools and techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Semantic-Based Top-k Retrieval for Competence Management

Umberto Straccia¹, Eufemia Tinelli², Simona Colucci², Tommaso Di Noia²,
and Eugenio Di Sciascio²

¹ ISTI-CNR, Via G. Moruzzi 1, I-56124 Pisa, Italy

² SisInfLab-Politecnico of Bari, via Re David 200, I - 70125 Bari, Italy
straccia@isti.cnr.it,

{e.tinelli,s.colucci,t.dinoia,disciascio}@poliba.it

Abstract. We present a knowledge-based system, for skills and talent management, exploiting semantic technologies combined with top-k retrieval techniques. The system provides advanced distinguishing features, including the possibility to formulate queries by expressing both strict requirements and preferences in the requested profile and a semantic-based ranking of retrieved candidates. Based on the knowledge formalized within a domain ontology, the system implements an approach exploiting top-k based reasoning services to evaluate semantic similarity between the requested profile and retrieved ones. System performance is discussed through the presentation of experimental results.

1 Introduction

Nowadays more and more companies choose to employ e-recruiting systems to automatically assign vacant job positions. Such systems allow for electronically managing the whole recruitment process, reducing the related cost. E-recruiting systems efficiency is therefore significantly affected by the efficacy of the framework underlying the match between recruiters requests and candidates profiles stored. In available skill management systems, information about candidates employment and personal data as well as certifications and competence is usually modeled through relational databases with customized and structured templates. Nevertheless, even though a Data Base Management System (DBMS) is surely suitable for storage and retrieval, relational query languages do not allow for the flexibility needed to support a discovery process as complex as recruitment. The `order by` statement and the `min` and `max` aggregation operators are generally used to retrieve the *best tuples* but, in real scenarios, there are no candidates that are better than the others ones w.r.t. every selection criteria. Moreover, if exact matches are lacking, worse alternatives must be often accepted or the original requirements have to be negotiated for compromises.

Logic-based techniques and technologies permit to make more efficient and flexible the recruitment process. The system we present here automatically performs a match-making process between available candidate profiles and vacant job positions according to mandatory requirements and preferences provided by a recruiter. In order to perform it, we need a language suitable for data intensive applications with a good compromise between expressiveness and computational complexity. The system performs non-exact match through top-k retrieval techniques: it uses a match engine which performs top-k queries over a DLR-lite [4] Knowledge Base (KB) providing a ranked list of candidates.

In the remaining we proceed as follows: Section 2 motivates our proposal, also by comparing it with relevant related work; Section 3 shortly recalls language and algorithms we adopted. In Section 4 the proposed system is presented with particular reference to the evaluation of its performance. Finally, conclusions close the paper.

2 Why Another System for HRM

Currently, several solutions for talent management¹ and e-recruitment are available on the market. Most of them are complete enterprise suites supporting human resource management, including solutions that, even though improving the recruitment process by means of innovative media and tools, do not bring a significant novelty charge with them. Available solutions in fact exploit databases to store candidate personal and employment information, and do not ground on a logic-based structure.

One of the few logic-based solutions to recruitment and referral process is, to the best of our knowledge, STAIRS², a system in use at US Navy Department allowing to retrieve referral lists of best qualified candidates w.r.t. a specific mansion, according to the number of required skills they match. The commercial software supporting STAIRS is RESUMIX³ an automated staffing tool making use of artificial intelligence techniques and adopted only as an internal tool. The system allows also to distinguish skills in *required* and *desired* ones in the query formulation: all required skills must be matched by the retrieved candidate, differently from *desired* ones.

We propose here a logic-based solution to recruitment process, allowing for distinguishing in required and preferred skills and exploiting a *Skills Ontology*, designed in (a subset of) OWL DL, to model experiences, education, certifications and abilities of candidates. The system translates a user request into a union of conjunctive queries for retrieving the best candidate to cover a given position. Hence, in order to perform a match both the user request and candidates CVs (which we generally call *profiles*) are defined w.r.t. the same Skills Ontology.

In order to understand the advantages of our system w.r.t. not logic-based solutions, we provide here a tiny example: imagine you are a recruiter, with the following request: *"I'm looking for a candidate, preferably expert in Artificial Intelligence with an experience of at least two years and necessarily endowed with a doctoral degree"*. Let us suppose that there are three candidates **Sarah**, **Paul** and **Bill** skilled as presented in Figure 1 all having a doctoral degree fulfilling the strict constraint of the user request. Looking both at the three profile descriptions and at the original request, we will rank the three candidates as (1) Paul; (2) Bill; (3) Sarah w.r.t. the preference expressed by the user. In fact, reasonably, the skills of Paul are very close to the requested ones even if he does not fully satisfy the requested experience (in years). On the other side, since ontology and semantic technologies relate to Artificial Intelligence Bill skills seems to be more useful than Sarah ones. It is easy to see that the only way to automatically perform such a ranking is exploiting a semantic-based approach, making use of a domain ontology modeling competence hierarchies and relations. Moreover, thanks to the

¹ <http://www.attract-hr.com/cm/about>,
http://www.oracle.com/applications/human_resources/irecruit.html

² <http://www.hrojax.navy.mil/forms/selectguide.doc>

³ <http://www.cpol.army.mil>

Name	Knowledge
Sarah	Excellent experience in Business Intelligence (5 years) ...
Paul	1 years experienced in Knowledge Representation and Fuzzy Logic. Good knowledge of OWL, DLs, DL-lite family, ...
Bill	Skilled in ontology modeling with knowledge of semantic technologies ...

Fig. 1. Example of candidate skills

information modeled in the ontology, the system is able to return all scores computed for each feature of the retrieved profiles.

A relevant aspect of our work is the exploitation of classical relational database systems (RDBMS) and languages *i.e.*, SQL, for storing the reference ontology and candidate CVs and to perform reasoning tasks. Using the system, both recruiters and candidates refers to the same model of the domain knowledge. Several approaches ([7], [22], [3], [16]) have been presented in which databases allow users and applications to access both ontologies and other structured data in a seamless way. A possible optimization consists in caching the classification hierarchy in the database and to provide tables maintaining all the subsumption relationships between primitive concepts. Such an approach is taken in *Instance Store (iS)* [2], a system for reasoning over OWL KBs specifically adopted in bio and medical-informatics domains. *iS* is also able –by means of a hybrid reasoner/database approach– to reply to instance retrieval queries w.r.t. an ontology, given a set of axioms asserting class-instance relationships. Nevertheless, *iS* reduces instance retrieval to pure TBox reasoning and is able to return only exact matches (*i.e.*, instance retrieval) whilst we use an enriched relational schema storing only the Abox (*i.e.*, facts) in order to provide a logic-based ranked list of results and the not classified ontology. Other systems using RDBMS in order to deal with large amounts of data are *QuOnto*⁴ and *Owlgres*⁵. They are DL-Lite reasoners providing consistency checking and conjunctive query services. Neither QuOnto nor Owlgres returns a ranked list of results.

As hinted before, our system also allows for formulating queries by distinguishing between preferred and required skills by exploiting top-k retrieval techniques. Top-k queries [12] ensure an efficient ranking support in RDBMSs letting the system to provide only a subset of query results, according to a user-specified ordering function (which generally aggregates multiple ranking criteria). The general problem of preference handling in RDBMS in information retrieval systems [5] has been faced from two competing perspectives: *quantitative*– models, coping with preferences by means of utility functions [12,15] and *qualitative*– models, using logical formulas [10,5,8]. Various approaches using numerical ranking in combination with either the top-k model [13,9,23], the Preference SQL [11] or the Preference XPath [10] have been also devised.

3 System Background: Top-k Retrieval for DLR-Lite

For computational reasons the particular logic we adopt is based on an extension of the DLR-Lite [4] Description Logic (DL) [1] without negation. DLR-Lite is different from usual DLs as it supports n -ary relations ($n \geq 1$), whereas DLs support usual unary

⁴ <http://www.dis.uniroma1.it/~quonto/>

⁵ <http://pellet.owldl.com/owlgres/>

relations (called *concepts*) and binary relations (called *roles*). The DL will be used in order to define the relevant abstract concepts and relations of the application, while data is stored into a database. On the other hand, conjunctive queries will be used to describe the information needs of a user and to rank the answers according to a scoring function. The logic extends DLR-Lite by enriching it with built-in predicates. Conjunctive queries are enriched with scoring functions that allow to rank and retrieve the top-k answers, that is, we support *Top-k Query Answering* [14,17,18,19,20], (find top-k scored tuples satisfying query), e.g., “find candidates with excellent knowledge in DLR-Lite”, where EXCELLENT is a function of the years of experience.

Due to lack of space, we do not delve into details about the query and representation language at the basis of top-k retrieval problem (detailed in [19] for the interested reader) and just recall its definition in the following.

Top-k Retrieval. Given a knowledge base \mathcal{K} , and a union of conjunctive queries \mathbf{q} , retrieve k tuples $\langle c, s \rangle$ that instantiate the query relation q with maximal score (if k such tuples exist), and rank them in decreasing order relative to the score s , denoted

$$ans_k(\mathcal{K}, \mathbf{q}) = Top_k ans(\mathcal{K}, \mathbf{q}) .$$

A knowledge base $\mathcal{K} = \langle \mathcal{F}, \mathcal{O} \rangle$ consists of a *facts component* \mathcal{F} and an *Ontology component* \mathcal{O} . Informally, facts component is used to store data into a database and the ontology component is used to define the relevant abstract concepts and relations of the application domain.

The detailed description of the algorithm embedded in our system to solve top-k retrieval problem is beyond the scope of this work. The algorithm is an extension of the one described in [4,17,19] and has been implemented also as part of the SoftFacts system⁶.

4 System Evaluation

The proposed system has been implemented by plugging the Top-K DLR-Lite retrieval into I.M.P.A.K.T. [21], a system for skills and knowledge management developed by *Data Over Ontological Models s.r.l.*⁷ as a commercial solution implementing the skill matching framework designed in [6]. The efficiency and scalability of the approach has been tested using the skill ontology underlying I.M.P.A.K.T. Both requests and candidate profiles have been modeled w.r.t. to this ontology containing 2594 relations, both unary (classes) and n-ary ones, and 5119 axioms. The main structure of the ontology is depicted in Figure 2.

Level represents profile education such as certifications, masters, doctorate, etc.; *Knowledge* represents technical skill and specific competences of the candidate; *ComplementarySkill* represents abilities and hobbies of the candidate; *JobTitle* represents work experiences of the candidate; *Industry* representing sectors (institutes, research laboratories, companies, etc.) in which candidate works/worked; *Language* represents the knowledge of foreign languages. Data properties, have been used to represent years of experience, degree final mark and knowledge level of foreign languages.

⁶ <http://gaia.isti.cnr.it/~straccia/software/SoftFacts/SoftFacts.html>

⁷ <http://www.doom-srl.it/>

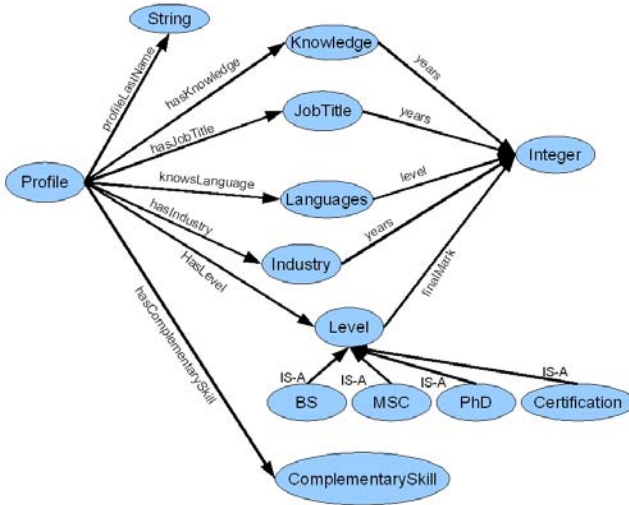


Fig. 2. A graphical representation of the ontology structure

The system exploits the user interface of *I.M.P.A.K.T.*, shown in Figure 3. Panels (a), (b) and (d) allow the recruiter to compose her semantic-based request. In fact, in menu (a) all the entry points are listed whilst panel (b) allows to search for ontology concepts according to their meaning and section (d) enables the user to explore both taxonomy and properties of a selected concept. Entry points in menu (a) represent, to some extent, the main classes and relations represented in Figure 2. Once an item is selected in panel (d), the corresponding panel, representing the item itself, is dynamically filled and added to panel (e). This latter enumerates all the requested features in the query. For each of them, the GUI of *I.M.P.A.K.T.* allows: (1) to define if the feature is strict (crisp) or negotiable (fuzzy); (2) to delete the whole feature; (3) to complete the description showing all the elements (concepts, object properties and data properties) that could be added to the selected feature; (4) to edit each feature piece as well as existing data properties. Finally, panel (c) enables searches like “*I’m searching a candidate like John Doe*” i.e., it is useful to model all those situations where you are looking for a candidate whose skills and knowledge are similar to the ones of *John Doe*. In this case, the job-seeker fills first and/or last name field of the known candidate and the system consider her/his profile as starting request. The user can view the query –automatically generated– and eventually she can edit it before starting a new search.

In the experiments we carried out, we considered 100.000 automatically generated CVs and stored them into a database having 17 relational tables. In Figure 4 we show the ontology axioms mapping the relational tables involved in the proposed queries, in order to provide the reader with the alphabet of the query language. Each axiom renames with the role name given as first parameter the table defined as second parameter with all its fields. We build several queries, with/without scoring atom and submitted them to the system, with different values for k in case of top-k retrieval ($k \in \{1, 10\}$). We run the experiments using the top-k retrieval *SoftFacts* system as back-end. No indexes

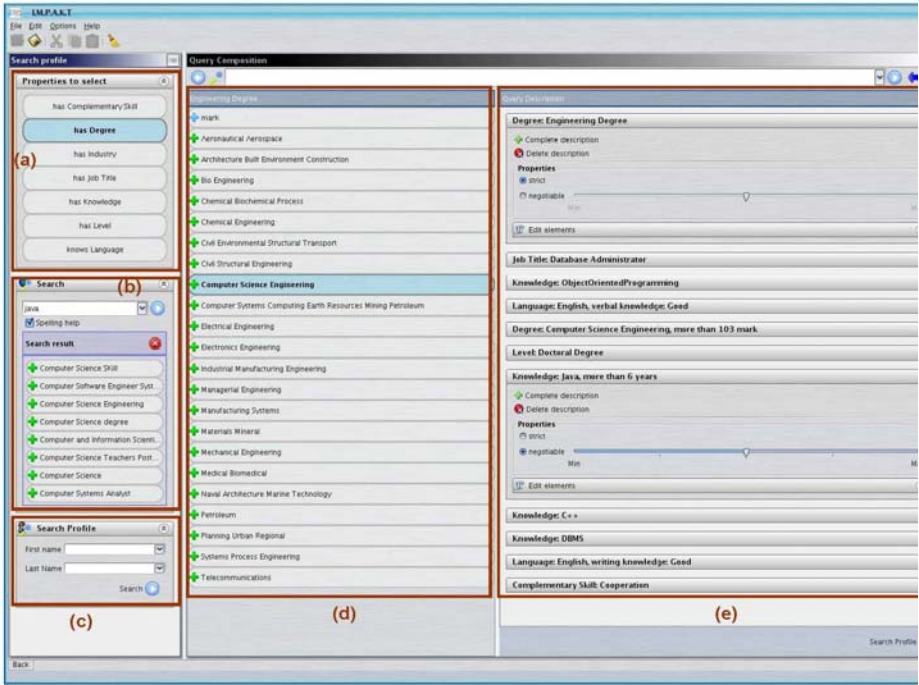


Fig. 3. Query composition GUI

have been used for the facts in the relational database. The concept and role hierarchy used in the experiment queries is clarified in Figure 5. The queries at the basis of the experimentation are listed below, together with the corresponding encoding in *Top-K DLR-Lite*.

1. Retrieve CV’s with knowledge in Engineering Technology

$$q(id, lastName, hasKnowledge, Years) \leftarrow profileLastName(id, lastName), hasKnowledge(id, classID, Years, Type, Level), knowledgeName(classID, hasKnowledge), Engineering_and_Technology(classID)$$

2. Retrieve CV’s referred to candidates with degree in Engineering

$$q(id, lastName, hasDegree, mark) \leftarrow profileLastName(id, lastName), hasDegree(id, classID, mark), DegreeName(classID, hasDegree), Engineering_Degree(classID)$$

3. Retrieve CV’s referred to candidates with knowledge in Artificial Intelligence and degree final mark not less than 100/110

$$q(id, lastName, hasKnowledge, Years, degreeName, mark) \leftarrow profileLastName(id, lastName), hasKnowledge(id, classID, Years, Type, Level), knowledgeName(classID, hasKnowledge), Artificial_Intelligence(classID), hasDegree(id, degreeID, mark), DegreeName(degreeID, hasDegree), (mark \geq 100)$$

```
(MAP-ROLE Profile Profile (profID, FirstName, LastName, Genre,
  BirthDate, CityOfBirth, Address, City, ZipCode, Country, IdentityCode,
  PhoneNumber, Email, WebPage, Nationality, ResidentIn,
  SuddenJobAvailability, JobLocation, FlexibleWorkHours,
  TravelingAvailability, CertificationInstitute, Salary, CarAvailability))
(MAP-ROLE degreeName Degree (degID, Name)) (MAP-ROLE knowledgeName
Knowledge (knowID, Name)) (MAP-ROLE knowledgeLevelName
KnowledgeLevel (knowLevelID, Name)) (MAP-ROLE knowledgeTypeName
KnowledgeType (knowTypeID, Name)) (MAP-ROLE knowledgeLevelName
KnowledgeLevel (knowLevelID, Name)) (MAP-ROLE knowledgeTypeName
KnowledgeType (knowTypeID, Name)) (MAP-ROLE hasDegree HasDegree
(profID, classID, Mark)) (MAP-ROLE hasKnowledge HasKnowledge
(profID, classID, Years, Type, Level))
```

Fig. 4. Excerpt of relational tables ontology mapping

```
(IMPLIES Engineering_and_Technology Knowledge)
(IMPLIES Artificial_Intelligence Computer_Science_Skill)
(IMPLIES Information_Systems Computer_Science_Skill)
(IMPLIES Computer_Science_Skill Engineering_and_Technology)
(IMPLIES Engineering_Degree Degree)
(IMPLIES Fuzzy_Artificial_Intelligence)
(IMPLIES Data_Mining Artificial_Intelligence)
(IMPLIES Machine_Learning Artificial_Intelligence)
(IMPLIES Knowledge_Rappresentation Artificial_Intelligence)
(IMPLIES Natural_Language Artificial_Intelligence)
(MAP-ROLE profileLastName Profile (profID, LastName))
(IMPLIES (SOME[1] profileLastName) Profile)
```

Fig. 5. Excerpt of concepts and roles hierarchy

4. Retrieve CV's referred to candidates with knowledge in Artificial Intelligence, degree in Engineering with final mark not less than 100/110

```
q(id, lastName, hasKnowledge, Years, degreeName, mark)
← profileLastName(id, lastName), hasKnowledge(id, classID, Years, Type, Level),
  knowledgeName(classID, hasKnowledge), Artificial_Intelligence(classID),
  hasDegree(id, degreeID, mark), DegreeName(degreeID, hasDegree), Engineering_Degree(degreeID),
  (mark ≥ 100)
```

5. Retrieve CV's referred to candidates experienced in Information Systems (not less than 15 years), with degree final mark not less than 100

```
q(id, lastName, hasKnowledge, Years, degreeName, mark)
← profileLastName(id, lastName), hasKnowledge(id, classID, Years, Type, Level),
  knowledgeName(classID, hasKnowledge), Information_Systems(classID), (Years ≥ 15)
  hasDegree(id, degreeID, mark), DegreeName(degreeID, hasDegree),
  (mark ≥ 100)
```

6. Retrieve top-k CV's referred to candidates with knowledge in Artificial Intelligence and degree final mark scored according to $rs(mark; 100, 110)$

```
q(id, lastName, degreeName, mark, hasKnowledge, years)
← profileLastName(id, lastName), hasDegree(id, degreeID, mark), degreeName(degreeID, degreeName),
  hasKnowledge(id, classID, years, type, level), knowledgeName(classID, hasKnowledge),
  Artificial_Intelligence(classID), OrderBy(s = rs(mark; 100, 110))
```

7. Retrieve CV's referred to candidates with degree in Engineering and final mark scored according to $rs(mark; 100, 110)$

```
q(id, lastName, hasDegree, mark)
← profileLastName(id, lastName), hasDegree(id, classID, mark), DegreeName(classID, hasDegree),
  Engineering_Degree(classID), OrderBy(s = rs(mark; 100, 110))
```

8. Retrieve top-k CV's referred to candidates with knowledge in Artificial Intelligence, degree in Engineering with final mark scored according to $rs(mark; 100, 110)$

```
q(id, lastName, hasKnowledge, Years, degreeName, mark)
← profileLastName(id, lastName), hasKnowledge(id, classID, Years, Type, Level),
  knowledgeName(classID, hasKnowledge), Artificial_Intelligence(classID),
  hasDegree(id, degreeID, mark), DegreeName(degreeID, hasDegree), Engineering_Degree(degreeID),
  OrderBy(s = rs(mark; 100, 110))
```

9. Retrieve CV's referred to candidates with knowledge in Information Systems and with degree final mark and years of experience both scored according to $rs(mark; 100, 110) \cdot 0.4 + rs(years; 15, 25) \cdot 0.6$;

```
q(id, lastName, hasKnowledge, Years, degreeName, mark)
← profileLastName(id, lastName), hasKnowledge(id, classID, Years, Type, Level),
  knowledgeName(classID, hasKnowledge), Information_Systems(classID), (Years ≥ 15)
  hasDegree(id, degreeID, mark), DegreeName(degreeID, hasDegree),
  OrderBy(s = rs(mark; 100, 110) · 0.4 + rs(years; 15, 25) · 0.6)
```

10. Retrieve CV's referred to candidates with good knowledge in Artificial Intelligence, and with degree final mark, years and level of experience scored according to $rs(mark; 100, 110) \cdot 0.4 + rs(years; 15, 25) \cdot pref(level; Good/0.6, Excellent/1.0) \cdot 0.6$;

```
q(id, lastName, degreeName, mark, hasKnowledge, years, kType)
← profileLastName(id, lastName), hasDegree(id, degreeID, mark), degreeName(degreeID, degreeName),
  hasKnowledge(id, classID, years, type, level), knowledgeLevelName(level, kType), Good(level),
  knowledgeName(classID, hasKnowledge), Artificial_Intelligence(classID),
  OrderBy(s = rs(mark; 100, 110) · 0.4 + rs(years; 15, 25) · pref(level; Good/0.6, Excellent/1.0) · 0.6)
```

Queries 1-5 are crisp queries. There is no preference expressed and no actual ranking. As each answer has score 1.0, we would like to verify whether there is a retrieval time difference between retrieving all records, or just the k answers. The other queries are top-k queries. In query 9, we show an example of score combination, with a preference on the number of years of experience over the degree's mark, but scores are summed up. In query 10, we use the preference scoring function

$$pref(level; Good/0.6, Excellent/1.0)$$

that returns 0.6 if the level is good, while returns 1.0 if the level is excellent. In this way we want to privilege those with an excellent knowledge level over those with a good level of knowledge. In Fig 6 we report the output of query 10.

10 Results found (Top-10):							
Score	id	lastName	degreeName	mark	hasKnowledge	Years	KType
0.42	299	Jensen	Animal_Science	109.0	Fuzzy	16.0	Excellent
0.36	938	Young	Civil_Structural_Engineering	109.0	Machine_Learning	11.0	Excellent
0.28	360	Taylor	Animal_Health	107.0	Knowledge_Rappresentation	15.0	Excellent
0.08	956	Cook	Physiotherapy	102.0	Natural_Language	7.0	Good
0.08	187	Allan	Ecology	102.0	Artificial_Intelligence	2.0	Excellent
0.04	1109	Graham	African_Sub_Saharan	101.0	Machine_Learning	4.0	Excellent
0.0	1081	James	Humanities	75.0	Artificial_Intelligence	5.0	Good
0.0	682	Scott	Film_Studies_Television	82.0	Data_Mining	7.0	Good
0.0	538	Cox	Development_Studies	95.0	Machine_Learning	6.0	Good
0.0	604	Scott	English_Literature	61.0	Knowledge_Rappresentation	1.0	Good

Fig. 6. Retrieval output of query 10

Size 100000				
Query	All	top-1	top-10	$ ans(\mathcal{K}, q) $
1	12.344	3.596	6.182	3985
2	0.375	0.116	0.125	445
3	0.366	0.117	0.118	19
4	4.263	3.877	3.897	8
5	0.397	0.325	0.357	19
6	0.104	0.103	0.099	40
7	0.209	0.178	0.189	128
8	4.086	3.895	3.998	20
9	0.471	0.422	0.395	201
10	0.391	0.357	0.373	19
Average	2.301	1.295	1.573	488
Median	0.394	0.341	0.365	30

Fig. 7. Retrieval times

The tests have been performed on a MacPro machine with Mac OS X 10.5.5, 2 x 3 GHz Dual-Core processor and 9 GB or RAM and the results are shown in Fig. 7 (time is measured in seconds). Let us consider few comments about the results:

- overall, the response time is quite good (almost fraction of second) taking into account the non negligible size of the ontology, the number of CVs and that we did not consider any index for the relational tables;
- if the answer set is large, e.g., query 1, then there is a significant drop in response time, for the top-k case;
- for each query, the response time is increasing while we increase the number of retrieved records.

5 Conclusion and Future Work

We presented an innovative and scalable logic-based system for efficiently managing skills and experiences of candidates in the e-recruitment field. The system grounds on a Skill Ontology in order to return a ranked list of profiles and on scoring functions in order to weight each feature of the retrieved profiles. Differently from existing recruitment systems, our approach allows to express a user request as the composition of both mandatory requirements and preferences, by means of top-k retrieval techniques. The implemented retrieval framework was embedded into an existing system for skill management and experiments conducted on a preliminary profiles dataset show a satisfiable behavior. Future work aims at evaluating system performance on several datasets and at providing user-friendly explanation facilities to better clarify scores of obtained results.

Acknowledgments

We wish to acknowledge partial support of PS_092 and PS_121.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
2. Bechhofer, S., Horrocks, I., Turi, D.: The OWL instance store: System description. In: Nieuwenhuis, R. (ed.) CADE 2005. LNCS (LNAI), vol. 3632, pp. 177–181. Springer, Heidelberg (2005)

3. Broekstra, J., Kampman, A., van Harmelen, F.: 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)
4. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Data complexity of query answering in description logics. In: Proc. of KR 2006, pp. 260–270 (2006)
5. Chomicki, J.: Querying with Intrinsic Preferences. In: Jensen, C.S., Jeffery, K., Pokorný, J., Šaltenis, S., Bertino, E., Böhm, K., Jarke, M. (eds.) EDBT 2002. LNCS, vol. 2287, pp. 34–51. Springer, Heidelberg (2002)
6. Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F.M., Ragone, A.: Semantic-based skill management for automated task assignment and courseware composition. *J. Univ. Comp. Sci.* 13(9), 1184–1212 (2007)
7. Das, S., Chong, E.I., Eadon, G., Srinivasan, J.: Supporting ontology-based semantic matching in RDBMS. In: Proc. of VLDB 2004, pp. 1054–1065. VLDB Endowment (2004)
8. Hafenrichter, B., Kießling, W.: Optimization of relational preference queries. In: Proc. of ADC 2005, pp. 175–184. Australian Computer Society, Inc., Darlinghurst (2005)
9. Hristidis, V., Koudas, N., Papakonstantinou, Y.: PREFER: A system for the efficient execution of multi-parametric ranked queries. In: Proc. of ACM SIGMOD, pp. 259–270. ACM, New York (2001)
10. Kießling, W.: Foundations of preferences in database systems. In: Proc. of VLDB 2002, pp. 311–322. Morgan Kaufmann, Los Altos (2002)
11. Kießling, W., Köstler, G.: Preference SQL - design, implementation, experiences. In: Proc. of VLDB 2002, pp. 990–1001. Morgan Kaufmann, Los Altos (2002)
12. Li, C., Chang, K.C.-C., Ilyas, I.F., Song, S.: RankSQL: query algebra and optimization for relational top-k queries. In: Proc. of ACM SIGMOD 2005. ACM Press, New York (2005)
13. Li, C., Soliman, M.A., Chang, K.C.-C., Ilyas, I.F.: RankSQL: supporting ranking queries in relational database management systems. In: Proc. of VLDB 2005, pp. 1342–1345. VLDB Endowment (2005)
14. Lukaszewicz, T., Straccia, U.: Top-k retrieval in description logic programs under vagueness for the semantic web. In: Prade, H., Subrahmanian, V.S. (eds.) SUM 2007. LNCS (LNAI), vol. 4772, pp. 16–30. Springer, Heidelberg (2007)
15. Bosc, P., Pivert, O.: SQLf: a relational database language for fuzzy querying. *IEEE Transactions on Fuzzy Systems* 3(1), 1–17 (1995)
16. Pan, Z., Heflin, J.: DLDB: Extending Relational Databases to Support Semantic Web Queries. In: Proc. of PSSS1, vol. 89, pp. 109–113. CEUR-WS.org (2003)
17. Straccia, U.: Answering vague queries in fuzzy DL-Lite. In: Proc. of IPMU 2006, pp. 2238–2245. E.D.K, Paris (2006)
18. Straccia, U.: Towards top-k query answering in deductive databases. In: Proc. of SMC 2006, pp. 4873–4879. IEEE, Los Alamitos (2006)
19. Straccia, U.: Towards top-k query answering in description logics: the case of DL-Lite. In: Fisher, M., van der Hoek, W., Konev, B., Lisitsa, A. (eds.) JELIA 2006. LNCS (LNAI), vol. 4160, pp. 439–451. Springer, Heidelberg (2006)
20. Straccia, U.: Towards vague query answering in logic programming for logic-based information retrieval. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 125–134. Springer, Heidelberg (2007)
21. Tinelli, E., Cascone, A., Ruta, M., Di Noia, T., Di Sciascio, E., Donini, F.M.: I.M.P.A.K.T.: An innovative, semantic-based skill management system exploiting standard SQL. In: Proc. of ICEIS 2009, vol. AIDSS, pp. 224–229 (2009)
22. Wilkinson, K., Sayers, C., Kuno, H.A., Reynolds, D.: Efficient RDF Storage and Retrieval in Jena2. In: Proc. of SWDB 2003, pp. 131–150 (2003)
23. Yu, H., Hwang, S.-W., Chang, K.C.-C.: RankFP: A Framework for Supporting Rank Formulation and Processing. In: Proc. of ICDE 2005, pp. 514–515. IEEE Comp. Soc. Press, Los Alamitos (2005)

A New Strategy Based on GRASP to Solve a Macro Mine Planning*

María-Cristina Riff, Eridan Otto, and Xavier Bonnaire

Departamento de Informática
Universidad Técnica Federico Santa María
Valparaíso, Chile

{Maria-Cristina.Riff,Eridan.Otto,Xavier.Bonnaire}@inf.utfsm.cl

Abstract. In this paper we introduce a greedy randomized adaptive search procedure (GRASP) algorithm for solving a copper mine planning problem. In the last 10 years this real-world problem has been tackled using linear integer programming and constraint programming. Our mine planning problem is a large scale problem, thus in order to find an optimal solution using complete methods, the model was simplified by relaxing many constraints. We now present a Grasp algorithm which works with the complete model and it is able to find better feasible near-optimal solutions, than the complete approach that has been used until now.

Keywords: Heuristic Search, Mine Planning, Real-world problems.

1 Introduction

Chile is the world's largest copper producer and the profit obtained by the copper extraction has an important role in the country economy. There are some approaches published in the literature related to mine problems, but they are usually applied to open pit mines, [15], [14]. Our particular problem is about an underground copper mine. In the last 10 years, the copper mine planning problem has been tackled using linear and mixed integer programming, and we have recently applied constraint programming techniques to it, [7]. However, none of these techniques has been able to entirely solve our problem; thus it must be simplified by relaxing some geological and physical constraints. This problem belongs to large scale combinatorial optimization problems.

On the other hand, metaheuristics have solved complex problems successfully like timetabling problems, [4], [2], scheduling [1], vehicle routing problems [3], travel salesman problems [5], constraint satisfaction problems [18], [19], [9], short-term electrical generation scheduling problems [17], and real-world applications [6], [8]. Our problem is similar to both, the scheduling problem and the travel salesman problem, but it has many other constraints that must be considered by the algorithm, in order to give better feasible solutions. GRASP (Greedy

* Partially supported by the FONDEF Project: Complex Systems, and Fondecyt Project 1080110.

Randomized Adaptive Search Procedure) is a metaheuristic for finding approximated solutions to combinatorial optimization problems. It was first introduced by Feo and Resende [11] in a paper describing a probabilistic heuristic for set covering. Since then, GRASP has experienced continued development and has been applied in a wide range of problem areas, [12].

The purpose of this work goes in two directions: The first one is related to the problem; in this context our goal is not to obtain the optimal solution but a good one, which can be better than the solution found by the traditional approach. The second one is related to the greedy randomized adaptive search procedure (GRASP) technique; our aim here is to show that it can be successfully applied to solve this real-world hard problem.

The paper is organized as follows: In the next section we define our real-world problem. In section three we present the linear integer programming model. The algorithm is introduced in section four. Section five presents the results obtained using random generated mine planning problems. Finally, in the last section we present the conclusions and the future issues that might come out of our work.

2 Problem Definition

For the purpose of resource modelling and mine planning, our mine has been divided into S sections. Each section is also subdivided into m blocks and each block is composed of 10 cells. The goal is to find the sequence of cells extraction that maximizes the profit. Our real-world problem is one section of an underground copper mine. The exploitation technique for this kind of mines requires the following two steps: To construct access tunnels, and to implement other facilities for extracting the cells of a block by using a bottom up procedure. The problem has many types of constraints, namely accessibility, geological, and capacity constraints.

Accessibility constraint: To have access to any cell within a block, its first cell must be previously extracted. We suppose that the access cost of a block is charged only once, when its first cell is extracted.

Geological constraint: The major set of constraints is called “subsidence constraints”. Mine subsidence is the movement of the ground surface as a result of the collapse or failure of underground mine work, [16]. These constraints determine a physical relation among blocks.

The action of extracting a block implies the constraints: “the blocks that belong to its upper cone can not be exploited in a future time”.

Figure 1 is a simplified 2D picture that shows the upper cone of block k . When the first cell of block k is extracted the blocks belonging to its upper cone become definitively *inaccessible blocks*, these blocks are painted in black. The white blocks could be exploited in a future time.

Capacity: The maximum number of cells extraction allowed is K_y cells by year.

The maximal profit value relates to both cell extraction and block access costs and to the cell copper concentration. The extraction of the first cell of block k implies the following actions:

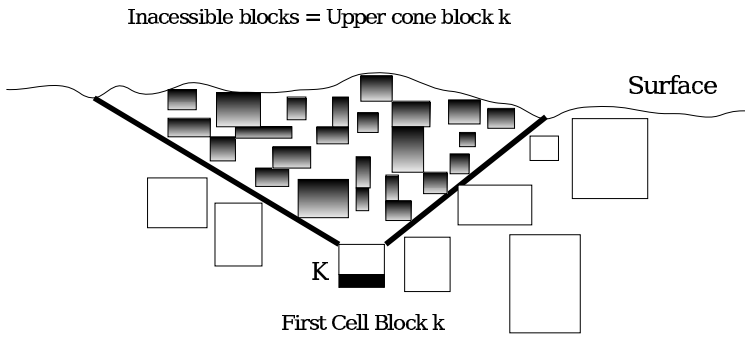


Fig. 1. Subsidence constraint

1. To pay the access cost to block k
2. To avoid in the future the exploitation of the blocks belonging to the block k upper cone
3. To allow the extraction of the other cells of block k
4. To pay the extraction cost of each cell
5. To obtain the copper profit given by the cell copper concentration

We consider that the extraction of j th cell of a block, $j > 1$ implies only the points 4 and 5 listed above. The optimization is for a 20-year planning.

Remark 1. We use the Net Present Value as a way of comparing the value of money now with the value of money in the future. Thus, we apply a discount rate which refers to a percentage used to reflect the time value of money. Because of discounting the idea is to exploit particularly attractive blocks early but this makes the blocks in the cone inaccessible.

3 Problem Model

In this section we present the linear programming model. The idea is to find the sequence of cells extractions that maximizes the profit. It is evaluated computing the Net Present Value of the planning.

Variables:

$$z_{i,t} = \begin{cases} 1 & \text{if block } i \text{ is exploited at time } t \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$h_{j,t} = \begin{cases} 1 & \text{if blocks group } j \text{ is accessible at time } t \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

considering that a section is composed by m blocks, the horizon planning time equal to H , and where t represents the time.

Goal : Maximize NPV (Net Present Value)

Objective Function

$$NPV = \sum_{i,t} r^t U_i z_{i,t} - \sum_{j,t} r^t C_j h_{j,t} \quad (3)$$

where r is the discount rate, U_i is the benefit of the block i , C_j is the accessibility cost of the block j .

Constraints

– Consistency

$$\sum_t z_{i,t} \leq 1, \forall i \quad (4)$$

The block i could be extracted at most once

$$\sum_t h_{j,t} \leq 1, \forall j \quad (5)$$

The access to the block is constructed only once

– Accessibility

$$z_{i,t} \leq \sum_{s \leq t} h_{j,s}, \forall i \in j, \forall t \quad (6)$$

The access to the block must be done before its exploitation

– Subsidence

$$z_{i,t} + z_{i',s} \leq 1, \forall i, \forall t, \forall i' \in I(i), \forall s \geq t \quad (7)$$

where $I(i)$ is the set of blocks belonging to the upper cone of block i . The blocks in the upper cone of block i cannot be exploited after the extraction of the block i .

– Capacity

$$\sum_i \sum_t z_{i,t} \leq \sum_y K_y \quad (8)$$

There is a maximum number of blocks to be extracted.

Looking at the model we can observe that the problem has a lot of constraints. The model becomes very complex to be solved in its complete version. We tried to apply constraint programming techniques, [7], in order to filter the domain of variables during the instantiations, but the problem is still hard to be solved with these techniques.

In the following sections we present our approach. This uses heuristics in a Grasp framework. We have selected this metaheuristic for this work given the success of some reported real applications using grasp based approaches, [13].

4 Grasp Approach

In the following sections we introduce the components of the GRASP algorithm for Mine Planning.

4.1 Representation

In our approach the representation is a list where each element represents a block number. The same block number could appear more than once on the list. The number of a block appears as many times as the number of its cells have been extracted. For instance a list (3 6 7 4 7 5 5 ..) means that the first cell extracted is from block three, the third one is from block 7 and the fifth one is also from block 7. This representation is useful to manage the constraints that we named *consistency constraints*, equation 4 in the problem model section. With this representation we do not need to worry about the sequence of cells extraction inside a block, because it is directly deduced of the order of the list. Thus, by using this representation, all possible solutions satisfy the consistency constraints. Moreover, we can easily identify the scheduling of cell extractions of each block.

4.2 Evaluation Function

The hardest group of constraints is the subsidence constraints. The evaluation function has the two components of the equation 3 of the objective function described in the problem model:

- The profit obtained by cell extractions
- The cost of blocks accessibility

We need to point out that we also include in the evaluation function an opportunity cost, that takes into account the following issues: “when a block is exploited all the blocks belonging to its upper cone become inaccessible for ever”. Thus, we consider a cost related to the physical impossibility of obtaining the copper concentration of these blocks in the future time. In each instantiation the algorithm is looking for the cell of a block k whose gain financially justifies the prohibition of extraction of the blocks in its k upper cone in the future. This cost is included as a penalization in the evaluation function. It is calculated by the addition of the profit expected from the blocks which belong to the upper cone of the blocks exploited before.

4.3 Algorithm

We introduces a GRASP algorithm which generates moves to obtain pre-solutions that satisfy all the constraints. Thus, the move is defined such as the algorithm obtains only feasible solutions. The most important constraints are the subsidence constraints. At the beginning, the algorithm selects randomly one block. A cell of this block will be extracted. Thus, as a consequence of its extraction the algorithm inhabilitates the other blocks which are in the upper cone of this block.

The blocks inhabilitated will be added in a Tabu List for ever. The next blocks feasible to be extracted will be the blocks that are not in the Tabu List, that means the blocks that satisfy the subsidence constraints. Each block belonging

to the Grasp-List is evaluated using the evaluation function. It takes into account the gain, the habilitation cost, the extraction cost and the opportunity cost. The opportunity cost is calculated using the gain of the blocks that will not be able to be exploited in the following extractions. Figure 2 shows the algorithm.

We use the idea of the roulette wheel from genetic algorithms to implement a selection strategy of a block from the Grasp-List. Obviously, this roulette is biased to the better evaluated blocks.

```

Begin /* procedure Grasp Algorithm */
iter=0
Repeat
Select randomly one block
k=1
Repeat
    Grasp-List = a set of the available blocks
    that satisfies subsidence constraints
    Evaluate the blocks in Grasp-List
    Construct a Roulette Wheel using the blocks
    in Grasp-List
    Select randomly using Roulette one block
    from Grasp-List
    k++
until k=max-blocks or Grasp-List = empty
iter++
until iter=max-tries
LocalSearch(solution)
end

```

Fig. 2. Grasp Algorithm for Macro Mine Planning Problem

```

LocalSearch(solution)
Begin
For k = 1 to max-tries-LS do
{
Swapped-solution = Swap( $b_i$ ,  $b_j$ , solution)
For each block  $b_1$  in Swapped-solution:
    For each block  $b_2$  after  $b_1$  in the extraction sequence:
        if ( $b_2 \in I(b_1)$ ):
            insert  $b_2$  before  $b_1$ .
        end if
    end for
end for
If f(Swapped-solution) > f(solution) then
solution=Swapped-solution
}
End

```

Fig. 3. Local Search Procedure

The procedure is repeated for a max-tries iterations. The solution obtained is taken by a Local Search procedure. Figure 3 shows the algorithm, where $b_2 \in I(b_1)$ means that b_2 belongs to the subsidence cone of block b_1 .

It does a hill-climbing procedure which tries doing swapping moves to improve the evaluation function for max-tries-LS times. In order to avoid the subsidence constraint violation it does the reparation procedure before accepts a candidate move. The procedure selects randomly two blocks b_i and b_j from the solution, given by the constructive procedure, for interchanging their positions. The new solution is called Swapped-solution. LocalSearch verifies the satisfaction of the subsidence constraints in the Swapped-solution. In case they are not satisfied a procedure is done to repair the violations.

5 Results

Because the information of our real mine is a confidential issue we are not able to report real results here. However, we have built a database of benchmarks¹, with 50 mines which have similar characteristics of a real one. The dimensions are in number of blocks in the three coordinates. The artificial mines have various kinds of copper concentration: at random, at the bottom, at the middle, at the borders, at the higher layers, both at bottom and in the middle. They have also different dimensions. We consider the following:

- A hard configuration: When the mine has many blocks with the same gain and not concentrated
- A normal configuration: When the mine has blocks that the algorithm can identify as the better ones to be extracted
- An easy configuration: When a mine has blocks with very different gain and the identification of the better ones is obvious.

Furthermore, some of these artificial mines are more complex than a real one. We report here the ten hardest mine configurations a_1, a_2, \dots, a_{10} . The common parameters of these virtual mines are:

$H = 20$	Time in years.
$K_y = 5$	the maximum number of blocks to be extracted by year
$C_k = US\$8.0M$	Access cost of block k
$r = 10\%$	Annual discount rate.

Our tests were made on an Athlon XP 1.6 GHz computer with 512MB of RAM, running Linux RedHat 7.2 and the GNU G++ compiler and optimizer.

The parameters values found by tuning are: max-tries = 100, max-tries-LS = 5. We have evaluated the behaviour of our algorithm using different sizes of the Grasp-List, $L = \{5, 7, 10\}$. We have tested three runs of the algorithm for each problem. The figure 4 shows for each problem a_i the values obtained for each run serie and its average.

¹ Available in <http://www.inf.utsm.cl/~bonnaire/Mines-Benchmarks>

Problem	Optimal	L=5	L=7	L=10
a_1	7454.6			
average		6173.5	6249.3	6140.85
a_2	7597.4			
average		6550.3	6676.5	6504.26
a_3	7597.4			
average		6823.8	6887.7	6740.26
a_4	7708.3			
average		6769.9	6900.2	6750.98
a_5	6493.1			
average		6410.9	6411.0	6130.23
a_6	9570			
average		5265.0	5308.5	5128.83
a_7	11319			
average		11262.3	11313.1	11236.5
a_8	10451.1			
average		10373.1	10342	10320
a_9	11030			
average		11007	11014	10876
a_{10}	9430.3			
average		8160.8	8130.3	7639.78

Fig. 4. Benchmarks with different sizes of Grasp-List

For these tests we know which are the optimal values. Our algorithm found, in average, a 20% bigger gain than the obtained by using the traditional approach which uses a relaxed model. In the table, we can see that for our algorithm the hardest one was a_6 . In this problem 75% of the blocks have the same gain, thus it is very difficult to discriminate which are the best to be extracted. We can observe that we obtain the best results using a Grasp-List of 7. We have also tested our algorithm with a real configuration, we have obtained a gain which is the best known for this problem. The algorithm uses a Grasp-List of size 7.

6 Conclusions

We have introduced a Grasp algorithm for solving a macro-mine planning problem. We shown that this kind of technique could be a good alternative to solve real world problems for which traditional techniques are not able to solve. In this work we have considered the mine divided into homogeneous blocks. Each block is composed by the same number of cells. However, it will be interesting to work with non-homogeneous structures, because it is closer to reality. Obviously non homogeneous blocks increase the complexity related to the subsidence constraints. We have obtained better results applying a Grasp-based technique than the complete approach that uses a relaxed model. Nevertheless, in order of being exact in the interpretation of the results, our algorithm uses a metaheuristic technique, thus is not able to obtain the optimal value and its performance depends strongly of the random number generator. We have also studied the

behaviour of the algorithm solving other kind of mine configurations, with more levels than the real one, and with some instances hardest than the real one. The algorithm shows be better using a Grasp-List of size equal to 7. Now, we are going towards to solve the Micro-sequence mine planning problem using other metaheuristics as genetic algorithms.

References

- [1] Burke, E.K., Smith, A.J.: Hybrid Evolutionary Techniques for the Maintenance Scheduling Problem. *IEEE Transactions on Power Systems* 15(1), 122–128 (2000)
- [2] Newall, J.P.: Hybrid Methods for Automated Timetabling, PhD Thesis, Department of Computer Science, University of Nottingham, UK (May 1999)
- [3] Taillard, E.: Heuristic Column Generation Method for the heterogenous VRP. *Recherche-Operationnelle* 33, 1–14 (1999)
- [4] Coloni, A., Dorigo, M., Maniezzo, V.: Metaheuristics for High-School Timetabling. *Computational Optimization and Applications* 9(3), 277–298 (1998)
- [5] Dorigo, M., Gambardella, L.M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation* 1(1), 53–66 (1997)
- [6] Karanta, I., Mikkola, T., Bounsaythip, C., Riff, M.-C.: Modeling Timber Collection for Wood Processing Industry. The case of ENSO, internal Technical Report, TTE1-2-98, VTT Information Technology, Information Systems, Finland (October 1998)
- [7] Tsang, E.P.K., Wang, C.J., Davenport, A., Voudouris, C., Lau, T.: A family of stochastic methods for constraint satisfaction and optimization. In: *The First International Conference on The Practical Application of Constraint Technologies and Logic Programming*, London, pp. 359–383 (1999)
- [8] Casagrande, N., Gambardella, L.M., Rizzoli, A.E.: Solving the vehicle routing problem for heating oil distribution using Ant Colony Optimisation. In: *ECCO XIV Conference of the European Chapter on Combinatorial Optimisation* (May 2001)
- [9] Riff, M.-C.: A network-based adaptive evolutionary algorithm for CSP. In: *The book Metaheuristics: Advances and Trends in Local Search Paradigms for Optimisation*, ch. 22, pp. 325–339. Kluwer Academic Publisher, Dordrecht (1998)
- [10] Breunig, M., Heyer, G., Perkhoff, A., Seewald, M.: An Expert System to Support Mine Planning Operations. In: Karagiannis, D. (ed.) *Proceedings of the International Conference on Database and Expert Systems Applications*, Berlin, Germany, pp. 293–298 (1991)
- [11] Feo, T., Resende, M.: A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters* 8, 67–71 (1989)
- [12] Resende, M., Ribeiro, C.: A GRASP and path-relinking for private virtual circuit routing. *Networks* 41, 104–114 (2003)
- [13] Resende, M., Ribeiro, C.: GRASP with path-relinking: Recent advances and applications. In: Ibaraki, T., Nonobe, K., Yagiura, M. (eds.) *Metaheuristics: Progress as Real Problem Solvers*, pp. 29–63. Kluwer, Dordrecht (2005)
- [14] Ricciardi, J., Chanda, E.: Optimising Life of Mine Production Schedules in Multiple Open Pit Mining Operations: A Study of Effects of Production Constraints on NPV. *Mineral Resources Engineering* 10(3), 301–314 (2001)

- [15] Gunn, E., Cunningham, B., Forrester, D.: Dynamic programming for mine capacity planning. In: Proceedings of the 23rd APCOM Symposium, Montreal, vol. 1, pp. 529–536 (1993)
- [16] Waltham, T., Waltham, A.: Foundations of Engineering Geology, 2nd edn. Routledge mot E F & N Spon (2002)
- [17] Maturana, J., Riff, M.-C.: An evolutionary algorithm to solve the Short-term Electrical Generation Scheduling Problem. *European Journal of Operational Research* 179(3), 677–691 (2007)
- [18] Solnon, C.: Ants Can Solve Constraint Satisfaction Problems. *IEEE Transactions on Evolutionary Computation* 6(4), 347–357 (2002)
- [19] Eiben, A.E., Van Hemert, J.I., Marchiori, E., Steenbeek, A.G.: Solving Binary Constraint Satisfaction Problems using Evolutionary Algorithms with an Adaptive Fitness Function. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, p. 201. Springer, Heidelberg (1998)

Food Wholesales Prediction: What Is Your Baseline?

Jorn Bakker and Mykola Pechenizkiy

Eindhoven University of Technology
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands
{j.bakker,m.pechenizkiy}@tue.nl

Abstract. Sales prediction is an important problem for different companies involved in manufacturing, logistics, marketing, wholesaling and retailing. Different approaches have been suggested for food sales forecasting. Several researchers, including the authors of this paper, reported on the advantage of one type of technique over the others for a particular set of products. In this paper we demonstrate that besides an already recognized challenge of building accurate predictive models, the evaluation procedures themselves should be considered more carefully. We give illustrative examples to show that e.g. popular *MAE* and *MSE* estimates can be intuitive with one type of product and rather misleading with the others. Furthermore, averaging errors across differently behaving products can be also counter intuitive. We introduce new ways to evaluate the performance of wholesales prediction and discuss their biases with respect to different error types.

1 Introduction

The success of different companies depends today on their ability to adapt quickly to the changes of their business environment. An accurate and timely sales prediction is particularly important for the companies involved in manufacturing, logistics, marketing, wholesaling, and retailing.

In the food and beverages market, food service companies often have to deal with short shelf-life products, and uncertainty and fluctuations in consumer demands. These variations in consumer demand may be impacted by the high number of factors including e.g. price change, promotions, changing consumer preferences, or weather changes [5]. Furthermore, a large share of the products sold in that market is sensitive to some form of seasonal change due to the different cultural habits, religious holidays, fasting, and alike. All these factors imply that some types of products are sold mostly during the limited period(s) of time.

Although it is known that some seasonal pattern is expected, the predictive features that define these season are not always directly observed. Therefore, drops and rises in sales which are accommodated by the changing seasons are often difficult to predict. Regarding inventory management, this results often in a stock-out at the start of the season and perishable or obsolete goods at the

end of a seasonal period. Thus, both shortage and surplus of goods can lead to loss of income for the company.

Time-series research has been traditionally suggesting ARIMA (autoregressive moving average) and ANN (artificial neural networks) approaches to address the problem of sales prediction. Despite of the continuous efforts devoted to come up with a right algorithm, and a number of comparative studies focused on identifying the strongest one, researchers are not clearly in favor of one particular method. Nonlinearity prevents the success of simple linear models, while rather short lengths of the time series are insufficient to learn more complex models [1]. It is rather intuitive that no single method is best in every situation and that combining different models might be an effective way to improve accuracy of (sales) prediction. Interestingly, both data mining and time series forecasting research pointed out into this promising direction [6] [4].

Anyhow, the challenge of building accurate predictive models has been already recognized among both researchers and practitioners. In this paper we reconsider the problem of evaluating the performance of time series forecasting and, particularly, food wholesales prediction and emphasize that this issue is also far from being trivial. Sales data typically comprises of many different products that exhibit very different types of behavior. Standard error measures like Mean Absolute Error (*MAE*) and Mean Squared Error (*MSE*) yield biased results when applied on the different types of time series. They can be intuitive with one type of product and rather misleading with the others. Naturally, we often want to compare the performance of several methods across a number of products (time series). This requires an error or accuracy performance measure to remain intuitive when aggregated over several datasets. Due to imbalances and structural differences this is not always possible (or not advisable). We compare the intuition behind the different popular error measures, discuss their limitations, and introduce new approaches and measures that may allow to get a better insight on the prediction performance.

2 Food Sales Prediction Evaluation

In this section we illustrate that not just the wholesales prediction but also the evaluation and comparison of different prediction techniques across various products (datasets) is not trivial and requires careful considerations. Let us illustrate first that traditional *MAE* or *MSE* like measures can be rather unintuitive because sales data typically comprises of many different products that exhibit very different types of behavior.

Consider wholesales figures for two products given in Fig. 1; *Product 1* has a lot of variation of (and no constant) demand whereas *Product 2* is periodic and shows constant demand between the peaks. Taking an error measure like the *MSE*, it would be easy to achieve a good performance on the highly periodic series (like with *Product 2*) by taking a naive predictor that just chooses the last observed value as the prediction for the next point, or always outputs the most popular value, i.e. the value corresponding to the constant demand in this case,

or computes a moving average. Thus, MSE of an optimal predictor will be close to MSE of a naive predictor that makes the comparison of MSE 's of different predictors meaningless. (Since from the domain perspective the peak demand is more important to predict than long lasting flat areas, we can see here also additional connections to the issues of class imbalance and one-class classification that are well-known in machine learning). It is not difficult to notice that MSE of the same naive predictor for the *Product 1* would lead to very bad results but e.g. a moving average approach would perform reasonably well, i.e. likely not worse than performance of any *learnable* predictor. Thus, if we try to aggregate the MSE 's over the two products, the average MSE 's will be misleading. Therefore, using the MSE is not preferable if we want the performance measure to yield a result that is intuitively comparable over all the time series in the data set.

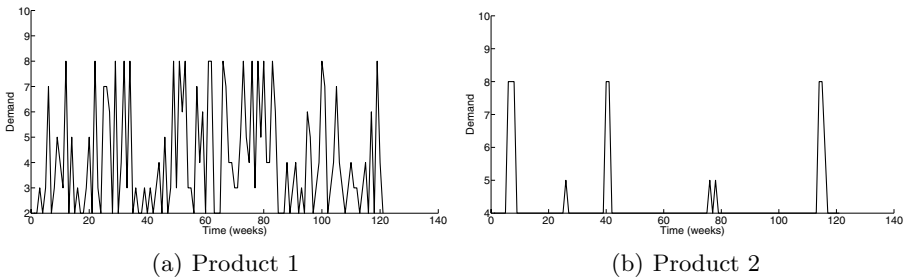


Fig. 1. The structural difference between two representative products

This claim with respect to the MSE can be generalized to any error measure that uses the unscaled prediction error. In order to address this issue, a scaled measure and a baseline that provides the reference scale for the performance measurement is needed. We will consider a couple of corresponding possibilities.

2.1 Error Measures

Error measures that have been proposed in the literature [2] and were commonly applied for evaluation of time series forecast include:

- Mean Squared Error: $MSE = \frac{1}{n} \sum_{t=1}^n e_t^2$,
- Root Mean Squared Error: $RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n e_t^2}$,
- Mean Absolute Error: $MAE = \frac{1}{n} \sum_{t=1}^n |e_t|$,
- Mean Absolute Percentage Error: $MAPE = \frac{1}{n} \sum_{t=1}^n |p_t|$,
- Mean Absolute Scaled Error: $MASE = \frac{1}{n} \sum_{t=1}^n \left| \frac{e_t}{MAE(Baseline)} \right|$,

where the error $e_t = y_t - f(t)$, y_t is the actual value, $f(t)$ is the predicted value by model f at time t , and the percentage error $p_t = \frac{e_t}{y_t}$.

Both *MSE* and *RMSE* are well known and widely used to validate the accuracy of a model. In the machine learning field these measures are used to evaluate the performance of a given algorithm. In the forecasting of time series, however, they are deemed not so suitable because of the aforementioned scaling differences and the sensitivity to outliers. We will present and discuss a scaled version of the *MSE*.

The *MAPE* has been recommended for measuring accuracy among many different time series. However, it should be noticed that in cases where y_t is very close to zero, the resulting *MAPE* will become infinite or invalid. The same holds for the *MASE*, i.e. in case the *MAE(Baseline)* is close to zero, but this case is special in the following sense. The *MASE* uses, in contrast with the other error measures, explicit scaling with respect to some baseline. Notice that if *MAE(Baseline)* is close to zero, the baseline itself is a good predictor. The advantage of *MASE* is that the accuracy of a given model can directly be related to the baseline regardless of scale.

2.2 Baseline Predictors

Selection of suitable baselines is important for identifying reference points which would allow comparing among different alternative techniques, but also to have a better understanding of how much worse (or better) a particular technique performs with respect to known optimal (or simply reasonably good) and worst (or clearly bad) cases.

Naive Prediction Baseline. The naive prediction baseline (“choose the last observed value as the prediction for the next point”) is a widely used baseline in forecasting methods. The intuition behind using this baseline is, that regardless of the accuracy of a given predictor it should always perform better than the naive prediction. Scaling towards the naive predictor does not have an upper bound. In our investigation we consider the *MAE* applied to the naive predictor (f_{naive}):

$$MAE(f_{naive}) = \frac{1}{n-1} \sum_{t=1}^n |y_t - y_{t-1}|. \quad (1)$$

Worst Case Prediction Baseline. The worst case scenario gives us an upper bound of poor performance and can be used to scale the error of different predictors between 0 and 1 and directly compare the predictive performance of different algorithms. This approach can only be used in cases where the maximal value can be computed. But, a priori, this baseline also suffers from a bias with respect to structurally different time series. In the new error measure that we introduce in the next subsection, the *MSE* of the worst case (f_{WC}) baseline is used:

$$f_{WC} = \frac{1}{n} \sum_{t=1}^n (y_t - \max_{i=1}^{\alpha} |i - y_t|)^2, \quad (2)$$

where α is the number of levels to which time series is approximated.

Sample Biased Evaluation. The bias of the f_{WC} prediction can be decreased by selecting “interesting” data points from test data. The “interesting” parts of the time series in Fig. 1 are not the long stretches of constant values, but the peaks. If only the peaks are taken into account in the accuracy calculation, the error estimate becomes more adequate from the domain point of view.

The selection of test data points to be considered in the accuracy calculation should be handled with care. The only points eligible to be deselected are points for which the following two properties hold:

- i) the last actual value y_{t-1} is equal to y_t , and
- ii) the error $e_t = 0$.

All other points are in the test data. In other words, this selection procedure selects everything except for the points that did not change in the recent past and have been predicted correctly by all the considered approaches. This approach is similar to the *MASE* with the important difference that it is scaled to an interval between 0 and 1:

$$f_{WCscaled} = \frac{MSE(f(\bar{t}))}{MSE(f_{WC}(\bar{t}))}, \quad (3)$$

where \bar{t} is the vector of selected points, $f(\bar{t})$ the output of the prediction model, and $f_{WC}(\bar{t})$ the worst case prediction. This approach has some similarities to computing misclassification error separately for the true positive class.

3 Experiment Design and Results

In order to demonstrate the characteristics of the aforementioned error measures we conducted experiments on a real wholesales data. In this section we present an overview of the experiment design, the results with respect to the error measures, and some additional tradeoffs in the evaluation of food sales prediction algorithms.

3.1 Experiment Design

For our study we selected several products provided by Sligro Food Group BV, which encompasses food retail and food service companies selling directly and indirectly to the entire Dutch food and beverages market and has about 60.000 products in stock. The products are selected in such a way that they represent different type of behavior (more seasonal vs. more chaotic, cf. Fig. 1) to demonstrate and investigate the bias of different accuracy measures in different types of time series.

Data Preprocessing. The data warehouse consists of all the transactions made in a period of over two years. For weekly predictions (which are most important

for wholesales), the resulting time series of aggregated transactional data contains 120 data points, from which the first 77 instances are used as the training set and the last 43 instances are used for progressive evaluation of the predictors.

Accumulated and aggregated transactional data was transformed with piece wise approximation to 8 levels that reflect the variation in sales from very low (1) to very high (8). Thus the data has a predefined upper bound and we can compute the error of the worst case.

Besides the standard time-series features like history of sales, moving averages, and slopes each data (time) point contain information about promotions, (school and public) holidays and weather which are known to impact the wholesales for certain types of products. A simple filter-based individual feature selection is used to address the problem of high dimensionality.

Learning Techniques. We experimented with three predictors: a moving average over a window of size 6 (*MA6*), a logistic regressor (*LR*), and an ensemble learning algorithm (*ENS*). The moving average, is a very basic predictor that is being used in practice to aid prediction of demand. The logistic regression, is a method that is commonly used in prediction problems. The ensemble learning, is known to be a promising approach for prediction in changing environments [3], and recent studies in time series forecasting and data mining have shown that combining different classifiers for sales prediction can lead to better results [6] [4].

3.2 Results

The results of the experiments are displayed in Table 1. For each prediction method and product we present the error estimates computed over the test (i.e. out-of-sample) data with different considered error measures. For all of the error measures in the table the smaller the value the accurate the predictor is.

The first thing to be observed is the difference between the *MSE*, *RMSE*, *MAE*, and *MAPE* and the results of *MASE* and $MSE(f_{WCscaled})$. For the first group of error measures it holds that the forecasting results on *Product 1* are worse than for *Product 2*. This is due to the fact that predicting the constant demand is easy for every considered technique. Not surprisingly, the *MASE* and $MSE(f_{WCscaled})$ send an opposite message. In the case of *MASE* all three predictors perform worse than the naive predictor in case of *Product 2* and better than the naive predictor in case of *Product 1*.

The second thing to note is the difference between *MASE* and $MSE(f_{WCscaled})$. While in the *MASE* case *MA6* performs worse than the *ENS*, the $MSE(f_{WCscaled})$ shows that the *MA6* performs better. Please notice that these two measures are on completely different scale, so a direct comparison is hard. In Fig. 2 we can see what kind of errors (i.e. difference between the true labels and predictions) different predictors make.

Apart from the $MSE(f_{WCscaled})$, each of the errors shown in Table 1 are unbounded. Since the $MSE(f_{WCscaled})$ is scaled between 0 and 1 with respect to the worst case policy, the values shown here can be considered as traditional misclassification errors.

Table 1. Performance of MA6, LR, and ENS on *Product 1* (P1) and *Product 2* (P2)

	Range	MA6		LR		ENS	
		P1	P2	P1	P2	P1	P2
MSE	0 .. ∞	3.79	1.09	6.47	1.05	5.88	1.23
RMSE	0 .. ∞	1.95	1.05	2.54	1.02	2.43	1.11
MAE	0 .. ∞	1.51	0.49	2.05	0.40	1.84	0.35
MAPE	0 .. 1	0.45	0.10	0.60	0.07	0.50	0.59
MASE (f_{naive})	0 .. ∞	0.93	2.28	0.93	2.17	0.90	1.63
MSE ($f_{WCscaled}$)	0 .. 1	0.14	0.16	0.22	0.05	0.20	0.18

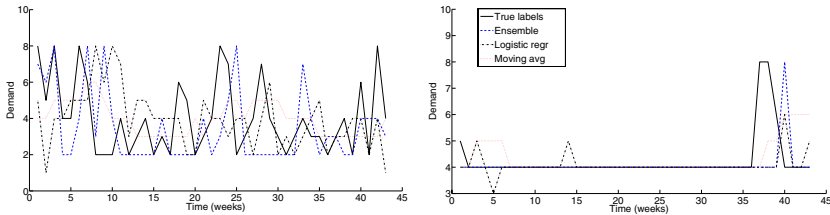


Fig. 2. The true and predicted sales for products from Fig. 1

Summary on the Considered Measures. Let us remind that our aim is to find a performance measure that enables aggregating the performance results over all data sets in a given database regardless of their structural differences. Any error measure that is not scaled cannot be used for this purpose. Compare, e.g., the outcome of the *MSE* (or *RMSE*) results between the two products in Table 1. For all algorithms that were tested, it holds that $MSE_{P1} > MSE_{P2}$, whereas the same is not true for the *MASE* and $MSE(f_{WCscaled})$ measure. The question is whether the *MASE* and $MSE(f_{WCscaled})$ measures are reliable enough in order to allow for cross product validation.

The *MASE* measure provides a direct way to compare the predictor to a meaningful baseline. When comparing the outcomes of the unscaled measures in Table 1 to the Fig. 2 for *Product 2*, it becomes immediately clear that something is wrong. Where the unscaled measures report a very low error, the *MASE* indicates that all of the algorithms perform worse than the naive predictor. However, since *MASE* is unbounded, it does not give a relative and normalized accuracy measure.

The $MSE(f_{WCscaled})$ measure provides an accuracy measure that is bounded. Since the measure is scaled towards the worst case predictor, it is only usable if the upper bound of the time series is known. In principle, this measure can be aggregated over all the different products in the database. What remains to be seen is whether the selection procedure is fair enough to provide an unbiased accuracy measure. If the amount of selected points in the test set is relatively low the measure can become biased because of the underlying *MSE* measure.

Both *MASE* and $MSE(f_{WCscaled})$ give a more accurate and intuitive performance measure than the traditional evaluation measures. The *MASE* is particularly useful in cases where ranking is used between different predictors because of its comparing nature. The $MSE(f_{WCscaled})$ gives a direct error measure on the prediction, but it assumes that a maximum value is known for the time series. This last assumption is not trivial in the context of data streams.

Other Biases in Predictions. In the domain of food sales predictions there are actually different *types* of errors with different impact on the performance. An overestimation of demand will have different impact on the outcome (application of) the prediction than an underestimation. Therefore, performance measures that take this into account seem natural in this context.

Comparing how often predictors forecast either too low or too high, might indicate a bias of each predictor towards certain type of error. In Fig. 3 the number of “misses” (estimated too low, i.e. points for which $y_t - f(t) < 0$) and “false alarms” (estimated too high, i.e. points for which $y_t - f(t) > 0$) are shown for a selection of products. Each of these points corresponds to a predictor, the *MA6* (dashed red circles) or *ENS* (solid black circles). Each pair of points corresponding to a certain product is connected via a line. We can observe that the products that have many flat parts are in the lower left corner, whereas the products having more chaotic behavior are in the upper right. We can also see that *MA6* always has higher number of misses, i.e. under predictions, but *ENS* for the majority of product has higher false alarm rates. Similar comparison of different predictors can be performed with respect to “too late” vs. “too early” or other types of errors.

In the field of food sales prediction an error might be less grave if the predicted amount is needed within some safety boundary. If a company overstocks at time t , it might be at some time $t + n$ demand is rising. If the time difference n is then small enough the stock might still get sold, resulting in a lower cost than predicted by the algorithm.

It is often important to know if the demand curve is close to the structural shape of the predicted curve. Dynamic Time Warping (DTW) can be applied to the demand curve and the predicted curve to find the distance reflecting the performance of the predictor (see Fig. 4). *ENS* has, apart from a few examples, a

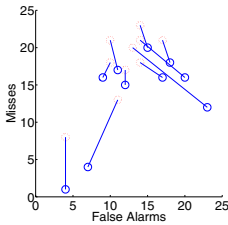


Fig. 3. Number of misses against the number of false alarms

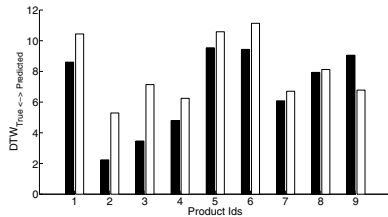


Fig. 4. DTW as accuracy measure; *ENS* (black), *MA6* (white)

clear advantage over *MA6* when it comes to structural differences. Fig. 5 shows actual DTW mappings for *Product 1* and *Product 2*. Please notice that assigning different costs to each of the aligning directions in DTW we can also introduce a desired bias.

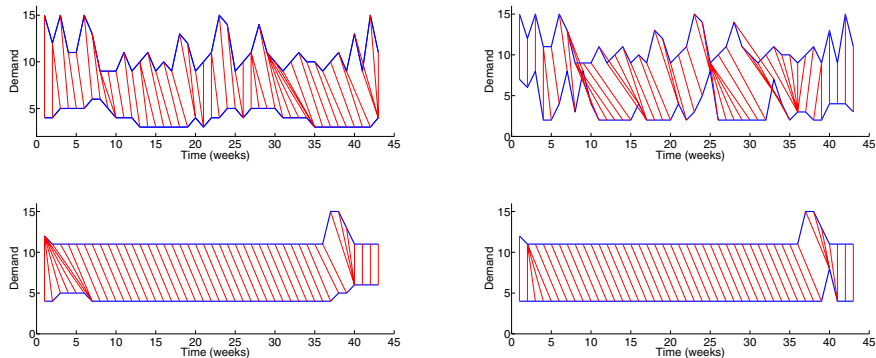


Fig. 5. Tolerating ‘wrong time’ prediction with nonlinear alignment of the true to the predicted labels for *Product 1* (top) and *Product 2* (bottom) for *MA6* (left) and *ENS* (right). For visualization purposes, the true label values are increased by 7.

4 Conclusions

Food sales prediction is an important and challenging problem having some connections to the problem of predicting in changing environments. In this paper we emphasized that besides this already recognized challenge, the problem of performance evaluation is also far from being trivial. Our previous experience showed that it was not always appropriate to use any of the suggested in the literature measures across different products within a business as a result of rather different behavior in sales, volume and supply characteristics. Here, we considered the different traditional ways of measuring the sales prediction accuracy (that is essential for monitoring and comparing the performance of employed approaches) and discussed and illustrated their limitation with real food sales data. Instead of averaging error estimates across products, someone may try to compare averaged ranks. However, with increasing number of learners to compare and yet questionable appropriateness of an error measure, the averaged rank can be also rather unstable and thus not informative.

In this paper we introduced and experimentally analyzed one new measure that does allow comparing performance of different predictors across different products with different types of time series structure. Beside this, we introduced and demonstrated the use of a generic approaches to measure other biases like optimistic vs. pessimistic and early vs. late prediction biases. We considered the use of the dynamic time warping distance as accuracy measure which may allow to prevent or to tolerate the certain types of errors.

Ultimately, the performance measure would be expressed in the form of a cost function (e.g. on the amount of money the company saves or loses by choosing a particular prediction strategy) that allows directly optimize various parameters with a cost-sensitive learning approach or multi-objective optimization. Our further work in this direction includes development of more generic cost-sensitive approach for evaluating foodsales prediction performance.

Acknowledgements. This research is supported by The Netherlands Organisation for Scientific Research NWO HaCDAIS project. We are thankful to Sligro Food Group BV for providing us with the data and domain knowledge.

References

1. Doganis, P., Alexandridis, A., Patrinos, P., Sarimveis, H.: Time series sales forecasting for short shelf-life food products based on artificial neural networks and evolutionary computing. *Journal of Food Engineering* 75(2), 196–204 (2006)
2. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. *International Journal of Forecasting* 22(4), 679–688 (2006)
3. Kuncheva, L.I.: Classifier ensembles for changing environments. In: Roli, F., Kittler, J., Windeatt, T. (eds.) *MCS 2004. LNCS*, vol. 3077, pp. 1–15. Springer, Heidelberg (2004)
4. Meulstee, P., Pechenizkiy, M.: Food sales prediction: if only it knew what we know. In: *ICDM Workshops*, pp. 134–143. IEEE Computer Society, Los Alamitos (2008)
5. van der Vorst, J., Beulens, A., de Wit, W., van Beek, P.: Supply chain management in food chains: improving performance by reducing uncertainty. *International Transactions in Operational Research* 5(6), 487–499 (1998)
6. Zhang, P.G.: Time series forecasting using a hybrid arima and neural network model. *Neurocomputing* 50, 159–175 (2003)

A Distributed Immunization Strategy Based on Autonomy-Oriented Computing

Jiming Liu^{1,2,*}, Chao Gao¹, and Ning Zhong^{1,3}

¹ International WIC Institute, Beijing University of Technology, Beijing, 100124

² Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, HK

jiming@comp.hkbu.edu.hk

³ Department of Life Science and Informatics, Maebashi Institute of Technology, Japan

Abstract. In recent years, immunization strategies have been developed for stopping epidemics in complex-network-like environments. So far, it remains to be difficult for the existing strategies to deal with distributed community networks, even though they are ubiquitous in the real world. In this paper, we propose a distributed immunization strategy based on the ideas of self-organization and positive feedback from Autonomy-Oriented Computing (AOC). The AOC-based strategy can effectively be applied to handle large-scale, dynamic networks. Our experimental results have shown that the autonomous entities deployed in this strategy can collectively find and immunize most of the highly-connected nodes in a network within just a few steps.

1 Introduction

With the growth of network applications, the threats of internet worms to network security have become increasingly serious. In Web intelligence [1,2,3], it is very important to design an effective and efficient mechanism to restrain virus propagation. The most popular strategy at present is network immunization that restrains virus propagation by immunizing a set of nodes. Currently, it is widely accepted that the best strategy is a global strategy, called targeted immunization [4,5]. The basic idea behind the targeted immunization strategy is to immunize the highest-degree nodes by ranking all nodes in a network. However, it would be practically impossible to rank all nodes in a large-scale, dynamic network due to the fact that the complete topology of the network (e.g., WWW) will not be given and the computational complexity for ranking would be too high (i.e., $O(N^2)$).

In order to obtain a tradeoff between efficiency and feasibility, some local strategies based on direct and/or indirect neighbors have been introduced, e.g., the acquaintance immunization [6,7] and the D-steps immunization [8]. The motivation for these local strategies is to avoid the need for knowing the global information. However, it remains to be difficult for them to deal with some real-world networks (e.g., P2P) that may contain decentralized community structures, since the behaviors of these strategies are fixed and those nodes with the second highest degree are ignored in their search scope.

* Prof. Jiming Liu is the corresponding author of this paper.

Although the D-steps immunization is suited to distributed networks, it is computationally too costly. Furthermore, it would be impractical to know other indirect neighbors, i.e., one's neighbors' neighbors, in the real-world situation.

Recently, the aim of an immunization strategy has become to achieve the efficiency (i.e., coverage rate, as to be defined later) of the targeted immunization in a distributed and dynamic network, i.e., to vaccinate the minimum number of highly-connected nodes in a decentralized network, in order to cut an epidemic path without the global knowledge. Thus, the problem of network immunization can be translated into that of distributed combinatorial optimization [10], i.e., *to obtain a set of nodes with the highest-degree in a decentralized network*. This paper presents a distributed immunization strategy by extending the distributed constrained heuristic search [11] to an AOC-based system [12, 17]. The aim of the AOC-based immunization strategy is to search the highest-degree nodes based on the local information in a decentralized network.

As a novel computing paradigm, Autonomy-Oriented Computing (AOC) [12, 13] is well suited to solving computationally hard problems [14] and to characterizing the mechanisms of complex systems [15, 16] by utilizing the notion of local autonomy and self-organization. It deploys a group of computational entities into a distributed environment. Thereafter, they autonomously interact with each other and update their local environments. Each entity aims to reside in the highly-connected node based on its own profit and the positive feedback from its local environment. Our experiments have shown that the AOC-based strategy can find most of the highest-degree nodes within a few steps, which is quite useful for providing an approximate solution given a hard deadline. We have simulated virus propagation in an interactive email model [18], which indicates that the AOC-based strategy can efficiently restrain virus propagation.

2 Problem Statements

In our work, a network (e.g., computer network or social network) is represented as a graph $G = \langle V, L \rangle$, where $V = \{v_1, v_2, \dots, v_N\}$ is the set of nodes and $L = \{ \langle v_i, v_j \rangle \mid 1 \leq i, j \leq N, i \neq j \}$ is the set of links. $N = |V|$ represents the total number of nodes in the network and $\langle v_i, v_j \rangle$ means there is an edge between v_i and v_j .

The general definition about immunization strategy in G has been given as follows.

Definition 1. *The immunization strategy refers to a scheme for restraining virus propagation by means of immunizing a set of nodes V_e , which is denoted as $V_e = IS(V_0, G)$, where $V_0 \subseteq V$ is the initial set of "seed" nodes that are the initial positions for the autonomous entities. The output V_e is the final positions of entities, i.e., a set of highest-degree nodes to be immunized.*

At present, the best strategy, i.e., the targeted immunization, will select the largest V_e nodes by ranking all of them. Therefore, here we will take the total degree of immunized nodes as a benchmark, and define our evaluation measure as follows.

Definition 2. *The coverage rate of different strategies can be defined as*

$$\text{coverage_rate} = \frac{\sum_{v_i \in V_e} v_i \cdot \text{degree}}{\sum_{v_j \in V_e} v_j \cdot \text{degree}} \quad (1)$$

where v_i and v_j are immunized by other strategies and the targeted immunization strategy, respectively.

There are three research questions to be addressed in evaluating different immunization strategies, i.e.,

1. **Coverage Efficiency:** Does the immunization strategy have the ability to search and find all the highest-degree nodes in a decentralized network?
2. **Computational Complexity:** How effective is an immunization strategy in finding an optimal solution? How quickly can such a desired solution be found? Note that this corresponds to the computational cost, i.e., sequential operations (the total CPU runtime) and distributed operations (the steps or cycles).
3. **Immunization Efficiency:** Can the immunized nodes that are selected by an immunization strategy effectively restrain virus propagation? The immunization efficiency can be evaluated based on the total number of infected nodes, e.g., here in the interactive email model, when the propagation reaches a stable state.

As mentioned above, the existing immunization strategies are ineffective in finding the highest-degree nodes in a large-scale decentralized network. In what follows, we propose a distributed strategy, called AOC-based strategy, for tackling this problem. We allocate N_e autonomous entities as the “seed” nodes, in order for them to search the highest-degree nodes based on local information.

3 The Formulation of the AOC-Based Strategy

In our AOC-based strategy, the “seed” nodes correspond to the initial positions of the deployed autonomous entities. In order to overcome the shortcomings of the existing immunization strategies and to find the highly-connected nodes in a distributed and/or dynamic network, the autonomous entities response to the changes of their local environments by selecting their local behaviors. Based on the shared local environment, the entities can realize indirect interactions between each other. More importantly, indirect interactions among entities provide a mechanism for *positive feedback* that will accelerate the convergence of the whole system.

This section presents the basic elements in an AOC-based system, i.e., the autonomous entity and its local environment, local behaviors and behavioral rules, and evaluation functions [12,17].

3.1 Definitions of an Entity and Its Local Environment

In the AOC-based strategy, a set of autonomous entities are deployed in a network (Graph G). They move and reside in the network based on their own local information. When the dynamic entities converge to a stable state, the nodes that are resided by the entities are the immunized nodes.

Definition 3. Let e be the entity in the network G . The profile of entity e includes $\langle id, nodeId, utility, lifecycle, rules \rangle$, where id denotes the identifier of an entity. $nodeId$ and $utility$ represent the identifier and the degree of a node resided by e , respectively. $lifecycle$ is the times for the entity to reside in one node. The rules set stores some local interaction rules, including Rational-move, Random-jump and Wait.

The autonomous entities can only sense their own local environment that includes the direct neighbors and indirect neighbors. The direct neighbors are composed of the nodes that have edges connecting to the resided node. And, the indirect neighbors are the highest-degree nodes that are encountered by entities.

Definition 4. *The local environment of an entity is defined as E_l , which is a set of direct and indirect neighbors of e . If entity e resides in node v_i , the local environment is $E_l(e) = \{v_j \vee v_j.friendId | < v_i, v_j > \in L\}$, where *friendId* stores the *nodeId* of the highest-degree node that is visited by one entity on its traveling. If no entity visits v_j and its direct neighbors, the *friendId* of v_j becomes null.*

The profit in a local environment of entity e is denoted as $\delta(e)$, which corresponds to the degree of a node that has the highest-degree in the local environment of e , i.e., $\delta(e) = Max.degree(E_l(e))$.

The local environment E_l is not a static set of nodes. Even if an entity e resides in one node, its local environment $E_l(e)$ can be modified by other entities. Through this means, the entities can realize the indirect interaction and thus achieve the *positive feedback* from their local environment.

3.2 Local Reactive Behaviors

The goal of an entity is to reside in the highest-degree node of a network without ranking all the nodes. The entities with higher utilities will reside in the present places, and those with lower utilities will move to the nodes whose degrees are higher than current one based on the local information and feedback from the environment. Meanwhile, the random long-jump will enable an entity to escape out of local optima.

1. **Rational-Move.** is the best strategy for an autonomous entity to move to a nearby node with the maximum degree. If there exists more than one highest-degree position, the entity will choose the first one from its friend list. Based on the Rational-move, the entity implicitly incorporates the influence of a global solution, i.e., the entities reside in the highest-degree nodes in the network. More importantly, this *implicit influence* (instead of directly dictating) is the *sufficient condition for self-organized computability* [17].

Rule 1: If $e.utility < \delta(e)$, then $e.MoveTo()$.

2. **Random-Jump.** not only helps the autonomous entity jump out of local optima, but also is a means for realizing the diversity of solution and emergence of computable configurations, which are the *necessary conditions for self-organized computability* [17]. Just like the random-walk in the local search, the random long-jump keeps the whole system from falling into local optima, especially in a decentralized community network.

Rule 2: If $e.utility > \delta(e)$ and $e.lifecycle < 1$, then $e.DumpToNode()$.

Besides the above two moving behaviors, the entity can also stay at the old place (a Wait behavior), if it cannot find other better places than the current one. Meanwhile, in order to effectively apply its behavioral rules, an entity needs to evaluate its current state and utility, as well as the current states of its local environment, by using an evaluation function [17].

3.3 Evaluation Functions

Each entity can only sense its local environment and apply three behavioral rules to govern its degree-selection moves. The entity uses rational behaviors to improve its degree and obtains a higher utility. The utility of an entity can be estimated iteratively by the following equation:

$$e_i.utility(t + 1) = \begin{cases} \delta_l(e_i(t)), & \text{if } e_i.utility \leq \delta_l(e_i(t)) \\ n_j.degree, & \text{if } e_i.lifecycle < 1 \text{ and } e_i.utility > \delta_l(e_i(t)) \\ e_i.utility(t), & \text{if } e_i.lifecycle \geq 1 \end{cases} \quad (2)$$

Based on Eq. (2), it can be noted that there exists a *positive feedback* mechanism between the entity and its environment. This formula contains three parts: three different results of behaviors, i.e., the results of Rational-move behavior, Random-jump behavior, and Wait behavior. The positive feedback based on a shared environment among entities is illustrated in Fig 1

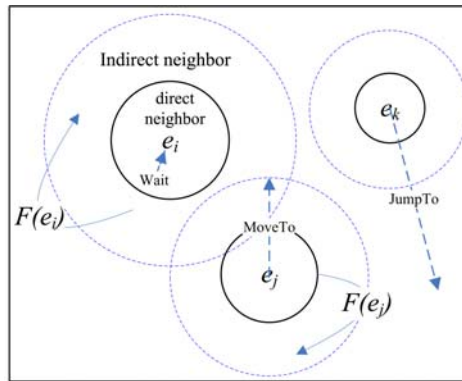


Fig. 1. The process of *positive feedback* between an entity and its environment. The dashed circle indicates the indirect neighbors (i.e., a set of $v_i.friendId$) and solid circle denotes the direct neighbors (i.e., a set of v_i). Each entity uses evaluation function F to update its own profiles and evaluate its local environment. The overlapping part between e_i and e_j denotes the indirect interaction. This indirect interaction can cause positive feedback and speed up the whole system to evolve to a desirable state.

At the end of each cycle, a local environment evaluation is performed that computes the local utility of e_i at t time, i.e., $\delta(e_i(t))$, based on the following:

$$\delta(e_i(t + 1)) = Max\{M(E_l(e_i(t + 1))), e_i.utility(t)\} \quad (3)$$

Meanwhile, the entity can improve the profit in its local environment by updating the friendId of the neighbors. The friendId stores the nodeId of a maximum degree node that is visited by one entity in the network. At the same time, the probability of selecting

Rational-move behavior can be indirectly affected by the increased profit of the local environment:

$$n_j.friendId = \operatorname{argmax}\{E_l(e_i(t+1)), e_i.nodeId\}, n_j \in E_l(e_i(t+1)) \quad (4)$$

where $\operatorname{argmax}\{\cdot\}$ returns a nodeId that has the maximum degree in the sets of $E_l(e_i(t+1))$ and $e_i.nodeId(t)$. For example, when entity e moves to v_i , the value of $v_j.friendId, \langle v_i, v_j \rangle \subseteq L$, is the maximum value between the direct neighbors of v_i and the node with the maximum degree that has been visited by entity e .

4 Experimental Validation

In this section, we provide several experiments to compare the AOC-based strategy with the D-steps strategy with respect to their coverage rate, efficiency, computational cost, and immunization efficiency, as mentioned in Section 2. We have downloaded and used some real benchmark networks, including the Enron email network¹, the AS network²(the internet at the level of autonomous system) and the coauthorship network³. All the numerical results are given in average values by simulating 100 times.

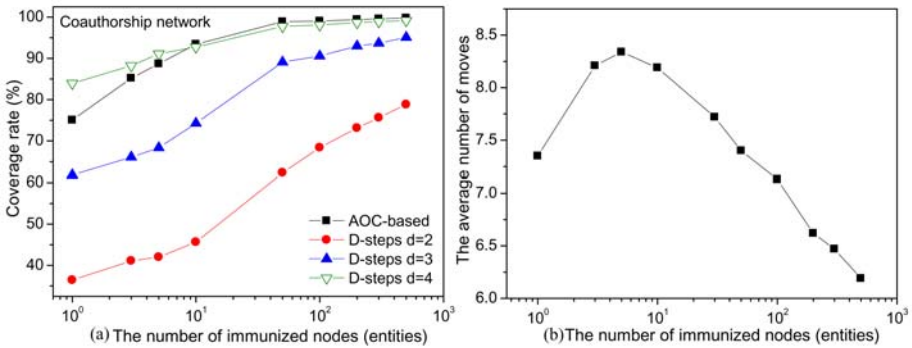


Fig. 2. The coverage rate and the average number of moves on the coauthorship network

4.1 Coverage Efficiency of Different Strategies

The comparison results of coverage rate in the coauthorship network are shown in Fig 2. Only when an adequate number of entities is reached, the AOC-based strategy can obtain better results based on the indirect interaction among entities, i.e., the emergent computing. Therefore, there is a point of inflection in Fig 2(b). The same situation happened in the AS network and Enron network. The comparison results in Fig 3 show that the coverage rate of the AOC-based strategy is very high. This means that our strategy can find most of highly-connected nodes in the network with a small number of moves.

¹ <http://bailando.sims.berkeley.edu/enron>

² <http://routeviews.org/>

³ <http://www-personal.umich.edu/~mejn/netdata/netscience.zip>

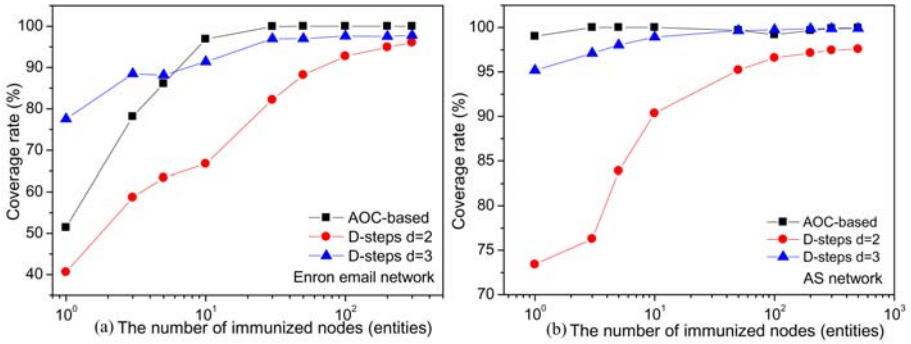


Fig. 3. The coverage rate on the Enron email network and AS network

4.2 Computational Complexity

In this section, we will further examine the worst-case computational complexities of the D-steps strategy and the AOC-based strategy. Because it is difficult to know how many neighbors one node has, the network degree ($\langle K \rangle$) is used to replace the average number of neighbors. The *D-steps immunization* selects the node with the highest degree around seed nodes in d steps. When $d=1$, the computational complexity is $O(n * (\langle K \rangle^2 + 1))$. When $d > 1$, each step will increase $n * \langle K \rangle$ nodes. Thus, the computational complexity is $O(n \langle K \rangle^{d-1} (\langle K \rangle^2 + 1))$.

The *AOC-based immunization* selects one node with the highest degree from its local environment based on Rational-move. And the long-range move randomly selects one node in the network. At worst, assuming that the whole environment needs \bar{p} times of updating to converge to a globally stable state, the computational complexity is $O(n * \bar{p} * (\langle K \rangle^2 + 1))$.

In order to validate our analysis, two large-scale benchmark networks (the AS network with 22963 nodes and the coauthorship network with 56276 nodes) are used to

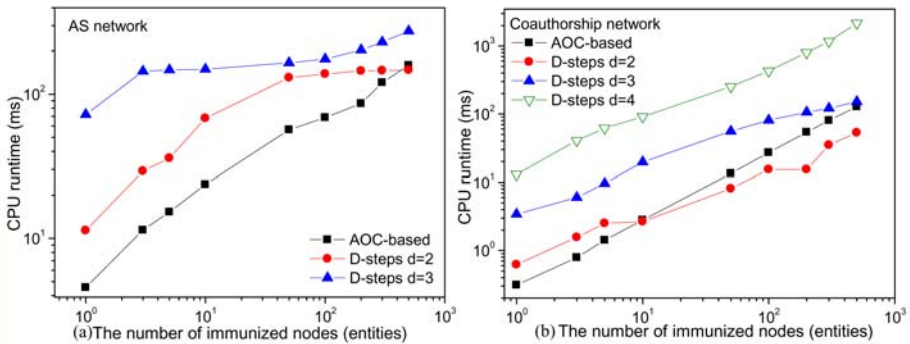


Fig. 4. The sequential cost with respect to the number of immunized nodes on the AS network and coauthorship network

compare the sequential time of different strategies which corresponds to the runtime as required by the sequential implementations (simulations) of the distributed algorithms. Our previous experiments shows that the coverage rate of the AOC-based strategy will slightly increase after updating 50 times. Therefore, we measures the sequential time for updating 50 times. The comparison results in Fig 4 show that the sequential time of the AOC-based strategy is shorter than that of others in most of the cases. That is because the direct computations among local neighbors can reduce the complexity of the AOC-based strategy.

Based on the above analysis of computational complexity, we know that $\langle K \rangle^{d-1}$ and \bar{p} will both determine the cost of the AOC-based strategy. The average degree of the AS network and coauthorship network are 4 and 11, respectively. And the system updating time \bar{p} is equal to 50. In this case, the runtime of the AOC-based strategy is more than the D-steps. However, the above analysis is based on the worst case. In the actual operations, the computing time of each entity is less than \bar{p} . Fig 4 shows that the cost of the AOC-based is lower than that of the D-steps (even if $d=2$ in the coauthorship network) at the initial stage. That is because the actual number of moves by each entity (entity steps) is fewer than \bar{p} (shown in Fig 2(b)) and most of the entities reside in their old positions.

4.3 Immunization Efficiency

We validate the immunization efficiency in an interactive email model [18], in which virus is triggered by the human behaviors, i.e, checking email intervals and the probability of opening suspectable emails, not the contact probability in the epidemic model [9].

Given the same number of immunized nodes, the criterion used to evaluate the efficiency of different immunization strategies is the final infected nodes in the network.

Figure 5(a) shows that there are the least infected nodes in the network based on the AOC-based strategy if selecting the same number of immunized nodes. Figure 5(b) is a propagation snapshot when there are 30 immunized nodes in the network. The result

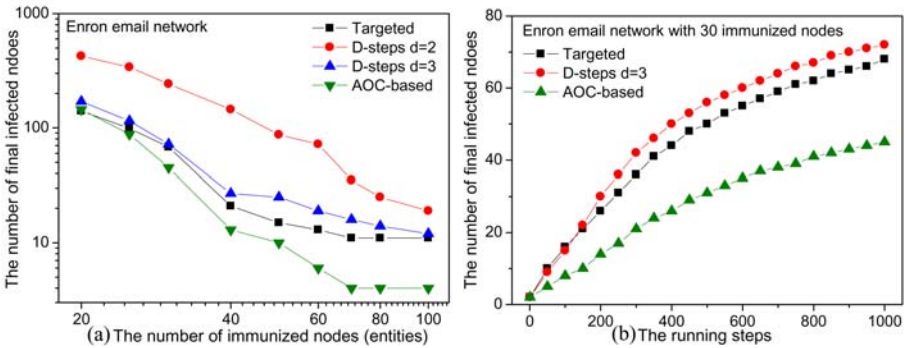


Fig. 5. The virus propagation in the Enron email network. (a) The change of the final infected nodes with the number of immunized nodes. (b) The final infected nodes with respect to the runtime when there are 30 immunized nodes.

shows that the efficiency of the AOC-based strategy is even better than the targeted immunization strategy sometimes. That is because the targeted immunization strategy selects immunized nodes by sorting and does not distinguish the importance of nodes. For example, if the nodes in the 40th to 60th are with the same degree after sorting, and the whole system need select 50 immunized nodes. The targeted immunization strategy will select 10 nodes from 20 nodes in the front part by index. However, the AOC-based strategy will select all immunized nodes along the edges and the autonomous entities aim to move to the highly-connected nodes in the network. This process is similar with the virus propagation. In most cases, the selected immunized nodes are frequently used in the network communication. Although the total immunized degree of the targeted immunization strategy is much larger than that of the AOC-based strategy, the importance of immunized nodes in the network communication is less than that of the AOC-based strategy.

5 Conclusions

This paper has presented an AOC-based, distributed immunization strategy [12,13,14,15,17]. With the AOC-based strategy, entities can find the highest-degree nodes in a relatively few steps. Furthermore, the efficiency of this strategy has been validated by means of measuring the number of final infected nodes, as compared to the targeted immunization strategy and the D-steps immunization strategy. The immunized nodes selected by the AOC-based strategy are not only with a maximal degree, but also frequently used in the network communication. Therefore, we conclude that the AOC-based immunization strategy can more effectively restrain the virus propagation.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China Grant (60673015), the Doctoral Student Innovation Program of Beijing University of Technology (bcx-2009-062), Open Foundation of Key Laboratory of Multimedia and Intelligent Software (Beijing University of Technology), Beijing, the Major State Basic Research Development Program of China (973 Program) (2003CB317001), and the Research Grant Council Central Allocation Fund of the Hong Kong SAR Government (HKBU 1/05C).

References

1. Zhong, N., Liu, J., Yao, Y.Y.: In Search of the Wisdom Web. *IEEE Computer* 35(11), 27–31 (2002)
2. Liu, J.: Web Intelligence (WI): What Makes Wisdom Web (Invited Talk). In: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003), pp. 1596–1601 (2003)
3. Zhong, N., Liu, J., Yao, Y.Y.: Envisioning Intelligent Information Technologies (iT) From the Stand-Point of Web Intelligence (WI). *Communications of the ACM* 50(3), 89–94 (2007)

4. Dezsó, Z., Barabási, A.-L.: Halting Viruses in Scale-Free Networks. *Physical Review E* 65, 055103 (2002)
5. Chen, Y., Paul, G., Havlin, S., Liljeros, F., Stanley, H.E.: Finding a Better Immunization Strategy. *Physical Review Letter* 101, 058701 (2008)
6. Cohen, R., Havlin, S., Ben-Avraham, D.: Efficient Immunization Strategies for Computer Networks and Populations. *Physical Review Letter* 91, 247901 (2003)
7. Gallos, L.K., Liljeros, F., Argyrakis, P., Bunde, A., Havlin, S.: Improving Immunization Strategies. *Physical Review Letter* 75, 045104 (2007)
8. Echenique, P., Gomez-Gardenes, J., Moreno, Y., Vazquez, A.: Distance- d Covering Problem in Scale-Free Networks with Degree Correlation. *Physical Review E* 71, 035102 (2005)
9. Pastor-Satorras, R., Vespignani, A.: Epidemic Spreading in Scale-Free Networks. *Physical Review Letters* 86(14), 3200–3203 (2001)
10. Sandholm, T.W., Lesser, V.R.: Coalitions among Computationally Bounded Agents. *Artificial Intelligence* 94(1-2), 99–137 (1997)
11. Sycara, K., Roth, S., Sadeh, N., Fox, M.: Distributed Constrained Heuristic Search. *IEEE Transactions on System, Man and Cybernetics* 21(6), 1446–1461 (1991)
12. Liu, J., Jin, X., Tsui, K.C.: *Autonomy Oriented Computing (AOC): From Problem Solving to Complex Systems Modeling*. Springer, Heidelberg (2005)
13. Liu, J., Jin, X., Tsui, K.C.: *Autonomy Oriented Computing (AOC): Formulating Computational Systems with Autonomous Components*. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans* 35(6), 879–902 (2005)
14. Liu, J., Han, J., Tang, Y.Y.: Multi-Agent Oriented Constraint Satisfaction. *Artificial Intelligence* 136(1), 101–144 (2002)
15. Liu, J., Tsui, K.C.: Toward Nature-Inspired Computing. *Communications of the ACM* 49(10), 59–64 (2006)
16. Liu, J., Zhang, S., Yang, J.: Characterizing Web Usage Regularities with Information Foraging Agents. *IEEE Transactions on Knowledge and Data Engineering* 16(5), 566–584 (2004)
17. Liu, J.: *Autonomy Oriented Computing (AOC): The Nature and Implications of a Paradigm for Self-Organized Computing (Keynote Talk)*. In: *Proceedings of the 4th International Conference on Natural Computation (ICNC 2008) and the 5th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2008)*, pp. 3–11 (2008)
18. Zou, C.C., Towsley, D., Gong, W.: Modeling and Simulation Study of the Propagation and Defense of Internet E-mail Worms. *IEEE Transaction on Dependable and Secure Computing* 4(2), 105–118 (2007)

Discovering Relevant Cross-Graph Cliques in Dynamic Networks

Loïc Cerf, Tran Bao Nhan Nguyen, and Jean-François Boulicaut

Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205, F-69621, France

Abstract. Several algorithms, namely CUBEMINER, TRIAS, and DATA-PEELER, have been recently proposed to mine closed patterns in ternary relations. We consider here the specific context where a ternary relation denotes the value of a graph adjacency matrix at different timestamps. Then, we discuss the constraint-based extraction of patterns in such dynamic graphs. We formalize the concept of δ -contiguous closed 3-clique and we discuss the availability of a complete algorithm for mining them. It is based on a specialization of the enumeration strategy implemented in DATA-PEELER. Indeed, clique relevancy can be specified by means of a conjunction of constraints which can be efficiently exploited. The added-value of our strategy is assessed on a real dataset about a public bicycle renting system. The raw data encode the relationships between the renting stations during one year. The extracted δ -contiguous closed 3-cliques are shown to be consistent with our domain knowledge on the considered city.

1 Introduction

Constraint-based mining is a popular framework for supporting pattern discovery. First, it provides more interesting patterns when the analyst specifies his/her subjective interestingness by means of a combination of primitive constraints. Then, this is known as a key issue to achieve efficiency and tractability: useful constraints can be deeply pushed into the extraction process such that it is possible to get complete (every pattern which satisfies the user-defined constraint is computed) though efficient algorithms.

In this paper, we focus on closed patterns that hold in ternary relations, i.e., Boolean cubes. They are a straightforward extension of the so-called formal concepts that hold in binary relations (see, e.g., [1]). This pattern domain has been extensively studied and efficient algorithms are available for mining constrained formal concepts. The extension of formal concept discovery to closed patterns in ternary relations has given rise to three proposals, namely CUBEMINER [2], TRIAS [3], and DATA-PEELER [4,5]. The main challenge of constraint-based closed pattern mining in 3-ary relations relies on the ability to push constraints during the extraction. To the best of our knowledge, only DATA-PEELER can mine closed patterns under a large class of constraints called *piecewise (anti)-monotonic constraints* [4,5]. This paper details how such constraints enable a

specialization of DATA-PEELER to the discovery of relevant patterns from dynamic graphs (e.g., dynamic interaction networks or dynamic co-interest graphs), considered as “stacks” of adjacency matrices, i.e., Boolean cubes.

Graph mining is a popular topic. Many researchers consider data analysis techniques for one large graph while others focus on graph pattern discovery from large collections of graphs. This article belongs to the latter family. More precisely, it deals with sets of timestamped graphs, i.e., dynamic networks. Our patterns are called δ -contiguous closed 3-cliques. Informally, they are maximal sets of vertices that are linked to each others and that run along some “almost” contiguous timestamped graphs. We show that this can be specified by means of primitive constraints which turn out to be piecewise (anti)-monotonic. As a result, the DATA-PEELER enumeration strategy can be specialized to support the search for knowledge nuggets from dynamic graphs, i.e., clique patterns that satisfy a wide range of user-defined constraints. Notice that such a formalization is new and has not been studied earlier. Due to the lack of space, we cannot recall the principles of the DATA-PEELER algorithm and we assume the reader can access to [4,5] for details.

In Sec. 2, we provide the problem setting where the mining task breaks into the satisfiability of a conjunction of constraints. They are proved piecewise (anti)-monotonic, i.e., they can be efficiently enforced by DATA-PEELER. Section 3 provides an experimental validation on a real dataset. Related work is discussed in Sec. 4, and Sec. 5 briefly concludes.

2 Extracting δ -Contiguous Closed 3-Cliques

2.1 Preliminaries

Let $\mathcal{T} \in \mathbb{R}^{|\mathcal{T}|}$ a finite set of timestamps. Let \mathcal{N} a set of nodes. A (possibly directed) graph is uniquely defined by its adjacency matrix $A \in \{0,1\}^{\mathcal{N} \times \mathcal{N}}$. A dynamic graph involving the nodes of \mathcal{N} along \mathcal{T} is uniquely defined by the $|\mathcal{T}|$ -tuple $(A_t)_{t \in \mathcal{T}}$ gathering the adjacency matrices of the graph at every timestamp $t \in \mathcal{T}$. Visually, such a stack of adjacency matrices can be seen as a $|\mathcal{T}| \times |\mathcal{N}| \times |\mathcal{N}|$ cube of 0/1 values. We write $a_{t,n^1,n^2} = 1$ (resp. $a_{t,n^1,n^2} = 0$) when, at the timestamp t , a link from n^1 to n^2 is present (resp. absent).

Example 1. Figure 1 depicts a dynamic directed graph involving four nodes a , b , c and d . Four snapshots of this graph are available at timestamps 0, 0.5, 2 and 3. Table 1 gives the related 4-tuple $(A_0, A_{0.5}, A_2, A_3)$.

Visually, a closed 3-set $(T, N^1, N^2) \in 2^{\mathcal{T}} \times 2^{\mathcal{N}} \times 2^{\mathcal{N}}$ appears as a combinatorial sub-cube of the data (modulo arbitrary permutations on any dimension) verifying both the *connection* and the *closedness* primitive constraints. Informally, it means that $T \times N^1 \times N^2$ only contains ‘1’ values (connection), and any “super-cube” of (T, N^1, N^2) violates the connection constraint (closedness).

Definition 1 ($\mathcal{C}_{\text{connected}}$). *It is said that a 3-set (T, N^1, N^2) is connected, denoted $\mathcal{C}_{\text{connected}}((T, N^1, N^2))$, iff $\forall (t, n^1, n^2) \in T \times N^1 \times N^2, a_{t,n^1,n^2} = 1$.*

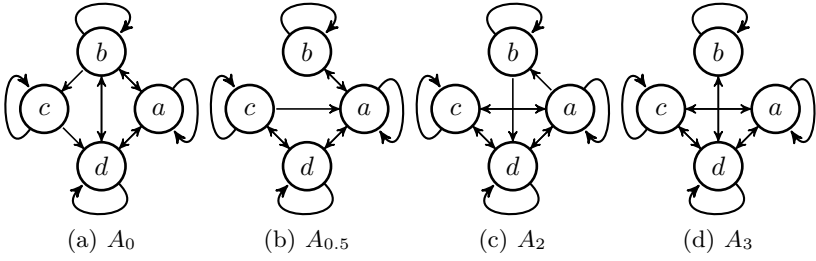


Fig. 1. Example of a dynamic directed graph ($\mathcal{N} = \{a, b, c, d\}$, $\mathcal{T} = \{0, 0.5, 2, 3\}$)

Table 1. ($A_0, A_{0.5}, A_2, A_3$) related to the dynamic graph depicted Fig. 1

	a	b	c	d	a	b	c	d	a	b	c	d	a	b	c	d
a	1	1	0	1	1	1	0	1	1	1	1	1	1	0	1	1
b	1	1	1	1	1	1	0	0	0	1	0	1	0	1	0	1
c	0	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
d	1	1	0	1	1	0	1	1	1	0	1	1	1	1	1	1
	A_0				$A_{0.5}$				A_2				A_3			

Definition 2 ($\mathcal{C}_{\text{closed}}$). It is said that a 3-set (T, N^1, N^2) is closed, denoted

$$\mathcal{C}_{\text{closed}}((T, N^1, N^2)), \text{ iff } \begin{cases} \forall t \in \mathcal{T} \setminus T, \neg \mathcal{C}_{\text{connected}}(\{t\}, N^1, N^2) \\ \forall n^1 \in \mathcal{N} \setminus N^1, \neg \mathcal{C}_{\text{connected}}(T, \{n^1\}, N^2) \\ \forall n^2 \in \mathcal{N} \setminus N^2, \neg \mathcal{C}_{\text{connected}}(T, N^1, \{n^2\}) \end{cases} .$$

Definition 3 (Closed 3-set). It is said that (T, N^1, N^2) is a closed 3-set iff it satisfies the conjunction $\mathcal{C}_{\text{connected}}((T, N^1, N^2)) \wedge \mathcal{C}_{\text{closed}}((T, N^1, N^2))$.

Example 2. $(\{0, 2, 3\}, \{a, b, c, d\}, \{d\})$ is a closed 3-set in the toy dataset from Tab. 1. $\forall (t, n^1, n^2) \in \{0, 2, 3\} \times \{a, b, c, d\} \times \{d\}$, we have $a_{t, n^1, n^2} = 1$, and

$$\begin{cases} \forall t \in \{0.5\}, \neg \mathcal{C}_{\text{connected}}(\{t\}, \{a, b, c, d\}, \{d\}) \\ \forall n^1 \in \emptyset, \neg \mathcal{C}_{\text{connected}}(\{0, 2, 3\}, \{n^1\}, \{d\}) \\ \forall n^2 \in \{a, b, c\}, \neg \mathcal{C}_{\text{connected}}(\{0, 2, 3\}, \{a, b, c, d\}, \{n^2\}) \end{cases} .$$

$(\{2, 3\}, \{a, c, d\}, \{a, c, d\})$ and $(\{0, 3\}, \{b, d\}, \{b, d\})$ are two other closed 3-sets. $(\{0.5, 2, 3\}, \{c, d\}, \{c, d\})$ is not a closed 3-set because it violates $\mathcal{C}_{\text{closed}}$: its second set can be extended with a , i.e., $\mathcal{C}_{\text{connected}}(\{0.5, 2, 3\}, \{c, d\}, \{a\})$ holds.

2.2 Piecewise (Anti)-Monotonicity

DATA-PEELER can extract the complete collection of the closed 3-sets that hold in a dynamic graph. Its enumeration strategy is based on an enumeration tree which safely prunes the search space (i.e., no closed 3-set is missed) thanks to $\mathcal{C}_{\text{connected}}$, $\mathcal{C}_{\text{closed}}$ and any other user-defined piecewise (anti)-monotonic constraint. The efficient enforcement of any piecewise (anti)-monotonic constraint

\mathcal{C} is one of the main advantages of this principled algorithm. By “efficient”, we mean DATA-PEELER *sometimes* can, without any access to the data, affirm that the sub-tree rooted by the current enumeration node is empty of (not necessarily connected or closed) 3-sets satisfying \mathcal{C} . When the node is a leaf, it, not only *sometimes*, but *always* can check \mathcal{C} , hence ensuring the correctness (i.e., every extracted closed 3-set verifies \mathcal{C}). [5] provides a detailed presentation of the DATA-PEELER algorithm. Because of space constraints, this article only recalls the definition of piecewise (anti)-monotonicity.

Definition 4 (Monotonicity on the first argument). *A constraint \mathcal{C} is monotonic on the first argument iff $\forall (T, T', N^1, N^2) \in 2^T \times 2^{T'} \times 2^{N^1} \times 2^{N^2}$, $T \subseteq T' \Rightarrow (\mathcal{C}(T, N^1, N^2) \Rightarrow \mathcal{C}(T', N^1, N^2))$. \mathcal{C} is anti-monotonic on the first argument iff $T \subseteq T' \Rightarrow (\mathcal{C}(T', N^1, N^2) \Rightarrow \mathcal{C}(T, N^1, N^2))$.*

In a similar way, the monotonicity (or anti-monotonicity) on the second (or third) argument can be defined.

Example 3. Consider the following constraint:

$$\text{A 3-set } (T, N^1, N^2) \text{ is 8-large} \Leftrightarrow |T \times N^1 \times N^2| \geq 8 .$$

This constraint is monotonic on the first argument. Indeed, $\forall (T, T', N^1, N^2) \in 2^T \times 2^{T'} \times 2^{N^1} \times 2^{N^2}$, $T \subseteq T' \Rightarrow (|T \times N^1 \times N^2| \geq 8 \Rightarrow |T' \times N^1 \times N^2| \geq 8)$. It is monotonic on the second and on the third argument too.

The class of piecewise (anti)-monotonic constraints not only contains every constraint which is either monotonic or anti-monotonic on each of its arguments but also every constraint whose expression is such that, when giving a different variable to every occurrence of the three original arguments (related to the three sets \mathcal{T} , \mathcal{N}^1 and \mathcal{N}^2), the newly obtained constraint is either monotonic or anti-monotonic on each of its arguments.

Example 4. Assume $\mathcal{T} \in \mathbb{R}_+^{|\mathcal{T}|}$ and consider the following constraint:

$$\mathcal{C}_{16\text{-small-in-average}}((T, N^1, N^2)) \Leftrightarrow T \neq \emptyset \wedge \frac{\sum_{t \in T} t}{|T|} \leq 16 .$$

This constraint is both monotonic and anti-monotonic on the second and the third argument (neither N^1 nor N^2 appearing in the expression of the constraint) but it is neither monotonic nor anti-monotonic on the first argument. However, giving three different variables T_1 , T_2 and T_3 to each of the occurrences of T creates this new constraint which is monotonic on the first and third arguments (T_1 and T_3) and anti-monotonic on the second one (T_2):

$$\mathcal{C}'_{16\text{-small-in-average}}(T_1, T_2, T_3, N^1, N^2) \equiv T_1 \neq \emptyset \wedge \frac{\sum_{t \in T_2} t}{|T_3|} \leq 16 .$$

Therefore $\mathcal{C}_{16\text{-small-in-average}}$ is, by definition, piecewise (anti)-monotonic.

Obviously, a conjunction of piecewise (anti)-monotonic constraints is piecewise (anti)-monotonic. We now formalize the search for relevant patterns (the, so-called, δ -contiguous closed 3-cliques) as the correct and complete extraction of closed 3-sets satisfying a conjunction of piecewise (anti)-monotonic constraints.

2.3 δ -Contiguous Closed 3-Sets

Given $\delta \in \mathbb{R}_+$, a δ -contiguous 3-set is such that it is possible to browse the whole subset of timestamps by jumps from one timestamp to another without exceeding a delay of δ for each of these jumps:

Definition 5 (δ -contiguity). *It is said that a 3-set (T, N^1, N^2) is δ -contiguous, denoted $\mathcal{C}_{\delta\text{-contiguous}}((T, N^1, N^2))$, iff $\forall t \in [\min(T), \max(T)], \exists t' \in T$ s.t. $|t - t'| \leq \delta$.*

The constraint $\mathcal{C}_{\delta\text{-contiguous}}$ is piecewise (anti)-monotonic.

Proof. Let $\mathcal{C}'_{\delta\text{-contiguous}}$ the following constraint:

$$\begin{aligned} & \mathcal{C}'_{\delta\text{-contiguous}}(T_1, T_2, T_3, N_1, N_2) \\ \equiv & \forall t \in [\min(T_1), \max(T_2)], \exists t' \in T_3 \text{ s.t. } |t - t'| \leq \delta . \end{aligned}$$

The three arguments T_1, T_2 and T_3 substitute the three occurrences of T (in the definition of $\mathcal{C}_{\delta\text{-contiguous}}$). $\mathcal{C}'_{\delta\text{-contiguous}}$ is monotonic on its third argument and anti-monotonic on its first and second arguments ($T \subseteq T_1 \Rightarrow \min(T) \geq \min(T_1)$ and $T \subseteq T_2 \Rightarrow \max(T) \leq \max(T_2)$). Moreover, since its two last arguments do not appear in its expression, $\mathcal{C}'_{\delta\text{-contiguous}}$ is both monotonic and anti-monotonic on them. Therefore, by definition, $\mathcal{C}_{\delta\text{-contiguous}}$ is piecewise (anti)-monotonic. \square

$\mathcal{C}_{\text{connected}} \wedge \mathcal{C}_{\delta\text{-contiguous}}$ being stronger than $\mathcal{C}_{\text{connected}}$ alone, a related and weaker closedness constraint can be defined. Intuitively, a δ -closed 3-set is closed w.r.t. both \mathcal{N} sets and to the timestamps of \mathcal{T} in the vicinity of those inside the 3-set. Hence, a timestamp that is too far away (delay exceeding δ) from any timestamp inside the 3-set, cannot prevent its δ -closedness.

Definition 6 (δ -closedness). *It is said that a 3-set (T, N^1, N^2) is δ -closed, denoted $\mathcal{C}_{\delta\text{-closed}}((T, N^1, N^2))$, iff*

$$\left\{ \begin{array}{l} \forall t \in \mathcal{T} \setminus T, (\exists t' \in T \text{ s.t. } |t - t'| \leq \delta \Rightarrow \neg \mathcal{C}_{\text{connected}}(\{\{t\}, N^1, N^2\})) \\ \forall n^1 \in \mathcal{N} \setminus N^1, \neg \mathcal{C}_{\text{connected}}((T, \{n^1\}, N^2)) \\ \forall n^2 \in \mathcal{N} \setminus N^2, \neg \mathcal{C}_{\text{connected}}((T, N^1, \{n^2\})) \end{array} \right. .$$

The constraint $\mathcal{C}_{\delta\text{-closed}}$ is piecewise (anti)-monotonic.

Proof. Let $\mathcal{C}'_{\delta\text{-closed}}$ the following constraint:

$$\begin{aligned} & \mathcal{C}'_{\delta\text{-closed}}(T_1, T_2, T_3, T_4, N_1^1, N_2^1, N_3^1, N_1^2, N_2^2, N_3^2) \\ \equiv & \left\{ \begin{array}{l} \forall t \in \mathcal{T} \setminus T_1, (\exists t' \in T_2 \text{ s.t. } |t - t'| \leq \delta \Rightarrow \neg \mathcal{C}_{\text{connected}}(\{\{t\}, N_1^1, N_1^2\})) \\ \forall n^1 \in \mathcal{N} \setminus N_2^1, \neg \mathcal{C}_{\text{connected}}((T_3, \{n^1\}, N_2^2)) \\ \forall n^2 \in \mathcal{N} \setminus N_3^2, \neg \mathcal{C}_{\text{connected}}((T_4, N_3^1, \{n^2\})) \end{array} \right. . \end{aligned}$$

$\mathcal{C}'_{\delta\text{-closed}}$ is anti-monotonic on its second argument and monotonic on all its other arguments. Therefore, by definition, $\mathcal{C}_{\delta\text{-closed}}$ is piecewise (anti)-monotonic. \square

Definition 7 (δ -contiguous closed 3-set). *It is said that (T, N^1, N^2) is a δ -contiguous closed 3-set iff it satisfies $\mathcal{C}_{\text{connected}} \wedge \mathcal{C}_{\delta\text{-contiguous}} \wedge \mathcal{C}_{\delta\text{-closed}}$.*

Example 5. $(\{2, 3\}, \{a, b, c, d\}, \{d\})$ is a 1.75-contiguous closed 3-set in the toy dataset from Tab. [1](#). However, it is neither 0.5-contiguous (the timestamps 2 and 3 are not close enough) nor 2-closed (0 can extend the set of timestamps). This illustrates the fact that the number of δ -contiguous closed 3-sets is not monotonic in δ .

A δ -contiguous closed 3-set is an obvious generalization of a closed 3-set. Indeed, $\forall \delta \geq \max(\mathcal{T}) - \min(\mathcal{T}), \mathcal{C}_{\delta\text{-contiguous}} \equiv \text{true}$ and $\mathcal{C}_{\delta\text{-closed}} \equiv \mathcal{C}_{\text{closed}}$.

2.4 δ -Contiguous Closed 3-Cliques

We want to extract sets of nodes that are entirely interconnected. In this context, a 3-set (T, N^1, N^2) where $N^1 \neq N^2$ is irrelevant.

Definition 8 (Symmetry). *It is said that (T, N^1, N^2) is symmetric, denoted $\mathcal{C}_{\text{symmetric}}((T, N^1, N^2))$, iff $N^1 = N^2$.*

$\mathcal{C}_{\text{symmetric}}((T, N^1, N^2)) \equiv N^1 \subseteq N^2 \wedge N^2 \subseteq N^1$ is an equivalent definition to the symmetry constraint. In this form, a piecewise (anti)-monotonic constraint is identified.

Proof. Let $\mathcal{C}'_{\text{symmetric}}$ the following constraint:

$$\mathcal{C}'_{\text{symmetric}}(T, N_1^1, N_2^1, N_1^2, N_2^2) \equiv N_1^1 \subseteq N_1^2 \wedge N_2^2 \subseteq N_2^1 .$$

$\mathcal{C}'_{\text{symmetric}}$ is monotonic on its third and fourth arguments (N_2^1 and N_1^2) and anti-monotonic on its second and fifth arguments (N_1^1 and N_2^2). Moreover, since the first argument (T) does not appear in the expression of $\mathcal{C}'_{\text{symmetric}}$, this constraint is both monotonic and anti-monotonic on this argument. Therefore, by definition, $\mathcal{C}_{\text{symmetric}}$ is piecewise (anti)-monotonic. \square

$\mathcal{C}_{\text{connected}} \wedge \mathcal{C}_{\delta\text{-contiguous}} \wedge \mathcal{C}_{\text{symmetric}}$ being stronger than $\mathcal{C}_{\text{connected}} \wedge \mathcal{C}_{\delta\text{-contiguous}}$, a related and weaker closedness constraint can be defined. Intuitively, if not *both* the line and the column pertaining to a node n can *simultaneously* extend a 3-set without breaking $\mathcal{C}_{\text{connected}}$, the closedness is not violated:

Definition 9 (Symmetric δ -closedness). *It is said that a 3-set (T, N^1, N^2) is symmetric δ -closed, denoted $\mathcal{C}_{\text{sym-}\delta\text{-closed}}((T, N^1, N^2))$, iff*

$$\begin{cases} \forall t \in \mathcal{T} \setminus T, (\exists t' \in T \text{ s. t. } |t - t'| \leq \delta \Rightarrow \neg \mathcal{C}_{\text{connected}}(\{\{t\}, N^1, N^2\})) \\ \forall n \in \mathcal{N} \setminus (N^1 \cap N^2), \neg \mathcal{C}_{\text{connected}}((T, N^1 \cup \{n\}, N^2 \cup \{n\})) \end{cases} .$$

The constraint $\mathcal{C}_{\text{sym-}\delta\text{-closed}}$ is piecewise (anti)-monotonic.

Proof. Let $\mathcal{C}'_{\text{sym-}\delta\text{-closed}}$ the following constraint:

$$\begin{aligned} & \mathcal{C}'_{\text{sym-}\delta\text{-closed}}((T_1, T_2, T_3, N_1^1, N_2^1, N_3^1, N_1^2, N_2^2, N_3^2)) \\ \equiv & \begin{cases} \forall t \in \mathcal{T} \setminus T_1, (\exists t' \in T_2 \text{ s.t. } |t - t'| \leq \delta \Rightarrow \neg \mathcal{C}_{\text{connected}}(\{\{t\}, N_1^1, N_1^2\})) \\ \forall n \in \mathcal{N} \setminus (N_2^1 \cap N_2^2), \neg \mathcal{C}_{\text{connected}}((T, N_3^1 \cup \{n\}, N_3^2 \cup \{n\})) \end{cases} . \end{aligned}$$

$\mathcal{C}'_{\text{sym-}\delta\text{-closed}}$ is anti-monotonic on its second argument (T_2) and monotonic on all its other arguments. Therefore, by definition, $\mathcal{C}_{\text{sym-}\delta\text{-closed}}$ is piecewise (anti)-monotonic. \square

Definition 10 (δ -contiguous closed 3-clique). *It is said that (T, N^1, N^2) is a δ -contiguous closed 3-clique iff it satisfies $\mathcal{C}_{connected} \wedge \mathcal{C}_{\delta\text{-contiguous}} \wedge \mathcal{C}_{symmetric} \wedge \mathcal{C}_{sym\text{-}\delta\text{-closed}}$.*

Example 6. Two out of the three closed 3-sets illustrating Ex. 2 are symmetric: $(\{2, 3\}, \{a, c, d\}, \{a, c, d\})$ and $(\{0, 3\}, \{b, d\}, \{b, d\})$. $(\{0.5, 2, 3\}, \{c, d\}, \{c, d\})$ is not closed w.r.t. \mathcal{C}_{closed} (see Ex. 2) but it is symmetric 1.75-closed. Indeed, the node a cannot simultaneously extend its second and third sets of elements without violating $\mathcal{C}_{connected}$.

Since $\mathcal{C}_{connected}$, $\mathcal{C}_{\delta\text{-contiguous}}$, $\mathcal{C}_{symmetric}$ and $\mathcal{C}_{sym\text{-}\delta\text{-closed}}$ are piecewise (anti)-monotonic, DATA-PEELER can handle them all, i.e., it can efficiently compute every 3-set satisfying $\mathcal{C}_{connected} \wedge \mathcal{C}_{\delta\text{-contiguous}} \wedge \mathcal{C}_{symmetric} \wedge \mathcal{C}_{sym\text{-}\delta\text{-closed}}$. In practical settings, the complete collection of the δ -contiguous closed 3-cliques is huge. It makes sense to constrain further the extraction tasks (i.e., to enforce also a new user-defined piecewise (anti)-monotonic constraint \mathcal{C}) to take subjective interestingness into account and to focus on more relevant patterns.

3 Experimental Results

Vélo'v is a bicycle rental service run by the city of Lyon, France. 338 Vélov stations are spread over this city. At any of these stations, the users can take a bicycle and return it to any other station. Whenever a bicycle is rented or returned, this event is logged. We focus here on the data generated along the year 2006. These data are aggregated to obtain one graph per period of time (we chose a period of 30 minutes). The set of nodes \mathcal{N} of such a graph corresponds to the Vélov stations. Its edges are labelled with the total number of rides in 2006 between the two linked stations (whatever their orientation) during the considered period of time. Setting a threshold allows to select the most significant edges. Many sensible strategies can be chosen to fix this threshold (which can be different between the graphs). Here is the strategy we opted for:

Binarization. Given a graph whose edges are labelled by real values, let m be the maximum of these values. The threshold is fixed to $0.2 \times m$.

Once the thresholds set, all edges linked to some station may be considered insignificant. Such an infrequently used station is removed from the dynamic graph. In our experiments, 204 stations remained after applying the binarization.

On an AMD SempronTM 2600+ running a GNU/LinuxTM operating system, our implementation in C++ extracts all 3849 closed 3-cliques within about three minutes and a half (extracting all 122581 closed 3-sets takes over nine minutes). Nevertheless, to assess, by hand, the quality of the extracted δ -contiguous closed 3-cliques, the returned collection must be small. Hence stronger constraints are enforced. The minimal number of Vélov stations that must be involved in a δ -contiguous closed 3-clique is set to 6 and the minimal number of periods to 4. Thanks to these constraints, only three 0.5-continuous closed 3-cliques are

returned within one minute and 49 seconds. Two of them take place during the evening (they start at half past 19) and gather stations that are in the center of Lyon. They differ by one station and one of them runs along one more time period. The third 0.5-contiguous closed 3-clique is displayed in Fig 2(a). The circles stand for the geographical positions of the Vélov stations. The filled ones are involved in the shown pattern. The disposition of the stations follows one of the greatest street in Lyon: “Cours Gambetta”. Obviously it is much used by the riders during the evening. The outlying Vélov station is, overall, the most frequently used one: “Part-Dieu/Vivier-Merle”. At this place, the rider finds the unique commercial center in Lyon, the main train station, etc. Extracting, with the same minimal size constraints, the 1-contiguous closed 3-cliques provides a collection of nine patterns. Among them, the three 0.5-contiguous closed 3-cliques are found unaltered; some slight variations of them are found (one or two stations are changed); one pattern takes place during the morning (to obtain patterns involving night periods the constraints must be weakened a lot: nightly rides do not comply much with a model). The majority of the extracted 1-contiguous closed 3-cliques involves Vélov stations in the city center. Figure 2(b) depicts one of them. The disposition of the stations follows the street binding the two most active districts in Lyon: “Rue de la Part-Dieu”. The outlying Vélov station is, overall, one of the most frequently used: “Opéra”. At this place, the rider can find, not only the opera, but also the town hall, a cinema, bars, etc.

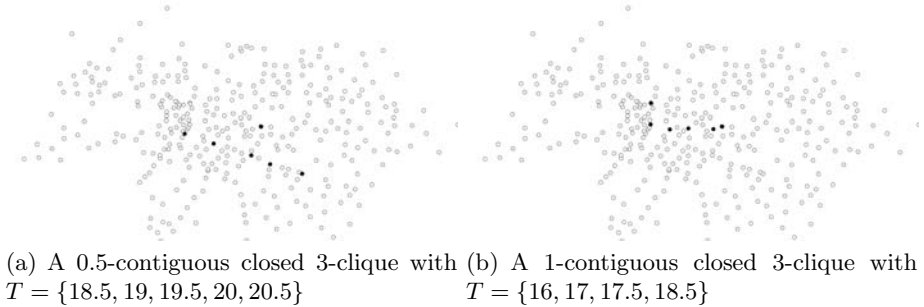


Fig. 2. Two closed 3-cliques extracted under strong constraints

4 Related Work

DATA-PEELER faces two competitors able to extract all closed 3-sets from ternary relations: CUBEMINER [2] and TRIAS [3]. None of them have the generality of DATA-PEELER. In particular, they cannot deal with n -ary relations and cannot enforce any piecewise (anti)-monotonic constraints. This latter remark makes them harder to specialize in the extraction of δ -contiguous closed 3-cliques. Furthermore, [5] shows that DATA-PEELER outperforms both of them by orders of magnitude. The interested reader will refer to the “Related Work” section of this article for a detailed analysis of what makes DATA-PEELER more efficient.

Collections of large graphs were built to help in understanding genetics. These graphs commonly have tens of thousands of nodes and are much noisy. There is a need to extract patterns crossing such graphs, e.g., patterns that remain valid across several co-expression graphs or patterns crossing the data pertaining to physical interactions between molecules (e.g., protein-gene) and more conceptual data (e.g., co-expression of genes). One of the most promising pattern helping in these tasks is the closed 3-clique or, better, the closed quasi-3-clique. CLAN [6] is able to extract closed 3-cliques from collections of large and dense graphs. Crochet+ [7], Cocain* [8] and Quick [9] are the state-of-the-art extractors of closed quasi-3-cliques. They all use the same definition of noise tolerance: every node implied in a pattern must have, in every graph independently from the others, a degree exceeding a user-defined proportion of the maximal degree it would reach if the clique was exact. An ongoing research for us is to generalize $\mathcal{C}_{\text{connected}}$ and $\mathcal{C}_{\text{closed}}$ to allow some fault-tolerance, i.e., to extract closed quasi-3-sets. Unlike the previous approaches, such a generalization of DATA-PEELER would allow the discovery of (possibly δ -contiguous) closed quasi-3-cliques in dynamic *directed* graphs. Discussing this further is out of the scope of this paper.

The δ -contiguity stems from an analogous constraint, called *max-gap* constraint, initially applied to sequence mining. It was introduced in the GSP approach [10]. The *min-gap* and the *window size* constraints [10] uses as well could be enforced in our approach too. Nevertheless, in [10], these constraints modify the enumeration order, whereas, in our approach, they reduce the search space and let the enumeration strategy unaltered. Furthermore, in the context of [10], the considered datasets are multiple sequences of itemsets and the extracted patterns are sub-sequences of itemsets whose order (but not position in time) is to be respected in all (1-dimensional) supporting sequences, whereas, in our approach, the supporting domain contains (2-dimensional) graphs that must be aligned in time. Notice that the max-gap constraint was used in other contexts. For example, [11] enforces it to extract frequent episodes and [12] extracts, under a max-gap constraint, frequent sub-sequences whose support is the sum of the number of repetitions in all sequences of the dataset.

5 Conclusion

This article deals with mining δ -contiguous closed 3-cliques. The constraints imposed to achieve this goal were proved piecewise (anti)-monotonic such that DATA-PEELER efficiently handles them all. Notice that ad-hoc optimizations for the studied conjunction of constraints have been already studied and implemented. However, they need the presentation of many technical details. Due to the severe space constraint, we decided to emphasize how, in its original form, the DATA-PEELER closed n -set extractor can be specialized in computing all δ -contiguous closed 3-cliques. We mentioned that an ongoing work concerns the declarative specification of quasi-cliques to support the discovery of more relevant patterns from real data sets. Notice also that DATA-PEELER can mine closed n -sets (or cliques) with n an arbitrary integer greater or equal to 2. It

can also enforce the contiguity of the patterns on several dimensions at the same time (possibly with different δ values). More generally, DATA-PEELER can mine closed n -sets adapted to any specific problem that can be expressed in terms of piecewise (anti)-monotonic constraints what appears definitively useful for many dynamic graph analysis processes.

Acknowledgments. This work has been funded by ANR BINGO2 (MDCO 2007-2010). Tran Bao Nhan Nguyen has contributed to this study thanks to a Research Attachment programme between the Nanyang Technological University (Singapore), where he is an undergraduate student, and INSA-Lyon. Finally, we thank Dr. J. Besson for exciting discussions.

References

1. Ganter, B., Stumme, G., Wille, R.: *Formal Concept Analysis, Foundations and Applications*. Springer, Heidelberg (2005)
2. Ji, L., Tan, K.-L., Tung, A.K.H.: Mining frequent closed cubes in 3D data sets. In: *VLDB 2006: Proc. of the 32nd Int. Conf. on Very Large Data Bases*, pp. 811–822. VLDB Endowment (2006)
3. Jaschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: TRIAS—an algorithm for mining iceberg tri-lattices. In: *ICDM 2006: Proc. of the Sixth Int. Conf. on Data Mining*, pp. 907–911. IEEE Computer Society, Los Alamitos (2006)
4. Cerf, L., Besson, J., Robardet, C., Boulicaut, J.F.: Data-Peeler: Constraint-based closed pattern mining in n -ary relations. In: *SDM 2008: Proc. of the Eighth SIAM Int. Conf. on Data Mining*, pp. 37–48. SIAM, Philadelphia (2008)
5. Cerf, L., Besson, J., Robardet, C., Boulicaut, J.F.: Closed patterns meet n -ary relations. *ACM Trans. on Knowledge Discovery from Data* 3(1) (March 2009)
6. Wang, J., Zeng, Z., Zhou, L.: CLAN: An algorithm for mining closed cliques from large dense graph databases. In: *ICDE 2006: Proc. of the 22nd Int. Conf. on Data Engineering*, pp. 73–82. IEEE Computer Society, Los Alamitos (2006)
7. Jiang, D., Pei, J.: Mining frequent cross-graph quasi-cliques. *ACM Trans. on Knowledge Discovery from Data* 2(4) (January 2009)
8. Zeng, Z., Wang, J., Zhou, L., Karypis, G.: Out-of-core coherent closed quasi-clique mining from large dense graph databases. *ACM Trans. on Database Systems* 32(2), 13–42 (2007)
9. Liu, G., Wong, L.: Effective pruning techniques for mining quasi-cliques. In: *ECML PKDD 2008: Proc. of the European Conf. on Machine Learning and Knowledge Discovery in Databases - Part II*, pp. 33–49. Springer, Heidelberg (2008)
10. Srikant, R., Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) *EDBT 1996*. LNCS, vol. 1057, pp. 3–17. Springer, Heidelberg (1996)
11. Casas-Garriga, G.: Discovering unbounded episodes in sequential data. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) *PKDD 2003*. LNCS (LNAI), vol. 2838, pp. 83–94. Springer, Heidelberg (2003)
12. Ding, B., Lo, D., Han, J., Khoo, S.-C.: Efficient mining of closed repetitive gapped subsequences from a sequence database. In: *ICDE 2009: Proc. of the 25th Int. Conf. on Data Engineering*. IEEE Computer Society, Los Alamitos (2009)

Statistical Characterization of a Computer Grid

Lovro Ilijašić and Lorenza Saitta

Università del Piemonte Orientale, Dip. Informatica, Alessandria, Italy
lovro@di.unito.it, saitta@mf.n.unipmn.it

Abstract. Large-scale statistical analysis of more than 28 million jobs collected during 20 months of grid activity was undertaken in order to examine the relations between users, computing elements and jobs in the network. The results give insight into the global system behaviour and can be used to build models applicable in various contexts of grid computing. As an example, we here construct probabilistic models that prove to be able to accurately predict job abortion.

1 Introduction

Grid computing has emerged as an important new field, focusing on large-scale resource sharing by heterogeneous user communities. It has proven to be a powerful means for merging the computational and storage power of geographically distributed sets of computers. In order to exploit the full potentiality of grid computing, achieving a deep understanding of the underlying topology of the computer network, of the behaviour of both the single elements and the whole, and of the interactions among computational elements and users is fundamental.

Grids produce huge amounts of data, which allow extensive analysis to be performed using statistical and data mining methods. The results provide a better global picture of the system and its behaviour.

Building models of grid network using Machine Learning techniques is still in its infancy, but it should prove useful in various applications. Models can be used to generate test data for simulating a grid in future research, for prediction of oncoming events in order to optimize the scheduling and workload distribution, as well as for detection of outliers, intrusion or other anomalous behaviours in the system. In this paper, we first describe some relevant results of the extensive statistical analysis of a production grid (EGEE), and then the probabilistic models built thereupon and tested on predicting relevant job parameters.

2 Context

We are here interested in the global characterization of a grid system, which means that our perspective is defined by the objects that are recognizable from the highest level of abstraction. Computational grid is a system of computational elements created and used by humans, and so are these two the main classes of objects that we investigate here – Users and Computing Elements (CEs). Actions

of users on CEs consist in sending jobs, which users register in the system, and that are processed on a given CE. Jobs can therefore be seen both as a link between Users and CEs, and as separate objects having their own attributes, like *total time* spent in the system, its *runtime length*, *abortion state*, etc.

One of the standard components in grid systems is the *Logging and Bookkeeping* (L&B) service. It tracks jobs managed by Workload Management Systems, gathers events from various components and processes them in order to give a higher level view, the status of job. All the important job data and events are fed to L&B internally from other middleware components transparently from the user's point of view. The Logging and Bookkeeping data contains detailed information about every job, but to get a broader view of the system, we needed the data that was collected during a longer period of time from various Virtual Organizations. Such a dataset is collected from all major Resource Brokers (RBs) by The Real Time Monitor, developed by Imperial College, London.

The dataset we use was gathered during 20 months period (September 1st, 2005 to April 30th, 2007). Each row in the dataset contains the summary of a single job. There are 28,384,971 rows (i.e. jobs) and each row contains attributes such as the job id, user id, Virtual Organization, registration time of the job, Resource Broker name, name of the Computing Element that processed the job, number of job resubmissions, job termination status, etc. The data is generated by the activity of 3,529 users on 760 Computing Elements during 607 days.

3 Explorative Analysis

Different analysis of jobs in grids have already been undertaken [5], [6], but here we perform a large-scale analysis in order to recognize certain relations that may appear on the global level.

Let $\mathbf{U} = \{U_j \mid 1 \leq j \leq N\}$ be the set of users, $\mathbf{CE} = \{CE_k \mid 1 \leq k \leq M\}$ the set of Computing Elements, and $\mathbf{J} = \{J_i \mid 1 \leq i \leq R\}$ the set of jobs. Let moreover T be the total time span of the data logs, and Δt a suitably chosen time interval. In this paper we have $T = 20$ months and we will use different levels of time abstraction: $\Delta t = 1 \text{ day}, 1 \text{ week}, 1 \text{ month}$.

We can further define: $\mathbf{CE}_h(U_j) = \{CE_k \mid U_j \text{ sends jobs to } CE_k \text{ during time interval } \Delta t_h\}$ be the subset of computing elements used by U_j , $\mathbf{U}_h(CE_k) = \{U_j \mid U_j \text{ sends jobs to } CE_k \text{ during time interval } \Delta t_h\}$ the subset of users who send jobs to CE_k , and $\mathbf{J}_h(U_j, CE_k) = \{(U_j, CE_k) \mid U_j \text{ sends jobs to } CE_k \text{ during time interval } \Delta t_h\}$ the set of pairs (user, computing element) such that the user sends jobs to the computing element. The time interval may be at any of the chosen time abstraction level.

3.1 Analysis from the Point of View of the Computing Element

During the period T , an average number of jobs arrived to any one among the $CE_k \in \mathbf{CE}$ in one day is 53.8 jobs. Let $r_k = \sum_{j=1}^N |J(U_j, CE_k)| =$ Total number of jobs received by CE_k during T . Figure 1 shows r_k for each CE_k , which are sorted in descending order by r_k , in linear and log-log scale.

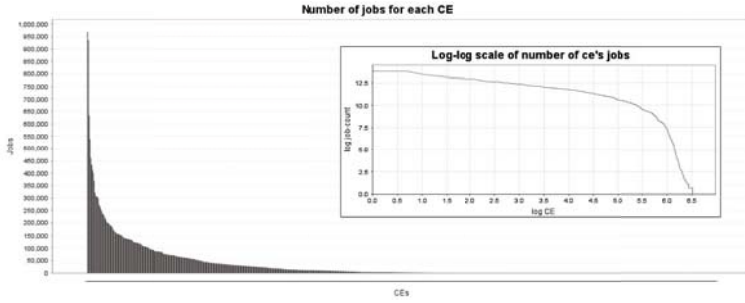


Fig. 1. Total number r_k of jobs received by CE_k during the global time T

To understand the temporal character of the grid, we plot the graph of number of active CEs through weeks. Figure 2 presents both number of CEs in a single week (dark bars) and overall number of CEs that appeared up to (and including) that week (lighter bars). We note the regular linear growth, both for total number of CEs in the system, and for the number of different CEs during the time period Δt .

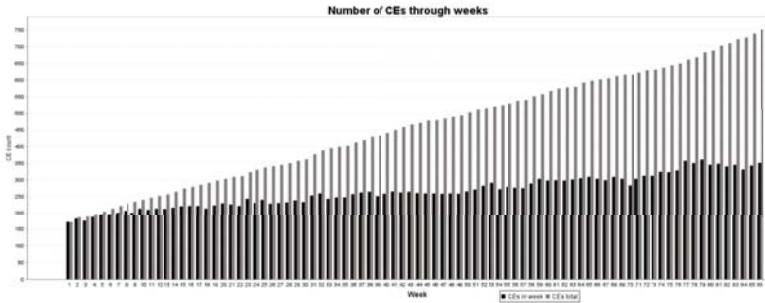


Fig. 2. Number of CEs through weeks

3.2 Analysis from the Point of View of the User

The analogous analysis can be performed focusing on a user. Let $s_j = \sum_{k=1}^M |J(U_j, CE_k)| = \text{Total number of jobs registered by } U_j \text{ during } T$. Figure 3(a) displays s_j 's, with the users ordered on x-axis according to descending values of s_j . The diagram in Figure 3(b) shows the same distribution in log-log scale.

To get insight in the temporal characteristics, we can plot the diagram of number of different users using grid every week. The diagram in Figure 4 shows both the number of unique users in a single week (darker bars) and the overall number of users in the system up to (and including) the given week (lighter bars). The hole around week 41 is due to missing user data in the dataset, where jobs are listed without information on who registered them.

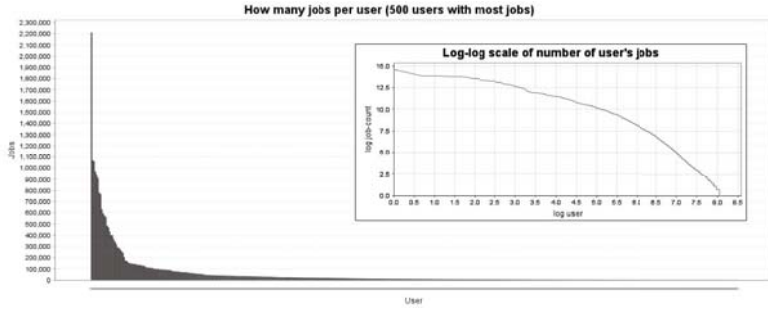


Fig. 3. Total number s_j of jobs registered by U_j during the global time T

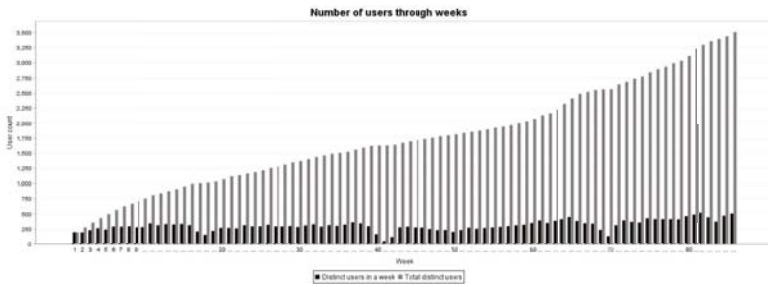


Fig. 4. Number of users through weeks

3.3 Analysis from the Point of View of the Job

On the set \mathbf{J} of all jobs, we observe two main time parameters for each job: its total length in the system and its processing time on a worker node, given in seconds. Calculated maximum and mean times of *total length* parameter are: maximum *total length* = 14,881,321 sec = 172.2 days; mean *total length* = 29,031.2 sec = 8.06 hours.

The second time parameter of a job that we observe is its *worker node length*. It is reported by log monitor and represents the time that job spent in processing: maximum *worker node length* = 6,717,807 sec = 77.75 days; mean *worker node length* = 9,792.5 sec = 2.72 hours. In average, *worker node length* takes 39.4% of job’s total time in the system. The time before processing, which includes the time spent on Resource Broker, in transfer and in queue on the Computing Element is 38.1%, while the time that job waits in the system to be cleared after the processing is finished is in average 22.5%.

The linear and log scale time distributions of *worker node length* parameter are given in Figure 5. On the linear scale, bin size for x-axis is 60 seconds and first 200 bins are shown. In log scale, each bin is twice as wide as the previous one: [0s, 15s), [15s, 30s), [30s, 60s), [60s, 120s), etc.

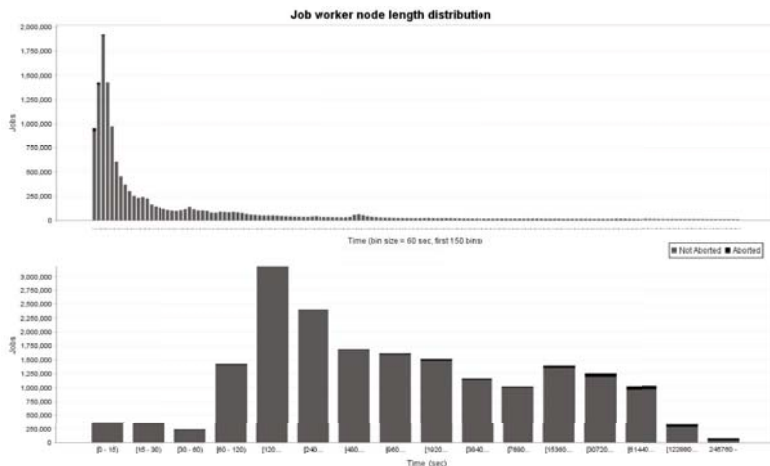


Fig. 5. Job worker node length time distributions in a) linear scale b) log scale

Again, in order to study the evolution of the network and its trends, we investigate the job statistics through time. In Figure 6 we show the number of registered jobs in the system through weeks. The height of the bar represents the number of jobs each week, while the black part of each bar represents the share of aborted jobs. Figure 6 clearly shows the growth of the network in terms of jobs registered.

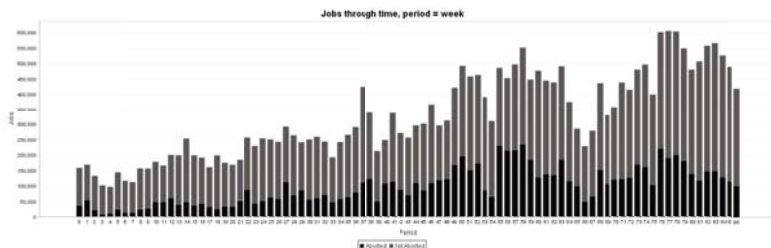


Fig. 6. Number of jobs through weeks during the 20 observed months. The black part of the bars represents the aborted jobs.

4 Grid as a Complex Network

One way to understand grid, and thus facilitate its analysis, is to present it as a bipartite, directed graph. The two types of nodes that appear in such a graph are Users and CEs. The edges in the graph exist only between nodes of different types if U sends at least one job to the CE. The edges are directed from U to CE. This network has both the properties of social and technological (information) network, gathering them into a unified picture of a grid.

Depending on the goals of analysis, this graph can also be weighted. Each edge can be assigned the number of jobs that U runs on CE. The graph is thus bipartite, directed, weighted graph. Having such definition of the graph we can use standard methods and tools of Complex Network Analysis. Visualisation of the part of the network presenting only nodes and edges with more than 10,000 jobs is created with such a tool and presented in Figure 7 [1].

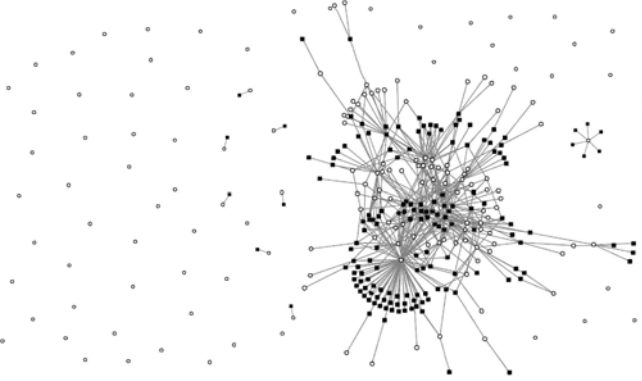


Fig. 7. Social network of ‘large’ nodes and edges (having more than 10.000) jobs. CEs are presented as black squares, users as white dots.

The whole network consists of three connected components, where the two small ones consist of only two nodes each (one user and one CE). Some parameters that describe the network [2] are: The diameter of the network, which in our case has the value 10; and the average shortest path between a pair of nodes, which is 3.56.

Valuable information on a network are its edge count distributions. Out-degree of a node is the number of directed edges that go out from it. As edges represent connections between users and CEs, it is in fact the number of different CEs that a user runs its jobs on. In Figure 8(a) we show a bar for each user with the height representing the number of CEs it uses (its out-degree). Users are sorted in descending order by the number of CEs they use. Figure 9 gives the reverse representation, the distribution of the out-degree parameter both in linear and in log-log scale in order to examine the probable power law distribution.

Respectively, the in-degree of a node is the number of edges that end in it. All the edges are directed from users to CEs, so the in-degree here represents the number of distinct users that run their jobs on the given CE. Figure 8(b) shows a bar for each CEs representing its in-degree, sorted in descending order.

We also investigated the correlation between out-degrees and in-degrees of neighbouring nodes. There actually exists a correlation between the degrees of neighbouring nodes, which shows that the network is disassortative [8]. A network is disassortative if large (small) degree nodes tend to be linked with small (large) degree nodes. Biological and technological networks (like the Internet) are examples of disassortative networks.

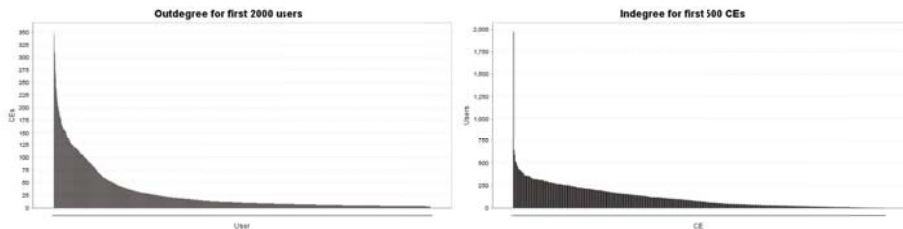


Fig. 8. a) Out-degrees of 2000 biggest users, b) In-degrees of 500 biggest CEs

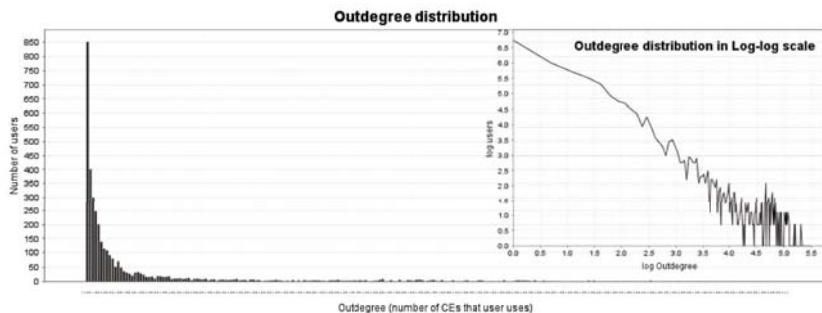


Fig. 9. Out-degree distributions in linear and log-log scale

5 Prediction of Job Abortion

We now try to use the information on jobs, users and CEs that we obtained in the previous analyses and apply them on predicting the outcome of an oncoming job (i.e. whether the job is going to be aborted or it will terminate successfully). We see that 30.3% of all the jobs in our dataset are aborted. Of all the aborted jobs, 38.4% are aborted on Resource Broker, mostly for the “No compatible resources” reason. We are, though, more interested in predicting job abortions on a Computing Element, as this result could be applicable to optimizing schedulers and grid performance.

Different attempts of predicting performance in grids are available in literature [4], [7], [9]. Here we try to build a probabilistic Graphical Model [3] that would make use of the gathered observations. By first examining the dependence of relevant job parameters on users and CEs, we tried to understand if some users (CEs) are more prone to have their jobs aborted than the others. In Figure 10(a) we plot a bar with probability of aborting a job for each user that has more than 1000 jobs, and as a result we see that average job abortion rate is a very individual parameter of a user. Figure 10(b) shows the analogous report for Computing Elements with more than 1000 jobs.

The simplest model would use only the information on the user (or CE) to calculate its personal historical abort rate and use it for predicting the outcome of the future one. But in order to get the more accurate prediction, we build the

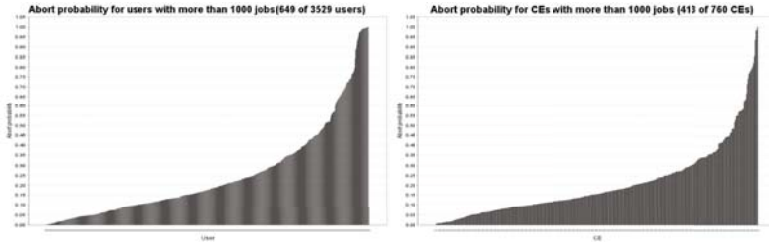


Fig. 10. Abort probabilities for a) users b) CEs with more than 1000 jobs

	Real aborted (5,078,639 jobs)	Real not aborted (18,998,586 jobs)
Predicted aborted	4,195,674 82.61% of real abortions	557,432 2.93% of real not abortions
Predicted not aborted	861,365 16.97% of real abortions	18,360,638 96.64% of real not abortions

Fig. 11. Confusion matrix for predicting job abortion using Dynamic Bayesian Network

model which combines the information on User and CE, which thus presumes that users usually behave in a similar way on the same CEs. The next step is to introduce a Dynamic Bayesian Network, which assumes a Markov behaviour of the job abortion, where the abortion state also depends on the abortion outcome of the previously registered job of the same User/CE pair.

The experiment consists of passing chronologically through all the jobs in the dataset and calculating the probability of abortion using information on User, CE and the previous job the User/CE pair. If the probability of abortion for the user is greater than 0.5, we decide that the following job is going to be aborted and compare it with the real outcome of the job. Afterwards, we update our model by incrementing the appropriate counter.

The result of the experiment is given by the confusion matrix in Figure 11.

The most important result is that 82.6% of all the aborted jobs are predicted correctly in this way, while the overall correct prediction for both aborted and not-aborted jobs is 94%. This model requires information on assigned CE, so the statistics includes only the jobs that were ran or aborted on a CE (more than 24 million jobs).

These experiments show strong dependency of job abortion on User, CE and the result of the previously registered job. The combined information on User and CE tells us that users indeed ‘behave’ differently on different CEs. This can also be understood as the result of the activity of Resource Brokers, which dispatch similar jobs on the same CEs and thus perform ‘clustering’ of user’s jobs. The dependency on the outcome of the previous job is also an important observation that tells us about the nature of job registering, which usually includes bursts of similar jobs.

These properties have been tested on historical data, where we have all the information already available offline. But for applying these results on real-time scheduling, we have to consider the on-line version of the described model. The main problem we encounter is the lack of information on the most similar jobs (i.e. the previous ones) if they aren't finished yet. Therefore, we must rely on the state of the previously finished job whose outcome we already have.

Most of the aborted jobs in our dataset have no timestamps which are needed for testing the on-line algorithm, but the results with the ones that have this information show that (with the threshold set to 0.5) it can correctly predict 27.5% of aborted jobs (98.5% of successfully terminated ones) at the job registration time, and 37.5% of aborted jobs (98.8% of not aborted) at the moment when the job is about to start being processed on a Computing Element. This difference is due to the time that passes between these two moments in a job's life cycle, during which we gather more information on previous jobs.

In our experiments the jobs having the probability of abortion higher than 0.5 were predicted aborted. If we define the gain of correctly guessing job abortion and not-abortion, this threshold can be further adjusted to give better overall outcome. For example, if we lower the threshold to 0.1, at registration time we correctly predict 36.5% of abortions and 96.2% of not abortions. At running time, these percentages are 47.8% and 96.6% respectively. As the result, we get more job abortions predicted correctly, but more error in predicting the successful outcome.

6 Conclusions and Future Work

We hereby presented some interesting results of the extensive large-scale analysis of the grid log data collected during a 20 months long period of time, containing more than 28 million jobs. The conclusions from such a huge dataset can present us valuable information in various fields of grid research.

We also applied these results on modelling job parameters. The model for predicting job abortion state is described here, while an analogous approach was used for predicting both *total* and *worker node* job length with significant success, but the results are not reported here for the sake of brevity. These models can be of much importance for scheduling of the jobs, both on the level of Workload Management System on Resource Broker and for local scheduling on a Computing Element. The important property of the models is the possibility to build them on-line, i.e. they are created and updated in real time with very low computational complexity, due to having only one task – to predict the next state. These models can be used for other purposes as well, like inferring the user that ran certain series of jobs (that might be used in intrusion detection, perhaps). Furthermore, if the prediction is needed for local scheduling on a Computing Element, the memory and processing requirements get even lower, because the Computing Element parameter is constant.

In the context of predicting job parameters, other attributes are to be examined further to see if they can improve the models. Some experiments were

already done by extending the model with the job interarrival times and the length of the previous job. In combining the results of these models, a meta-learner could also be a suitable solution.

Acknowledgments

This work has been partially supported by the European Infrastructure Project EGEE-III INFSO-RI-222667.

References

1. Adar, E.: Guess: a language and interface for graph exploration. In: CHI 2006: Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 791–800. ACM, New York (2006)
2. Albert, R., Barabási, A.-L.: Statistical mechanics of complex networks. *Rev. Mod. Phys.* 74(1), 47–97 (2002)
3. Bishop, C.M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, New York (2006)
4. Kapadia, N.H., Fortes, J.A., Brodley, C.E.: Predictive application-performance modeling in a computational grid environment. In: *International Symposium on High-Performance Distributed Computing*, vol. 0, p. 6 (1999)
5. Li, H., Groep, D.L., Wolters, L., Templon, J.: Job failure analysis and its implications in a large-scale production grid. In: *e-Science*, p. 27 (2006)
6. Hui, L., Michael, M.: Analysis and modeling of job arrivals in a production grid. *SIGMETRICS Perform. Eval. Rev.* 34(4), 59–70 (2007)
7. Li, H., Wolters, L.: An investigation of grid performance predictions through statistical learning 1st workshop on tackling. In: *Computer System Problems with Machine Learning Techniques (SysML)*, in conjunction with ACM Sigmetrics (2006)
8. Pastor-Satorras, R., Vazquez, A., Vespignani, A.: Dynamical and correlation properties of the internet. *Physical Review Letters* 87(25) (January 2001)
9. Smith, W., Foster, I.T., Taylor, V.E.: Predicting application run times using historical information. In: Feitelson, D.G., Rudolph, L. (eds.) *IPPS-WS 1998, SPDP-WS 1998, and JSSPP 1998*. LNCS, vol. 1459, pp. 122–142. Springer, Heidelberg (1998)

On Social Networks Reduction*

Václav Snášel¹, Zdeněk Horák¹, Jana Kočíbová¹, and Ajith Abraham²

¹ VSB Technical University Ostrava, Czech Republic
{`vaclav.snasel,zdenek.horak.st4,jana.kocibova.st1`}@vsb.cz

² Center of Excellence for Quantifiable Quality of Service
Norwegian University of Science and Technology, Norway
`ajith.abraham@ieee.org`

Abstract. Since the availability of social networks data and the range of these data have significantly grown in recent years, new aspects have to be considered. In this paper, we use combination of Formal Concept Analysis and well-known matrix factorization methods to address computational complexity of social networks analysis and clarity of their visualization. The goal is to reduce the dimension of social network data and to measure the amount of information, which has been lost during the reduction. Presented example containing real data proves the feasibility of our approach.

1 Introduction

As a **social network** we denote a set of subjects which are linked together by some kind of relationship. Social networking – in the sense of providing services to persons to stay in touch, communicate and express their relations – received great attention in the recent years.

Freeman in [6] underlines the needs for Social Networks Visualization and provides overview of the development of their visualization. The development from hand drawn images to complex computer-rendered scenes is evident. Also the shift from classical sociograms to new approaches and methods of visualization is evident. What remains is the need for clarity of such visualization.

As a specific kind of network data can be considered so-called **two-mode network data**. This data consists of two sets – set of subjects and set of events, which are, or are not, connected. Paper [7] introduces the usage of Formal Concept Analysis (FCA), a well-known general purpose data analysis method, in the area of social networks and reviews the motivation for finding relations hidden in data that are not covered by simple graph visualization. The paper shows that the **Galois lattice** is capable of capturing all three scopes of two-mode network data – relation between subjects, relation between events and also the relation between subjects and events.

* This research was supported in part by Czech Science Foundation (GACR) project 201/09/0990.

1.1 Complexity Aspects

As can be seen both from the mentioned paper and experiments presented below – with the increasing range of input data, the Galois lattice becomes soon very complicated and the information value decreases. Also the computational complexity grows quickly.

Comparison of computational complexity of algorithms for generating concept lattice can be found in [11]. As stated in the paper, the total complexity of lattice generation depends on the size of input data as well as on the size of output lattice. This complexity can be exponential. Important aspect of these algorithms is their time delay complexity (time complexity between generating two concepts). Recently published paper [4] describes linear time delay algorithm. In many applications it is possible to provide additional information about key properties interesting to the user, which can be used to filter unsuitable concepts during the lattice construction [1]. In some applications it is also possible to select one particular concept and navigate through its neighbourhood. These approaches allow us to manage larger scale of data, but cannot provide the whole picture of the lattice.

Many social network data can be seen as object-attribute data or simply matrix (binary and fuzzy). Therefore they can be processed using matrix factorization methods, which have been proven to be useful in many data mining applications dealing with large scale problems. Our aim is to allow processing of larger amount of data and our approximation approach is compatible with the two mentioned in the previous paragraph.

Clearly, some bit of information has to be forgotten, but we want to know, how close or far from the original result we are. The paper [15] introduces a method for measuring so-called normalized correlation dimension which can be seen as the number of independent variables in the dataset. This idea comes from the field of fractal dimension. Another way could be to directly compare the results from the original and reduced datasets. [3] introduces the modification of classical Lorenz curve to describe dissimilarity between presence-absence data.

Singular value decomposition has already been used in the field of social network data ([5]) to determine the position of nodes in the network graph. Next chapter of this paper reviews some basic notions of aforementioned theories. In the third chapter we describe our experiments in detail.

2 Preliminaries

2.1 Formal Concept Analysis

Formal concept analysis (shortly FCA, introduced by **Rudolf Wille** in 1980) is well known method for object-attribute data analysis. The input data for FCA we call **formal context** C , which can be described as $C = (G, M, I)$ – a triplet consisting of a set of objects G and set of attributes M , with I as relation of G and M . The elements of G are defined as objects and the elements of M as attributes of the context.

For a set $A \subseteq G$ of objects we define A' as the set of attributes common to the objects in A . Correspondingly, for a set $B \subseteq M$ of attributes we define B' as the set of objects which have all attributes in B . A **formal concept** of the context (G, M, I) is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$. $\mathcal{B}(G, M, I)$ denotes the set of all concepts of context (G, M, I) and forms a complete lattice (so called **Galois lattice**). For more details, see [9], [8].

Galois lattice may be visualized by so-called Hasse diagram. In this diagram, every node represents one formal concept from the lattice. Nodes are usually labeled by attributes (above the node) and objects (below the node) possessed by a concept. For the sake of clarity it is sometimes used so-called reduced labeling (see fig. 1 for illustration), which means that attributes are shown only at the first node (concept) they appear in. This holds reciprocally for objects. These two labelings are equivalent.

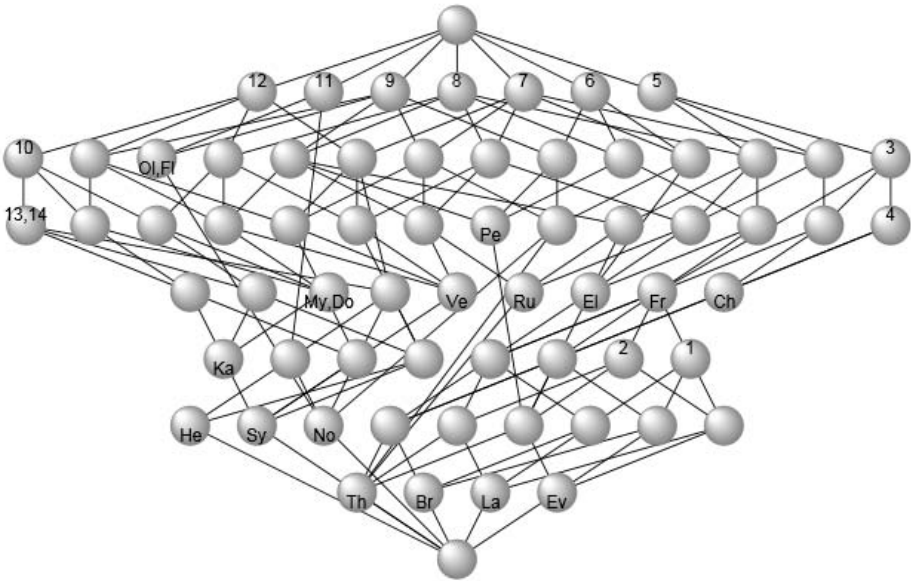


Fig. 1. Concept lattice before reduction

2.2 Non-negative Matrix Factorization

Matrix factorization methods decompose one – usually huge – matrix into several smaller. Non-negative matrix factorization differs by the use of constraints that produce non-negative basis vectors, which make possible the concept of a parts-based representation.

Common approaches to NMF obtain an approximation of V by computing a (W, H) pair to minimise the Frobenius norm of the difference $V - WH$. Let $V \in R^{m \times n}$ be a non-negative matrix and $W \in R^{m \times k}$ and $H \in R^{k \times n}$ for

$0 < k \ll \min(m, n)$. Then, the objective function or minimisation problem can be stated as $\min \|V - WH\|^2$ with $W_{ij} > 0$ and $H_{ij} > 0$ for each i and j . There are several methods for computing NMF. We have used the multiplicative method algorithm proposed by Lee and Seung [13], [12].

Description of Singular Value Decomposition (SVD) and Semidiscrete decomposition method is omitted due to the lack of space. For the purpose of our paper, these methods work in a similar way as the two mentioned above. Detailed explanation can be found in [14] and [10].

2.3 Correlation Dimension

The idea behind the Correlation dimension comes from the theory of Fractal dimension and is based on studying the distance between two random data points. Suppose we have a binary dataset D containing $|D|$ objects and K attributes. Consider random variable (denoted by Z_D) whose value is L_1 distance (attributes used as coordinates) between two randomly chosen objects from D . The distance varies from 0 (objects have the same attributes) to K (objects differ in all attributes). Now we can define the function $f : \mathbb{N} \rightarrow \mathbb{R}$ as $f(r) = \mathbb{P}(Z_D < r)$ and the set of points

$$\mathcal{I}(D, r_1, r_2, N) = \left\{ (\log r, \log f(r)) \mid r = r_1 + \frac{i(r_2 - r_1)}{N}, i = 0 \dots N \right\}.$$

The correlation dimension $\text{cd}_R(D, r_1, r_2, N)$ for a binary dataset D and parameters r_1, r_2 is the slope of the least-squares linear approximation \mathcal{I} . One would expect that the dimension of dataset with K independent attributes is K . To achieve this, we can normalize the result using random binary dataset having K independent variables such that the probability of i th variable being one is equal to the probability of randomly chosen object from dataset D having i th attribute. For more details see [15].

2.4 Lorenz Curves

To evaluate similarity we can use Lorenz Curves, an approach well-known from economy, in the way proposed in [3]. Let's suppose we have two presence-absence (binary) arrays $r = (x_i)_{i=1, \dots, N}$ and $s = (y_i)_{i=1, \dots, N}$ of dimension N . In the same manner as we normalize vectors, we can create arrays a_i and b_i by dividing each element of the array by their total sum. Formally $a_i = \frac{x_i}{T_r}, b_i = \frac{y_i}{T_s}, \forall i = 1, \dots, N$, where $T_r = \sum_{j=1}^N x_j$ and $T_s = \sum_{j=1}^N y_j$. Next, we can compute difference array $d = (d_i)_{i=1, \dots, N}$ as $d_i = a_i - b_i$, ordered from the largest value to the smallest one. By putting $c_i = \sum_{j=1}^i d_j$ we obtain the coordinates of the Lorenz similarity curve by joining the origin $(0, 0)$ with the points of coordinates $(\frac{i}{N}, c_i)_{i=1, \dots, N}$.

3 Experiments

3.1 Real-World Experiment

In our first example, we will use well known dataset from [2]. It contains information about participation of 18 women in 14 social events during the season. This participation can be considered as two-mode network or as formal context (binary matrix with rows as women and columns as social events). Visualization of this network as bipartite graph can be seen in the upper part of figure 3. Events are represented by nodes on the first row. These nodes are labeled by the event numbers. The second row contains nodes representing women and are labeled by two first letters of their names. Participation of the woman in the event is represented by edge between corresponding nodes. Illustration of the formal context (resp. binary matrix) can be seen in the left part of figure 4.

Now, let's describe the computed Galois lattice (figure 1). Each node in the graph represents one formal concept. Every concept is a set of objects (women in this case) and set of corresponding attributes (events). Edges express the ordering of concepts. Aforementioned reduced labeling is used here. The lattice contains all combinations of objects and attributes present in the data. One can easily read, that Sylvia participated in all events that Katherine did. Also everyone who participated in the events 13 and 14, also participated in the event 10. The reasons for these nodes to be separate, are the women Dorothy and Myrna that took part in the event 10, but not in the events 13 and 14.

After reduction. Due the high number of nodes and edges, many interesting groups and dependencies are hard to find. Now we will try to reduce the formal

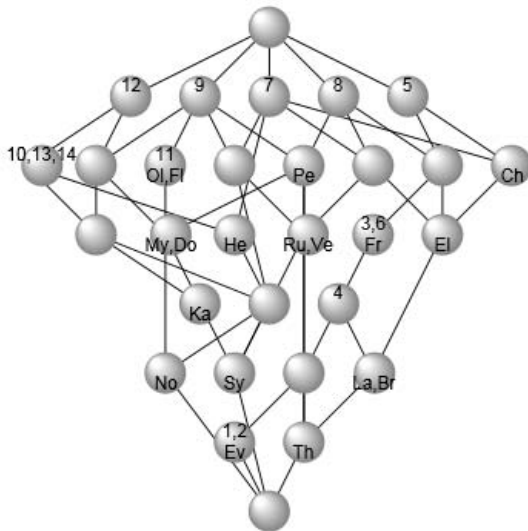


Fig. 2. Concept lattice at rank 5

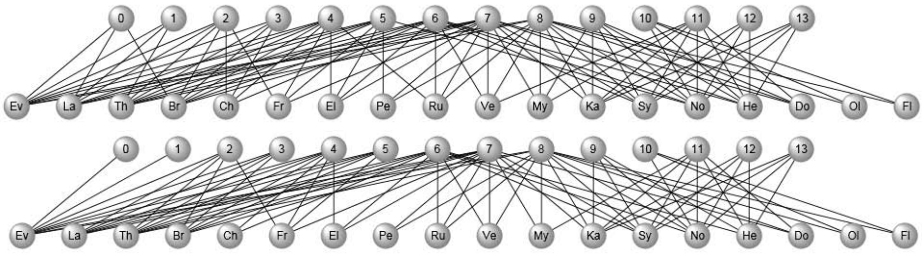


Fig. 3. Social network - before and after reduction to rank 5

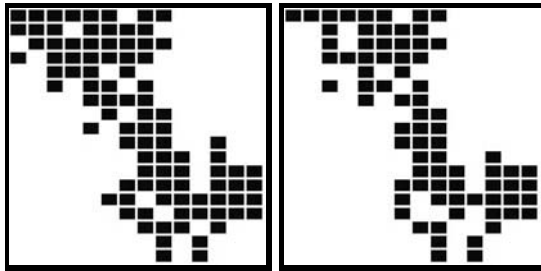


Fig. 4. Context visualization (original, rank 5)

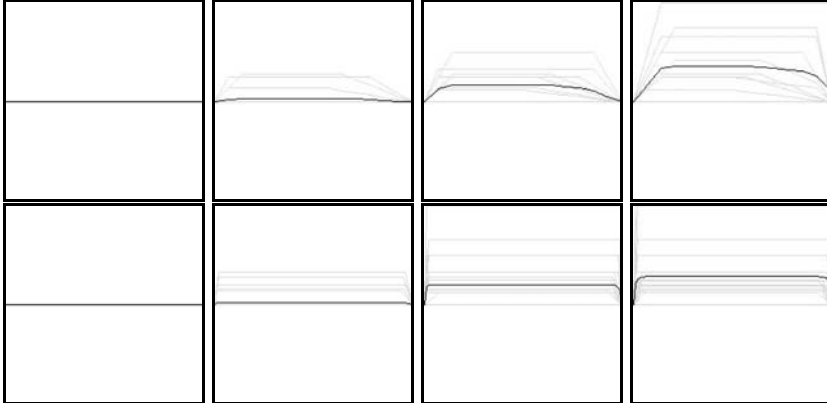


Fig. 5. Lorenz curve comparing contexts (first row) and lattices (second row) - before (first column) and after reduction to ranks 8, 5, 3 (remaining columns)

context to lower dimension and observe the changes. We have performed reduction of original 18x14 context to lower ranks and computed corresponding concept lattices. For illustration we have selected results obtained for rank 5 using NMF method. Modified context can be seen in the remaining part of figure 4. Visualization of network into bipartite graph (fig. 3) reveals some changes, but is still too complicated. The concept lattice can give us better insight. Detailed

look at the reduced lattice (fig. 2 for rank 5) shows, that the general layout has been preserved as well as the most important properties (e.g. mentioned implication about Sylvia and Katherine). The reduction to rank 5 caused merging of nodes previously marked by attributes 10, 13, 14 (which we have discussed earlier).

To illustrate the amount of reduction, we can compute similarity between the original and the reduced context and draw Lorenz curves (see first row of fig. 5). A larger area under the curve means higher dissimilarity (lower similarity). Because we compare the context using object-by-object approach, we obtain several curves (drawn using gray color on the figure). To simplify comparison, we have averaged these curves (result drawn using black color). In the same manner, we have computed these curves for formal concepts (second row of fig. 5).

3.2 Synthetic Datasets

To analyse results of described approach on larger data, we have generated synthetic binary dataset with 400 rows, 40 attributes and 20% density. This corresponds to two-mode network with 400 subjects and 40 events. Each subject participated at average in 8 events.

The table 1 contains results of this experiment. We have tested three different reduction methods - NMF, SVD and SDD. First column of each group contains the number of formal concepts in computed lattice. Different rows correspond to different ranks of reduction (first one contains information about original data). Second column contains normalized correlation dimension (ncd).

Since the original data have been created as uncorrelated, their normalized correlation dimension is close to the number of columns. The reduction tries to resemble the original data maximally, so it often preserves repeatedly appearing patterns. Therefore we expect ncd to decrease during the rank reduction. Computed results verify this expectation.

To estimate roughly the ratio of reduction, one does not have to compute the whole original lattice. The normalized correlation dimension – which is computed more rapidly and using formal context only – can be used to do this. Since the

Table 1. Reduction progress for synthetic dataset (400x40)

	NMF		SVD		SDD	
	$ \mathcal{B} $	ncd	$ \mathcal{B} $	ncd	$ \mathcal{B} $	ncd
original	15477	39	15477	39	15477	39
rank 35	10672	43	15459	39	7750	31
rank 30	5429	35	15127	38	4747	23
rank 25	2665	30	14621	28	2824	23
rank 20	1016	25	11831	29	1377	17
rank 15	348	21	7288	19	514	14
rank 10	149	17	3322	10	169	8
rank 5	56	6	526	6	4	4

probabilistic nature of ncd computation, we can expect more precise results for datasets containing larger number of objects.

4 Conclusions

We have seen that Galois lattice is suitable for displaying dependencies in two-mode network data. The restrictive factor is the size and inner structure of input data. Using matrix factorization methods, we can simplify the structure to allow better insight into the data, but still to retain the most important properties.

This approach has potentially many uses - for example generating fast preview of large social network data, approximate analysis of huge World Wide Web data where exact analysis is computationally unmanageable, etc. As we have mentioned in the introduction, the complexity of many algorithms involved in concept lattice analysis bears (e.g. linearly or exponentially) upon the number of concepts. Therefore the ratio of concepts reduction gives us directly (knowing the complexity of used algorithm) the speed up of computation.

Important fact is, that the progress of reduction is gradual. Every small change in the formal context made during the reduction can be seen as a small change in the corresponding concept lattice. User may be also involved to decide what amount of reduction is suitable to his purpose. Taking the changes in the context into account, we are also able to (considering all the changes made in affected objects and attributes) reconstruct the unreduced concept lattice or its part. This fact maybe useful when more precise result are needed at the detailed level.

From the results it may look like SVD being the best method for reduction, because for fixed rank it gives more concepts than other methods. However, the situation is not that simple. For example the NMF, due the mentioned parts-based representation, uses more natural and independent factors and therefore gives more intuitive results. Thus in our future work we would like to analyse the effects of different reduction methods in detail and illustrate the usage on practical problems.

References

1. Belohlavek, R., Sklenar, V.: Formal concept analysis constrained by attribute-dependency formulas ICFCA. In: Ganter, B., Godin, R. (eds.) ICFCA 2005. LNCS (LNAI), vol. 3403, pp. 176–191. Springer, Heidelberg (2005)
2. Davis, A., Gardner, B.B., Gardner, M.R.: *Deep South: A Social Anthropological Study of Caste and Class*. University of Chicago Press, Chicago (1965)
3. Egghe, L., Rousseau, R.: Classical retrieval and overlap measures satisfy the requirements for rankings based on a Lorenz curve. *Information Processing and Management* 42, 106–120 (2006)
4. Farach-Colton, M., Huang, Y.: A linear delay algorithm for building concept lattices. In: Ferragina, P., Landau, G.M. (eds.) CPM 2008. LNCS, vol. 5029, pp. 204–216. Springer, Heidelberg (2008)
5. Freeman, L.C.: *Graphical Techniques for Exploring Social Network Data*. In: *Models and Methods in Social Network Analysis* (2005)

6. Freeman, L.C.: Visualizing social networks. *Journal of social structure* 1 (2000)
7. Freeman, L.C., White, D.R.: Using Galois Lattices to Represent Network Data. *Sociological Methodology* 23, 127–146 (1993)
8. Ganter, B., Wille, R.: Applied Lattice Theory: Formal Concept Analysis. In: Grätzer, G.A. (ed.) *General Lattice Theory*, pp. 592–606. Birkhäuser, Basel (1997)
9. Ganter, B., Wille, R.: *Formal Concept Analysis: Mathematical Foundations*. Springer, New York (1997)
10. Kolda, T.G., O’Leary, D.P.: A semidiscrete matrix decomposition for latent semantic indexing information retrieval. *ACM Transactions on Information Systems (TOIS)* 16, 322–346 (1998)
11. Kuznetsov, S.O., Obedkov, S.A.: Comparing Performance of Algorithms for Generating Concept Lattices. *Journal of Experimental and Theoretical Artificial Intelligence* 14, 189–216 (2002)
12. Lee, D., Seung, H.: Algorithms for Non-Negative Matrix Factorization. *Advances in Neural Information Processing Systems* 13, 556–562 (2001)
13. Lee, D., Seung, H.: Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 788–791 (1999)
14. Letsche, T., Berry, M.W., Dumais, S.T.: Computational methods for intelligent information access. In: *Proceedings of the 1995 ACM/IEEE Supercomputing Conference* (1995)
15. Tatti, N., Mielikainen, T., Gionis, A., Mannila, H.: What is the dimension of your binary data? In: *Proceedings of the Sixth International Conference on Data Mining*, pp. 603–612 (2006)

Networks Consolidation through Soft Computing

Sami Habib, Paulvanna Nayaki Marimuthu, and Mohammad Taha

Kuwait University,
Computer Engineering Department,
P.O. Box 5969, Safat 13060, Kuwait
sami.habib@ku.edu.kw

Abstract. This paper reports the application of soft computing in redesign operations on the customers' clusters (sub-networks), since many customers maybe initially located in a cluster with less intra-cluster traffic. Here we assume an existing network with reconfigurable architecture and we propose a number of redesign techniques to reduce the extra-traffic and maximize the intra-traffic within the clusters by considering customers' movement and clusters consolidation. Furthermore, the proposed search approach is based on Genetic Algorithm (GA) with an object-oriented chromosome representation. Our experimental results for a network size of 50 customers show an average of 22% reduction in the extra-traffic through the proposed redesign operations.

Keywords: Clusters consolidation, redesign, optimization, soft computing, Genetic Algorithm.

1 Introduction

One of the major challenges facing companies today is to meet increasing customer demands due to the higher bandwidth multimedia applications. This necessitates an incremental change in the intra-organization of the network. Network redesign is a difficult combinatorial optimization problem, which searches for good network rearrangements that satisfy customers' geographical constraints and customer traffic constraints from a large number of solutions.

We have formulated the redesign of the existing network through clusters consolidation as an optimization problem, where the objective function is to minimize the extra-traffic subject to a set of constraints. Since the problem is known as an NP-complete problem; therefore, Genetic Algorithm (GA) algorithm is employed to search for optimal solutions.

In this paper, we have studied the simulation of various re-clustering operations such as adding new cluster, deleting the existing cluster, and relocating the customers to other clusters by move and swap operations on customers with a dual emphasis on minimizing the extra-traffic and improving the distribution of customers in the network. The simulation results of various clusters consolidation operations show that Genetic Algorithm gives more freedom with its operations with respect to re-clustering operations and the proposed re-clustering operations improve the network utilization by reducing the extra-traffic around 22%, thereby reducing the maintenance cost of the existing network.

2 Related Work

Network Redesign/Reconfiguration is a collective term of various operations namely, relocation of customers among the existing clusters, relocation and replication of servers, consolidation of servers and clusters. In the case of communication networks, research is going on in the areas like optimal placement of network resources, such as servers, gateways, network managers etc., server splitting, server relocation in an optimal position conscious to minimize the traffic load, delay and network cost.

Rodolakis et al. [3] presented a pseudo-polynomial approximation algorithm for placing replicated servers with QOS constraints. They focused on the problem of maximizing the performance by minimizing the cost of the communication network.

Giladi et al. [4] solved the problem of placing various network resources (servers, gateways, mainframes, network managers etc.) in an optimal way into an existing small network. They proposed a heuristic greedy algorithm based on separate optimal solution for each resource. They provided a set of analytical tool for finding the optimal location of the source to be added, without changing the existing topology.

Shim et al. [5] described an optimum way to relocate the server in a distributed network by considering the total communication delays between clients and servers. Spellman et al. [6] evaluated the effect of server consolidation by relocating an application and its associated servers as a group. They also analyzed the network performance by physical consolidation of servers. Montana and Hussain [7] made a preliminary study on dynamic redesign of a network with reconfigurable links using genetic algorithm.

Soft computing is the fusion of methodologies that were designed to model and enable solutions to real world problems, which are difficult to model mathematically. Genetic Algorithm is an evolutionary algorithm [8], which has been employed by various researchers [9][10][11][12] to solve optimization problems in the topology design of local area network (LAN), autonomous transfer (ATM) network, backbone design of communication network and optical network. Since GA consists of various components and parameters that can be modified, researchers [13][14][15] have studied the impact of various factors including problem encoding, cross over and mutation operators, population size, cross over and mutation rate etc.

The detailed literature survey reveals that most of the research work addresses the network topology design issues and few on server or customer replacement with and without using GA. In this paper, we have proposed network redesign operations, such as, moving and swapping the customers among the clusters, and adding and deleting the clusters. We have employed Genetic Algorithms based approach to search for an optimized solution in each redesign operation.

3 Network Model

We view an enterprise network as a group of local area networks (LANs), where each LAN clusters a set of customers. The overall network model comprised of 2-level with LAN clusters and backbone as illustrated in Figure 1.

Our enterprise example is based on 50 customer nodes, with prior traffic pattern, distributed uniformly in an area of 50m x 50m. The customers are grouped into 8

clusters, and Table 1 describes the distribution of customers inside the clusters of the defined network. The traffic matrix describing the volume of traffic between each pair of customer is generated randomly with zeros only in the main diagonal and it describes the ideal network condition at the initial design time. The given network model, with known traffic inflow and outflow for each cluster reflects the practical environment for simulating the various redesign operations.

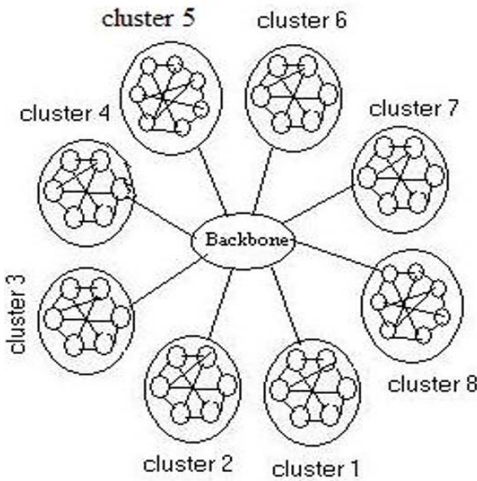


Fig. 1. Network model

Table 1. Distribution of customers

Cluster Identification	Number of Customers
Cluster 1	6
cluster 2	6
cluster 3	6
cluster 4	6
cluster 5	7
cluster 6	6
cluster 7	6
cluster 8	7

4 Genetic Algorithm Overview

The theory of Genetic Algorithms (GA), which are using the mechanism of evolution and natural selection as a problem solver, date back only to 1975 [1]. The basic structure of GA is a well-known for its simplicity, as shown in Figure 2. However GA is a powerful search technique, which is used to solve many combinatorial problems [2].

GA starts with an initial population $P(t=0)$ of solutions encoded as chromosomes (line 3). An initial population is most often generated randomly but a heuristic method can be used to generate the initial population. Each chromosome is made of a sequence of genes and every gene controls the inheritance of specific attributes of the solution's characteristics. A fitness function in lines 4 and 11 measures the quality of the chromosomes. A fit chromosome suggests a better solution. The while-loop in lines 5-12 represents the evolution process, where relatively fit chromosomes reproduce new chromosomes and inferior chromosomes die. This process continues until a chromosome with desirable fitness is found or a number of generations had been passed since the beginning of the evolution process. Line 8 selects the best chromosomes within the current generation based on their fitness values. These selected

chromosomes, known as parents, are used to reproduce the next generation of chromosomes, known as offspring. The evolution process involves two generic operations namely, mutation in line 9 and crossover in line 10.

```

Algorithm Genetic( )
1 Begin
2 t=0;
3 initialize Chromosomes P(t);
4 evaluate Chromosomes P(t);
5 while (not stopping conditions met) do
6   Begin
7     t=t+1;
8     Select P(t) from P(t-1);
9     mutate some of P(t);
10    crossover some of P(t);
11    evaluate Chromosomes P(t);
12  End
13 End

```

Fig. 2. An overview of Genetic Algorithm

A mutation operator arbitrarily alters one or more genes of a randomly selected chromosome. The intuition behind the mutation operator is to add missing gene to the population. On the other hand a crossover operation combines features of two selected chromosomes (parents) to form two chromosomes (offspring) by swapping genes of the parent's genes. The intuition behind the crossover operator is to exchange information between different potential solutions. To apply Genetic Algorithm for finding the good solutions of computer network redesign problem, a set of possible solutions are generated based on the existing network infrastructure by varying its initial genes (features).

5 Mathematical Model

We have formulated the clusters consolidation problem as an optimization problem, where the objective function is to minimize the extra-traffic as stated in Equation (1). The term $\Psi_{i,j}$ represents the traffic flow from customer i to customer j belonging to different clusters. Constraint (2) indicates that the total number of clusters within the network should be bound between 2 clusters and the ratio of N over 2 to the nearest integer, where N represents the total number of customers in the network. Constraint (3) ensures that a customer is bound to a single cluster. Constraint (4) guarantees that the summation of all customers in all the clusters should be equaled to N . Thus, there is no overlooked customer and no added new customer. Constraint (5) makes certain that the total cable length within a cluster is greater than zero and less than or equal to a given threshold values, D_{TH} . The function $COM()$ estimates the center-of-mass of a given cluster; moreover, the function $Dist()$ calculates the Euclidian distance from a customer to its center-of-mass. Constraint (6) ensures that the total number of bound customers to a cluster should be balanced.

$$Min \sum_{\substack{i \in C_k \text{ and } j \in C_l \\ i \neq j \text{ and } k \neq l}} \psi_{i,j} \tag{1}$$

subject to

$$2 \leq \sum_{k=1} C_k \leq \left\lfloor \frac{N}{2} \right\rfloor \tag{2}$$

$$\sum_{k=1}^{\left\lfloor \frac{N}{2} \right\rfloor} \alpha_{i,k} = 1 \text{ for } i = 1, 2, \dots, N \tag{3}$$

$$\sum_{k=1}^{\left\lfloor \frac{N}{2} \right\rfloor} |C_k| = N \tag{4}$$

$$0 < \sum_{i \in C_k} Dist((x_i, y_i), COM(C_k)) \leq D_{TH} \text{ for } k = 1, 2, \dots, \left\lfloor \frac{N}{2} \right\rfloor \tag{5}$$

$$2 \leq \sum_{i \in C_k} \alpha_{i,k} \leq \left\lfloor \frac{N}{2} \right\rfloor \text{ for } k = 1, 2, \dots, \left\lfloor \frac{N}{2} \right\rfloor \tag{6}$$

6 Network Redesign Methodology

The proposed network redesign methodology carries on the redesigns by using a number of operations, such as moving and swapping of customers and also the consolidation of existing clusters such as adding and deleting of clusters. The various mutation operations explained in Figure 3 to Figure 6, on the randomly selected customers and the randomly selected clusters simulate the redesign process. Here S represents the set of clusters that it is estimated by Constraint (2); moreover, N represents the total number of nodes in the network.

```

Operation MOVECUSTOMER(S, N)
1  begin
2  select src_cluster = select_random (S);
3  select customer = select_random (src_cluster);
4  select dest_cluster = select_random (S);
5  move customer from src_cluster to dest_cluster;
6  end
    
```

Fig. 3. Pseudo code for customer moving operation

The relocation of customers by move operation has been implemented by the random selection of two clusters, followed by the random movement of a node among them as shown in Figure 3. Two customers frequently communicating with each other in different clusters may be placed together in a single cluster to reduce the traffic, using *swap* operation, as presented in Figure 4. A new cluster has been added by *add-cluster*, by the random placement of one customer from each cluster, as illustrated in

Figure 5. Cluster with predominantly less traffic can be deleted by *delete-cluster* and the customers in that cluster to be distributed to the remaining clusters randomly, as illustrated in Figure 6.

```

Operation SWAPCUSTOMER(S, N)
1  begin
2    select cluster1 = select_random (S);
3    select customer1 = select_random (cluster1);
4    select cluster2 = select_random (S);
5    select customer2 = select_random (cluster2);
6    swap customers 1-2 between the clusters 1-2;
7  end

```

Fig. 4. Pseudo code for customer swapping operation

```

Operation ADDCLUSTER(S, N)
1  begin
2    If (number of clusters < N/2)
3      create an empty cluster;
4      for each cluster
5        begin
5          If( number of customers within selected cluster > 2)
6            select a customer at random;
7            move selected customer to the new cluster;
8          end
9  end

```

Fig. 5. Pseudo code for adding a new cluster operation

```

Operation DELETECLUSTER(S, N)
1  begin
2    If ( $S \geq 2$ )
3      select cluster1 = select_random(S);
4      for each customer in cluster1;
5        begin
6          select dest_cluster = select_random(S) - cluster1;
7          move customer from cluster1 to dest_cluster;
8        end
9      delete cluster1;
10 end

```

Fig. 6. Pseudo code for delete-cluster operation

7 Experimental Results

We have coded the optimization problem of clusters consolidation within the Genetic Algorithm, using Microsoft.net Framework; moreover, all our six experiments were

run on a typical desktop computer. The first experiment examined the proposed re-design operations in the original network infrastructure. The experiments 2 to 4 looked at the reduction at traffic and the effect of redesign operations on the reduced network infrastructure. The experiments 5 and 6 showed the effect of varying the mutation rate in original traffic. We started with the network (Figure 1), having heavy extra-traffic of around 120,000 Kbits/s. The goal is to optimize the extra-traffic in each redesign operation using GA approach. Constraints are added with the algorithm to maintain a minimum of 2 clusters in the network and 2 customers in each cluster at all times. In this study, we selected the initial population to be 100. The chromosome selection ratio for the mutation operation was 50%, the number of generations to be 2000, and the mutation rate as 20%.

We have simulated the optimal behavior of our algorithm from the reduction of the extra-traffic in various proposed re-clustering operations. Figure 7 describes the changes in the extra-traffic, after various mutation operations in the original traffic matrix, described in section 3.

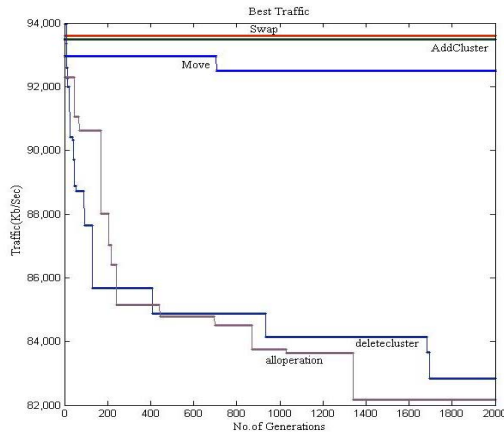


Fig. 7. Results of simulation of various redesign operations implemented in the original traffic matrix with mutation rate of 0.20

It has been observed that extra-traffic decreased to the maximum of 31% for cluster consolidation operations such as delete cluster and combination of all proposed operations together. In other operations, the decrease is around 23%. The stopping condition of the algorithm is further refined by the addition of distance constraint threshold value, which is 400m for each cluster.

After repeated execution of the algorithm, we managed to end up with 60% reduction in the number of clusters, in delete cluster operation. In all the categories, our algorithm shows better performance by the reduction in extra-traffic. A minimum of 22% decrease in extra-traffic, along with 25% reduction in the number of clusters has been observed in add-cluster operation. It is also observed that for swapping customers and adding clusters operations, the algorithm converges at an earlier stage. It is justified by the fact that the random distribution of customers to the newly added

clusters and to different clusters in adding cluster and swapping operations respectively, may result in a probability of separating the customers in the same cluster with heavy intra-traffic. This separation may cause less decrease in the extra-traffic compared to other operations. We tested the robustness of the proposed algorithm by varying the network traffic load. The network traffic was decreased by 5%, 10% and 15% from the original traffic. Figures 8-10 describe the behavior of the algorithm for various input traffic matrix. The algorithm is repeated for each volume of traffic.

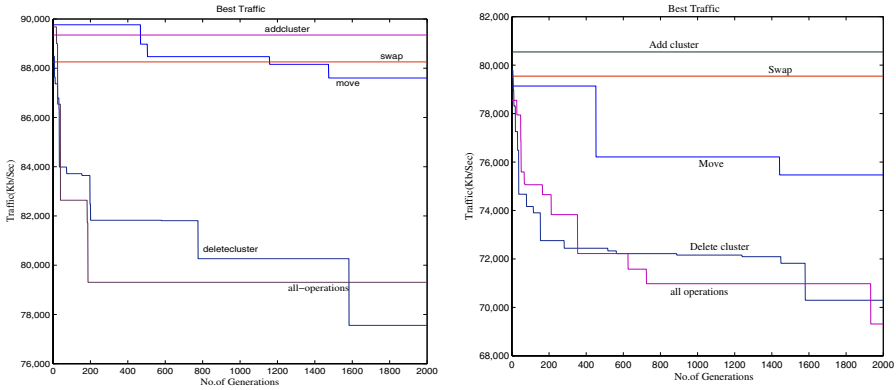


Fig. 8–9. Behavior of the proposed algorithm for reduced volume of traffic (5%) and (10%) with mutation rate of 0.20

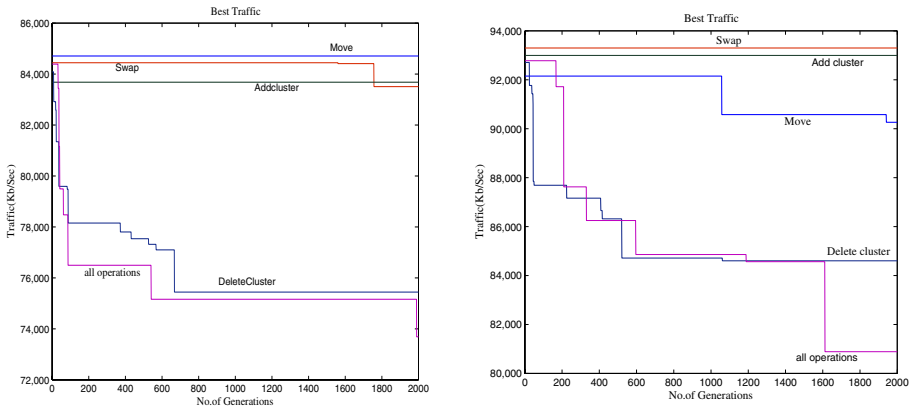


Fig. 10–11. Behavior of the proposed algorithm for reduced volume of traffic (15%) and (10%) with mutation rate of 0.20 (Figure 11. decreased mutation rate (0.10) in original traffic)

The simulation runs were also repeated for various mutation rates. The mutation rate was reduced by 50% and 75% from the initial mutation rate (0.20). By reducing the mutation rate by 50%, say 0.10, we observed a slight decrease in the extra traffic of around 2%, as explained by Figure 11. Further reducing it by 25% (Figure 12), say

0.05, the extra traffic is found to increase slightly around 1% and the increase was maximum (3.5%) for the combined mutation operations. Hence, from the simulation results, we observed that our algorithm shows optimal solution for the mutation rate of 0.10. From Figure 11 and figure 12, we observed that the decrease in the mutation rate decreases the rate of convergence of the algorithm at earlier stage, which showed additional decrease in the extra-traffic.

The simulation results showed that the genetic algorithm with distance constraint based traffic matrix makes our algorithm suitable for automated redesign operation of any existing network.

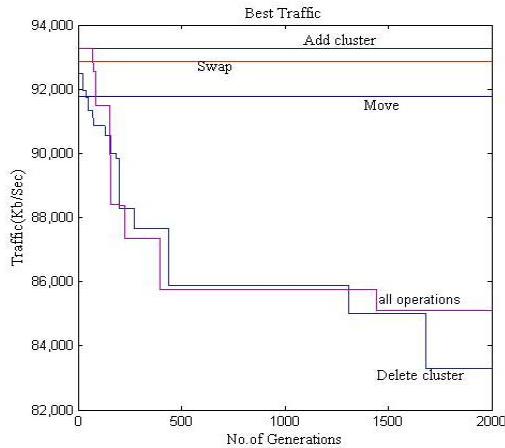


Fig. 12. Behavior of the proposed algorithm with decreased mutation rate (0.05) in original traffic

8 Conclusions and Future Work

In this paper, we studied the behavior of the proposed new redesign operations for an enterprise network with known customers' locations and traffic pattern. We represented the initial solution in GA using an object-oriented chromosome representation. We also tested the robustness of the algorithm by varying the mutation rate. With the given initial network configuration, the algorithm showed good results for the mutation rate of 0.10 with minimal the extra-traffic to the backbone. Our present work can be improved by including more re-cluster operations and also by parametrical variation of the optimization algorithm.

Acknowledgement

The authors would like to acknowledge the support by Kuwait Foundation for the Advancement of Sciences (KFAS) under a research grant no. 2006-1510-03.

References

1. Holland, J.: *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge (1975)
2. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Heidelberg (1994)
3. Rodolakis, G., Siachalou, S., Georgiadis, L.: Replicated Server Placement with QoS Constraints. In: Ajmone Marsan, M., Bianchi, G., Listanti, M., Meo, M. (eds.) *QoS-IP 2004*. LNCS, vol. 3375, pp. 207–220. Springer, Heidelberg (2005)
4. Giladi, R., Korach, E., Ohayon, R.: Placement of Network Resources in Communication Networks. *Computer Networks* 23, 195–209 (2003)
5. Shim, J., Lee, T., Lee, S.: A server placement algorithm conscious of communication delays and relocation costs. In: *Workshop on Networking, Italy*, pp. 83–89 (2002)
6. Spellmann, A., Erickson, K., Reynolds, J.: Server Consolation Using Performance Modeling. *IEEE computer*, 31–36 (2003)
7. Montana, D., Hussain, T.: Adaptive Reconfiguration of Data Networks Using Genetic Algorithm. *Applied Soft Computing* 4(4), 433–444 (2004)
8. Srinivas, M., Patnaik, L.M.: Genetic Algorithms: A Survey. *IEEE Computer Magazine*, 17–26 (1994)
9. Elbaum, R., Sidi, M.: Topological Design of LANs using Genetic Algorithms. *IEEE/ACM Transactions on Networking* 4(5), 766–779 (1996)
10. Tang, K., Ko, K., Man, K.F., Kwong, S.: Topology Design and Bandwidth Allocation of Embedded ATM Networks Using Genetic Algorithm. *IEEE Communication Letters* 2(6), 171–173 (1998)
11. Webb, A., Turton, B.C.H., Brown, J.M.: Application of Genetic Algorithm to a network optimization problem. In: *IEE Conference on Telecommunications*, Edinburgh, UK, pp. 62–66 (1998)
12. Roy, K., Naskar, K.M.: Genetic Evolutionary Algorithm for Static Traffic Grooming to SONET over WDM Optical Networks. *Computer Communications* 30, 3392–3402 (2007)
13. Dengiz, B., Altiparmak, F., Smith, A.E.: Local Search Genetic Algorithm for Optimal Design of Reliable Networks. *IEEE Transactions on Evolutionary Computation* 1, 179–188 (1997)
14. Din, D.R., Chiu, Y.S.: A Genetic Algorithm for Solving Virtual Topology Reconfiguration Problem in Survivable WDM Networks with Reconfiguration Constraint. *Computer Communications* 31, 2520–2533 (2008)
15. Chou, H., Premkumar, G., Chu, C.: Genetic Algorithms for Communications Network Design - an Empirical Study of the Factors that Influence Performance. *IEEE Transactions on Evolutionary Computation* 5(3), 236–249 (2001)

Lacking Labels in the Stream: Classifying Evolving Stream Data with Few Labels

Clay Woolam, Mohammad M. Masud, and Latifur Khan

Department of Computer Science, University of Texas at Dallas
{clayw,mehedy,lkhan}@utdallas.edu

Abstract. This paper outlines a data stream classification technique that addresses the problem of insufficient and biased labeled data. It is practical to assume that only a small fraction of instances in the stream are labeled. A more practical assumption would be that the labeled data may not be independently distributed among all training documents. How can we ensure that a good classification model would be built in these scenarios, considering that the data stream also has evolving nature? In our previous work we applied semi-supervised clustering to build classification models using limited amount of labeled training data. However, it assumed that the data to be labeled should be chosen randomly. In our current work, we relax this assumption, and propose a label propagation framework for data streams that can build good classification models even if the data are not labeled randomly. Comparison with state-of-the-art stream classification techniques on synthetic and benchmark real data proves the effectiveness of our approach.

1 Introduction

Data stream classification has gained increasing attention in recent years because large volumes of data are being generated continuously in different domains of knowledge. Data stream classification poses several challenges because of fundamental properties: infinite length and evolving nature. Stream evolution may occur in two ways. First, a new class of data may evolve in the stream that has not been seen before. This phenomenon will be referred to henceforth as *concept-evolution*. Second, the underlying concepts of the data may change. This will be referred to as *concept-drift*. Many solutions have been proposed to classify evolving data streams [1,2,3,4,5,6]. However, most of those techniques assume that as soon as a data point (or a batch of data points) has arrived in the stream and classified by the classifier, that data point (or the batch of data points) would be labeled by an independent labeling mechanism (such as a human expert), and can be used for training immediately. This is an impractical assumption, because in a real streaming environment, it is far beyond the capability of any human expert to label data points at the speed at which they arrive in the stream. Thus, a more realistic assumption would be that only a fraction of the instances would be labeled for training. This assumption was first made by us in our previous work [7].

In the previous work, we [7] proposed a technique to train classification models with $P\%$ randomly chosen labeled data from each chunk. So, if a training data chunk contained 100 instances, then the algorithm required only P labeled instances. However, the prediction accuracy of the trained model in this technique may vary depending on the quality of the labeled data. That is, this approach should work better on a sample that is uniformly distributed in the feature space rather than a biased, non-uniform sample. In our current work, we propose a more robust technique by making no prior assumption about the uniformity of the labeled instances. Our only requirement is that there should be some labeled instances from each class.

Our ensemble classification technique works as follows. First, we classify the latest (unlabeled) data chunk using the existing ensemble. Second, when $P\%$ of instances in the data chunk have been labeled, we apply constraint-based clustering to create K clusters and split them into homogeneous clusters (micro-clusters) that contain only unlabeled instances, or only labeled instances from a single class. We keep a summary of each micro-cluster (e.g. the centroid, number of data points etc.) as a “pseudo-point” and discard all the raw data points in order to save memory and achieve faster running time. Finally, we apply a label propagation technique on the pseudo-points to label the unlabeled pseudo-points. These labeled pseudo-points act as a classification model. This new model replaces an old model in the ensemble if necessary and the ensemble is kept up-to-date with the current concept. We also periodically refine the existing models to cope with stream evolution.

This paper details several contributions. First, we propose a robust stream classification technique that works well with limited amount of labeled training data. The accuracy of our classification technique is not dependent on the quality of the labeled data. Second, we suggest an efficient label propagation technique for stream data. This involves clustering training instances into pseudo-points and applying label propagation on the pseudo-points. To the best of our knowledge, no label propagation technique exists for data streams. Third, in order to handle concept-evolution and concept-drift, we introduce pseudo-point injection and deletion techniques and analyze their effectiveness both analytically and empirically. Finally, we apply our technique to synthetic and real data streams and achieve better performance than other data stream classification techniques that use limited labeled training data.

The paper is organized as follows: section 2 discusses related works, section 3 presents an overview of the whole process, section 4 describes the training process, section 5 discusses the ensemble technique, section 6 explains the experiments and analyzes the results, and section 7 concludes with directions to future works.

2 Related Work

Our work is closely related to both data stream classification and label propagation techniques. We explore both of these methods below.

Data stream classification techniques can be divided into two major categories: single model and ensemble classification. Single model classification techniques

apply incremental learning so that the models can be updated as soon as new training data arrives [8,9]. The main limitation of these single model techniques is that only the most recent data is used to update the model, and so, the influence of historical data is quickly forgotten. Other single model approaches like [2,6] have been proposed to handle concept-drift efficiently.

Ensemble techniques like [3,4,5,10,11] can update their models efficiently and cope with the stream evolution effectively. We also follow an ensemble approach that is different from most other ensemble approaches in two aspects. First, ensemble techniques like [4,5,10] mainly focus on building an efficient ensemble whereby the underlying classification technique, say decision tree, is a blackbox. We concentrate on building an efficient learning paradigm rather than focusing on the ensemble construction. Second, most of the previous classification techniques assume that all instances in the stream will eventually be labeled and can be used for training, but we assume that only a fraction, like 10%, will be labeled and be available for training. In this regard, our approach is related to our previous work [7], which will henceforth be referred as SmSCLuster.

Our current approach is different from SmSCLuster, the previous approach, in several aspects. First, it was assumed in SmSCLuster that the labeled instances would be uniformly distributed, which may not be the case in a real world scenario. We do not make any such assumption. Second, SmSCLuster applied only semi-supervised clustering to build the pseudo-points, but did not apply cluster splitting, pseudo-point deletion, or label propagation. Finally, SmSCLuster applied K -Nearest Neighbor classification, whereas we apply inductive label propagation for classification.

Algorithm 1. LabelStream

Input: \mathcal{X}^n : data points in chunk D_n

K : number of pseudo-points to be created

M : current ensemble of L models $\{M_1, \dots, M_L\}$

Output: Updated ensemble M

1. Predict the class labels of each instance in \mathcal{X}^n with M (section 5).
- /* Assuming that $P\%$ instances in D_n has now been labeled */
2. $M' \leftarrow \mathbf{Train}(D_n)$ /* Build a new model M' */
3. $M \leftarrow \mathbf{Refine-Ensemble}(M, M')$ (section 5.1)
4. $M \leftarrow \mathbf{Update-Ensemble}(M, M', D_n)$ (section 5.2)

Function $\mathbf{Train}(D_n)$ Returns Model

- 2.1. Set of macro-clusters, $\mathcal{MC} \leftarrow \mathbf{Semi-supervised-Clustering}(D_n)$ (section 4.1)
 - 2.2. Set of micro-clusters, $\mu\mathcal{C} \leftarrow \mathbf{Build-Micro-clusters}(\mathcal{MC})$ (section 4.1)
 - 2.3. **for each** micro-cluster $\mu\mathcal{C}_i \in \mu\mathcal{C}$ **do** pseudo-point $\psi_i \leftarrow \mathbf{Summary}(\mu\mathcal{C}_i)$
 - 2.4. $M' \leftarrow$ Set of all pseudo-points ψ_i
 - 2.5. $M' \leftarrow M' \cup_{t=n-r}^{n-1}$ Set of all labeled pseudo-points in Chunk D_t
 - 2.6. $M' \leftarrow \mathbf{Propagate-Labels}(M')$
 - 2.7. **return** M'
-

3 Overview of the Approach

Algorithm 1 summarizes the overall process. Line 2 executes the Train operation on an incoming data chunk. Training begins with a semi-supervised clustering technique. The clusters are then split into pure micro-clusters in line 2.2 of the training function. Then, the summary of each micro-cluster is saved as a pseudo-point in line 2.3. In line 2.4 and 2.4, we combine our new set of pseudo-points with the labeled pseudo-points from the last r contiguous chunks. By a labeled pseudo-point we mean the pseudo-points that correspond to only the manually labeled instances. In line 2.6, a modified label propagation technique, [12], is applied on the combined set of pseudo-points. Once we complete training a new model, we return to the main algorithm. In line 3 and 4, the ensemble is refined and updated. The following sections describe this process in detail.

4 Model Generation

The training data is a mixture of labeled and unlabeled data. Training consists of three basic steps. First, semi-supervised clustering is used to build K clusters, denoted as **macro-clusters**, from the training data. Second, to build homogeneous clusters, denoted as **micro-clusters**, in order to facilitate label propagation process, and save cluster summaries as **pseudo-points**. Third, to propagate labels from the labeled pseudo-points to the unlabeled pseudo-points, a transductive label propagation algorithm, from [12], is used. The collection of the labeled pseudo-points are used as a classification model for classifying unlabeled data. The classification and ensemble updating process is described in the next section, section 5.

4.1 Semi-supervised Clustering

With semi-supervised clustering, clusters can be built efficiently in terms of both running time and storage space. The label propagation algorithm takes $O(n^3)$ time in a dataset having n instances. Although this running time is tolerable in a static environment, it may not be practical for a streaming environment where fast training is a critical issue. Training time is reduced by reducing the number of instances to a constant K . This is done by partitioning the instances into K clusters and using the cluster centroids as pseudo-points. This also reduces memory consumption because rather than storing the raw data points, we store the pseudo-points only. Thus, the storage requirement goes from being linear to constant.

The semisupervised clustering objective is to minimize both cluster impurity and intra-cluster dispersion, expressed by

$$\mathcal{O}_{ImpDisp} = \sum_{i=1}^K \left(\sum_{\mathbf{x} \in \mathcal{X}_i} \|\mathbf{x} - \mathbf{u}_i\|^2 + \sum_{\mathbf{x} \in \mathcal{L}_i} \|\mathbf{x} - \mathbf{u}_i\|^2 * (\mathcal{L}_i)^2 * Gini_i * Ent_i \right) \quad (1)$$

where K is the total number of clusters, μ_i is the centroid of cluster i , \mathcal{X}_i is the set of instances belonging to cluster i , Imp_i is the impurity of cluster i , \mathcal{L}_i is the set of labeled instances in cluster i , and $|\mathcal{L}_i|$ is the corresponding cardinality, $Gini_i$ is the Gini index of cluster $i = \sum_{c=1}^C (p_c^i)^2$, C being the total number of classes in the dataset, and Ent_i is the entropy of cluster $i = \sum_{c=1}^C (-p_c^i * \log(p_c^i))$. This minimization problem, equation [1](#), is an incomplete-data problem which we solve using the Expectation-Maximization (E-M) technique. Since we follow a similar approach to [7](#), the details of these steps are omitted here.

Although most of the macro-clusters constructed in the previous step are made as pure as possible, some of them may contain instances from a mixture of classes. A completely pure macro-cluster may also contain some unlabeled instances. So, the macro-clusters are split into micro-clusters so that each micro-cluster contains only unlabeled instances or only labeled instances from a single class. At this point, the reader may ask whether we could create the pure micro-clusters in one step using K -means clustering separately for each class and the unlabeled data in a supervised fashion, rather than creating them in two steps, namely, semi-supervised clustering and splitting. The reason for this two-step process is that when limited amount of labels are available, semi-supervision is usually more useful than full supervision. It is likely that supervised K -means would create low quality, less dense or more scattered, clusters than semi-supervised clustering. So, the cluster representatives, or pseudo-points, would have less precision in representing the corresponding data points. As a result, the label-propagation may also perform poorly.

Building micro-clusters is done as follows. Suppose \mathcal{MC}_i is a macro-cluster. In the first case, \mathcal{MC}_i contains only unlabeled instances or only labeled instances from a single class. It is assumed to be a valid micro-cluster and no splitting is necessary. In the second case, \mathcal{MC}_i contains both labeled and unlabeled instances, and/or labeled instances from more than one classes. For each class, we create a micro-cluster with the instances of that class. If \mathcal{MC}_i contains unlabeled instances, then another micro-cluster is created with those unlabeled instances (see figures [1\(a\)](#) and [1\(b\)](#)). So, each micro-cluster contains only unlabeled instances, or labeled instances from a single class. If the total number of macro-clusters is K and the total number of classes is C , then the total number of micro-clusters will be at most $C * K = \hat{K}$, which is also a constant. However, in practice, we find that \hat{K} is almost the same as K . This is because in practice most of the macro-clusters are purely homogeneous and need not be split.

Splitting unlabeled micro-clusters: The unlabeled micro-clusters may be further split into smaller micro-clusters. This is because, if the instances of an unlabeled micro-cluster actually come from different classes, then this micro-cluster will have a negative effect on the label propagation (see the analysis in section [5.3](#)). However, there is no way to accurately know the real labels of the unlabeled instances. So, we use the predicted labels of those instances that were obtained when the instances were classified using the ensemble. Therefore, the unlabeled micro-clusters are split into purer clusters based on the predicted labels of the unlabeled instances (see figure [1](#)).

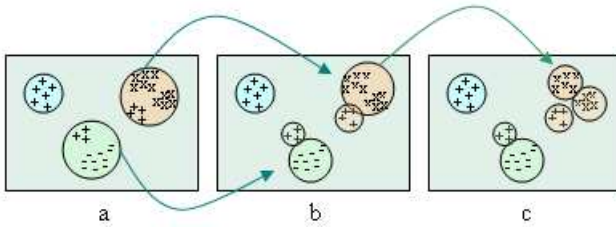


Fig. 1. Illustrating micro-cluster creation. ‘+’ and ‘-’ represent labeled data points and ‘x’ represents unlabeled data points. (a) Macro-clusters created using the constraint-based clustering. (b) Macro-clusters are split into micro-clusters. (c) Unlabeled micro-clusters are further split based on the predicted labels of the data points.

Creating pseudo-points: The centroid of each micro-cluster is computed and a summary of each micro-cluster is saved as a pseudo-point. This summary contains three fields: i) the *centroid*, ii) the *weight*, i.e., the total number of instances in the micro-cluster, and iii) the assigned class label. After saving the pseudo-point, we discard all the raw instances from the main memory. A pseudo-point will be referred to henceforth with the symbol ψ . The centroid of ψ will be denoted with $\mathcal{C}(\psi)$ and the weight of ψ will be denoted with $\mathcal{W}(\psi)$. We consider a pseudo-point as labeled if all the data points in the micro-cluster corresponding to the pseudo-point are labeled.

5 Ensemble Classification

An unlabeled test data may be classified using the transductive label propagation technique by adding the point to an existing model and rerunning the entire label propagation algorithm. Unfortunately, running the transduction for every test point would be very expensive. The efficient alternative is to use the inductive label propagation technique from [12], $\hat{y} = \frac{\sum_j \mathbf{W}_{\psi}(x, \psi_j) \hat{y}_j}{\sum_j \mathbf{W}_{\psi}(x, \psi_j) + \epsilon}$, x is the test point, ψ_j 's are the pseudo-points in the model, \mathbf{W}_{ψ} is the function that generated the matrix \mathbf{W} on $\Psi = \{\psi_1, \dots, \psi_{\hat{K}}\}$, and ϵ is a small smoothing constant to prevent the denominator from being zero. The complexity of this is linear with respect to the number of pseudo-points in a model, \hat{K} .

5.1 Ensemble Refinement

We may occasionally need to refine the exiting models in the ensemble if a new class arrives due to concept-evolution in the stream or old models become outdated due to concept-drift.

Pseudo-point injection: This is required when a new class arrives in the stream. We call a class \hat{c} as a “new” class if no existing model in the ensemble contains any pseudo-point with class label \hat{c} , but M' , the new model built from

the latest training data, contains some pseudo-points with that class label. Refinement is done by injecting pseudo-points of class \hat{c} into the existing models. When a pseudo-point is injected in a model $M_i \in M$, two existing pseudo-points in M_i are merged to ensure that the total number of pseudo-points remains constant. The closest pair of pseudo-points having the same class labels are chosen for merging.

Pseudo-point deletion: In order to improve the classification accuracy of a classifier M_i , we occasionally remove pseudo-points that may have negative effect on the classification accuracy. For each pseudo-point $\psi \in M_i$, we maintain the accuracy of ψ as $A(\psi)$. $A(\psi)$ is the percentage of manually labeled instances for which ψ is a nearest neighbor and whose class label is the same as that of ψ . So, if for any labeled instance x , its nearest pseudo-point ψ has a different class label than x , then $A(\psi)$ drops. This statistic help us to determine whether any pseudo-point has been wrongly labeled by the label propagation or if the pseudo-point has become outdated because of concept-drift. In general, we delete ψ if $A(\psi)$ drops below 70%.

5.2 Ensemble Updating

Our ensemble classifier M consists of L classification models $= \{M_1, \dots, M_L\}$. The training algorithm described in the previous section (section 4) builds one such model M' from the training data. Each of the $L+1$ models, M' and L models in the ensemble, are evaluated using the labeled instances of the training data and the best L of them based on accuracy are chosen for the new ensemble. The worst model is discarded. The ensemble is always kept up to date with the most recently trained model. This is an efficient way to handle concept-drift.

5.3 Error Reduction Analysis for Cluster Splitting and Removal

First, we introduce the notion of *swing voter* for a test instance x . Let the pseudo-point ψ_i be the swing voter for x if the label of ψ_i determines the predicted label of x according to the inductive equation. Note that such a voter must exist for any test instance x . Also, $W_{\psi}(x, \psi_i)$ is the weight from inductive label propagation. Usually, the ψ_i that have the highest $W_{\psi}(x, \psi_i)$ should be the swing voter for x . In other words, the nearest pseudo-point to x is most likely be its swing voter. Let the probability that ψ_i is a swing voter for a test point x be α_i . Also, let the probability that the class label of the swing voter ψ_i is different from the actual class of x be p_i . For example, if there are 100 test points and ψ_i is the swing voter for 20 test points, then $\alpha_i=20/100 = 0.2$. We denote an operation called $CL(x)$ to return the class label of a point or pseudopoint. Also, among the 20 test points, if 10 have different class label than ψ_i , then $p_i = 10/20 = 0.5$. It would be clear shortly that α_i is related to deletion and p_i is related to splitting. Therefore, the probability that the next test instance x will not be misclassified because of this pseudo-point $= P(\psi_i$ will not be a swing voter for $x) + P(\psi_i$ will be a swing voter for x and $CL(x) = CL(\psi_i)) = (1 - \alpha_i) + \alpha_i(1 - p_i) = 1 - \alpha_i + \alpha_i - \alpha_i p_i = 1 - \alpha_i p_i$.

\Rightarrow Probability that none of the next N test instances will be misclassified because of this pseudo-point, assuming independence among the test instances $= (1 - \alpha_i p_i)^N$.

\Rightarrow Probability that one or more of the next N test instances will be misclassified because of this pseudo-point (i.e., probability of error), $P(\mathcal{E}_i) = 1 - (1 - \alpha_i p_i)^N$. The expected error of a classifier m is the weighted average of the error probabilities of its pseudo-points, i.e.,

$$\begin{aligned} E(\mathcal{E}_m) &= \frac{\sum_i \alpha_i P(\mathcal{E}_i)}{\sum_i \alpha_i} = \frac{\sum_i \alpha_i (1 - (1 - \alpha_i p_i)^N)}{\sum_i \alpha_i} = \frac{\sum_i \alpha_i}{\sum_i \alpha_i} - \frac{\sum_i \alpha_i (1 - \alpha_i p_i)^N}{\sum_i \alpha_i} \\ &= 1 - \sum_i \alpha_i (1 - \alpha_i p_i)^N \quad (\text{since } \sum_i \alpha_i = 1) \end{aligned} \quad (2)$$

According to equation 2, the expected error can be minimized if the second term $\sum_i \alpha_i (1 - \alpha_i p_i)^N$ can be maximized. There are two ways to maximize this quantity: making $p_i=0$ by splitting unlabeled micro-clusters or making $\alpha_i = 0$ by removing pseudo-points from the model.

Splitting unlabeled micro-clusters: If it is assumed that the test instances are identically distributed as the training instances, then we can apply a heuristic to reduce p_i . Recall that p_i is the probability that a test point (for which ψ_i is a swing voter) will have a different class label than the label of the pseudo-point ψ_i . Intuitively, p_i would be zero if all the training instance in the corresponding micro-cluster has the same class label as ψ_i . Although this is ensured for the micro-clusters that have labeled instances, it cannot be ensured for the micro-clusters that have unlabeled instances. Therefore, we use the classifier-predicted labels of the unlabeled instances to determine whether an unlabeled micro-cluster is pure or not. If it is not pure based on the predicted labels, then we split the micro-cluster into purer micro-clusters. Thus, splitting the unlabeled micro-clusters help to keep p_i to a minimum, and reduce the expected error of the corresponding classifier.

Deleting pseudo-points: If for some ψ_i , p_i is too high, then the pseudo-point has a negative effect on the overall classifier accuracy. In this case, we can remove the pseudo-point to improve accuracy, because removal of ψ_i would make $\alpha_i=0$. Intuitively, $p_i = 1 - A(\psi_i)$, where $A(\psi_i)$ is the accuracy of the pseudo-point ψ_i . However, removal helps only if there is no *cascading effect* of the removal on other pseudo-points. A cascading effect occurs if the removal of ψ_i increases p_j of another pseudo-point ψ_j . This is possible if ψ_j becomes the new swing voter for a test instance x , whose original swing voter had been ψ_i , and the class label of x is the same as that of ψ_i , but different from that of ψ_j . To account for this we implemented a simple threshold (i.e. max 50%) deleted.

6 Experiments

Synthetic datasets, **SYN-E** and **SYN-D**, are standard methods for evaluating stream mining methods. These are described in detail in [1]. SYN-E simulates

concept evolution by adding new classes into the stream as time progresses. SYN-D simulates concept drift by changing the slope of a hyperplane over time. The KDDCUP 99 intrusion detection dataset, **KDD**, is also very widely used in stream mining literature, see [1]. It contains 23 different classes, 22 of which are labeled network attacks. The NASA Aviation Safety Reporting System database, **ASRS**, is our second real dataset. The dataset contains around 150,000 text reports, each describing some kind of flight anomaly. See [13] for more details.

6.1 Experimental Setup

Hardware and software: We implement the algorithms in Java. We use a windows-XP based Intel P-IV machine 3GHz processor and 2GB main memory.

Parameter settings: We will refer to our technique as LabelStream. parameter settings of LabelStream are as follows, unless mentioned otherwise: K (number of macro-clusters) = 50; Chunk-size = 1,600 records for real datasets, and 1,000 records for synthetic datasets; L (ensemble size) = 6;

Baseline method: We compare our algorithm with that of Masud et al [7] and Aggarwal et al [1]. We will refer to these approaches as SmSCluster and OnDemandStream, respectively. We run our own implementation of both these baseline techniques. For the SmSCluster, we use the following parameter settings: K (number of micro-clusters) = same as K of LabelStream; Chunk-size = same as the chunk-size of LabelStream; L (ensemble size) = same as the ensemble-size of LabelStream; ρ (injection propability) = 0.75, as suggested by [7]; Q (nearest neighbors in K-NN classification) = 1, as suggested by [7]. For OnDemandStream, we use the following parameter settings: Buffer-size = same as the chunk-size of LabelStream; Stream speed = 80 for real dataset and 200 for synthetic dataset (as suggested by the authors). Other parameters of OnDemandStream are set to the default values.

In the following subsections, we would use the terms “P% labeled” to mean that P% of the instances in a chunk are labeled. So, when we mention that LabelStream is run with 10% labeled data and OnDemandStream is run with 100% labeled data, it means for the same chunk-size (e.g. 1000), LabelStream is trained with a chunk having 100 labeled and 900 unlabeled instances, whereas OnDemandStream is trained with the same chunk having 1000 (all) labeled instances.

6.2 Performance Study

To illustrate the effectiveness of the proposed approach, table 1 shows overall accuracy values for previous approaches SmSCluster and OnDemandStream against LabelStream against all four datasets. There are two methods to decide labeled training instances: *bias* and *random*. Under bias sampling, a point from a class is drawn at random and a labeled set is initialized with that point. Then the nearest neighbor to the labeled set belonging to the same class is added to the labeled set.

Table 1. Performance comparison of SmSCluster and LabelStream at 10% labeled data and OnDemandStream at 100% labeled data

	LabelStream		SmSCluster	OnDemandStream
	Bias	Random		
SYN-E	99.76	98.35	90.28	69.78
SYN-D	84.48	86.40	75.15	73.25
KDD	97.69	98.06	92.57	96.07
ASRS	48.30	41.07	30.33	28.02

Table 2. Testing, training, and manual labeling speeds for the four datasets

	LabelStream		SmSCluster		Manual	OnDemandStream		Manual
	Train	Test	Train	Test	10%	Train	Test	100%
SYN-E	1.1	1.33	1.49	3.0	-	1.15	10.22	-
SYN-D	0.47	0.49	0.66	0.59	-	0.27	7.49	-
KDD	13.2	0.93	7.30	7.39	9600	1.23	20.03	96000
ASRS	60.3	24.5	16.17	41.9	9493.7	613.6	446.8	94936.7

This continues until P% of the points have been drawn for that class. This is done for each class. In random sampling, points are randomly drawn at uniform to be marked as labeled datapoints. The experiment is repeated 20 times and the accuracy value is averaged. SmSCluster and OnDemandStream are run with 10% and 100% labeled data, respectively. LabelStream values are at both 10% randomly drawn data and a special dataset containing biased labeled data. LabelStream performs better than SmSCluster and OnDemandStream in general. For example, table 1 shows under biased and random sampling LabelStream has a 48.9% and 41.07% accuracy, respectively, on the ASRS dataset while SmSCluster has a 30.33% accuracy and OnDemandStream has a 28% accuracy.

LabelStream seems to outperform SmSCluster and OnDemandStream in terms of classification accuracy. Now, we will investigate the difference in processing speed among these algorithms. Table 2 shows a comparison of running times of the three methods across all four datasets. LabelStream and SmSCluster were run with 10% labeled data and OnDemandStream was run with 100% labeled data as in the previous graphs in this section. Results are given in two columns, training and testing times, for each algorithm with the addition of times for the amount of manual annotation needed for each dataset, 60 seconds per instance. These times are just used to illustrate the true gain of LabelStream and SmSCluster over previous approaches as true labeling time is likely much higher. Also, synthetic datasets do not get annotation times because they were machine generated. Times listed are processing times, in seconds, for each data chunk. For example, a chunk containing 1600 data points may take 60.3 seconds to train on LabelStream, 16.17 seconds on SmSCluster, and 613.6 seconds on OnDemandStream. Testing takes 24.5 seconds for LabelStream, 41.9 seconds for SmSCluster, and 446.8 seconds for OnDemandStream. The machine training time is always insignificant compared to the manual annotation time.

References

1. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for on-demand classification of evolving data streams. *IEEE Transactions on Knowledge and Data Engineering* 18(5), 577–589 (2006)
2. Chen, S., Wang, H., Zhou, S., Yu, P.: Stop chasing trends: Discovering high order models in evolving data. In: *Proc. ICDE*, pp. 923–932 (2008)
3. Fan, W.: Systematic data selection to mine concept-drifting data streams. In: *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, Seattle, WA, USA, pp. 128–137 (2004)
4. Scholz, M., Klinkenberg, R.: An ensemble classifier for drifting concepts. In: *Proc. Second International Workshop on Knowledge Discovery in Data Streams (IWKDD)*, Porto, Portugal, October 2005, pp. 53–64 (2005)
5. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: *Proc. ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington, DC, USA, pp. 226–235. ACM, New York (2003)
6. Yang, Y., Wu, X., Zhu, X.: Combining proactive and reactive predictions for data streams. In: *Proc. KDD*, pp. 710–715 (2005)
7. Masud, M., Gao, J., Khan, L., Han, J., Thuraisingham, B.: A practical approach to classify evolving data streams: Training with limited amount of labeled data. In: *Proc. International Conference on Data Mining (ICDM)*, Pisa, Italy, December 15–19, pp. 929–934 (2008)
8. Domingos, P., Hulten, G.: Mining high-speed data streams. In: *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, Boston, MA, USA, pp. 71–80. ACM Press, New York (2000)
9. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: *Proc. seventh ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, San Francisco, CA, USA, August 2001, pp. 97–106 (2001)
10. Gao, J., Fan, W., Han, J.: On appropriate assumptions to mine data streams. In: *Proc. Seventh IEEE International Conference on Data Mining (ICDM)*, Omaha, NE, USA, October 2007, pp. 143–152 (2007)
11. Kolter, J., Maloof, M.: Using additive expert ensembles to cope with concept drift. In: *Proc. International Conference on Machine Learning (ICML)*, Bonn, Germany, August 2005, pp. 449–456 (2005)
12. Bengio, Y., Delalleau, O., Le Roux, N.: Label propagation and quadratic criterion. In: *Chapelle, O., Schölkopf, B., Zien, A. (eds.) Semi-Supervised Learning*, pp. 193–216. MIT Press, Cambridge (2006)
13. Woolam, C., Khan, L.: Multi-label large margin hierarchical perceptron. *IJD-MMM* 1(1), 5–22 (2008)

Novelty Detection from Evolving Complex Data Streams with Time Windows

Michelangelo Ceci, Annalisa Appice, Corrado Loglisci, Costantina Caruso,
Fabio Fumarola, and Donato Malerba

Dipartimento di Informatica, Università degli Studi di Bari
via Orabona, 4 – 70126 Bari, Italy

{ceci,appice,loglisci,caruso,ffumarola,malerba}@di.uniba.it

Abstract. Novelty detection in data stream mining denotes the identification of new or unknown situations in a stream of data elements flowing continuously in at rapid rate. This work is a first attempt of investigating the anomaly detection task in the (multi-)relational data mining. By defining a data block as the collection of complex data which periodically flow in the stream, a relational pattern base is incrementally maintained each time a new data block flows in. For each pattern, the time consecutive support values collected over the data blocks of a time window are clustered, clusters are then used to identify the novelty patterns which describe a change in the evolving pattern base. An application to the problem of detecting novelties in an Internet packet stream is discussed.

1 Introduction

A data stream is an ordered sequence of data elements generated at rapid rate. Differently from data in traditional static databases, data streams are continuous, unbounded, usually come with high speed and have a data distribution which may change with time because of fundamental changes in the underlying phenomena [8]. These characteristics of data stream pose specific computational issues which prevent the application of traditional data mining algorithms, which are designed to extract knowledge from static data only. First, the continuous, unbounded, and high speed characteristics of data streams require abilities to collect and process a huge amount of data. Anyway, there is neither enough time to rescan the whole stream each time an update occurs nor enough space to store the entire data stream for online processing. Second, the temporal evolution which typically characterizes the distribution of data in a stream demands for techniques that can capture the evolution of extracted patterns as well.

Several efficient and effective algorithms have been proposed in the literature to extract knowledge from data streams, mainly for clustering, association rules discovery, time series analysis and novelty detection [3,10,5,6]. Most of these algorithms perform an incremental learning [14] which makes them able to face issues posed by continuous, unbounded, evolving characteristics of data streams. However, a common limitation is that they are designed to mine data elements which arrive as vectors of fixed attribute values. In many applications, the stream

is actually a sequence of complex data elements, composed of several objects of various data types which are somehow related. For instance, network traffic in a LAN can be seen as a stream of connections, which are described by some properties (e.g., protocol) as well as by the sequence of one or more packets which flows consecutively in the network as part of the same connection. The structure of these complex data elements is naturally modeled by several relations of a relational databases, hence making methods of (multi-)relational data mining [4] more suitable for the discovery of useful patterns from these data streams. Currently, query processing engines [2,9] have been realized in order to query stream stored into relational database systems and deliver result sets on-the-fly. These systems integrate next generation query languages which extend SQL in order to extract data from the stream stored in the database, but they do not integrate multi-relational data mining systems to discover novel and unknown patterns from these data.

In this work we focus on the novelty (or anomaly) detection task in complex data stream mining. Anomaly detection targets learning algorithms [7,12,17] being able to identify unknown situations which represent a change with respect to situations experienced before. This work addresses the anomaly detection problem by firstly resorting to a multi-relational data mining algorithm, called Mr-NoDeS (*M*ulti-*R*elational *N*ovelty *D*etection in *D*ata *S*tream), which is based on an incremental approach to mine a relational pattern base from the complex data lastly flowed in a stream and provides a human interpretable description of the changes which occur in this evolving pattern base. New iterations of mining results are built based on old mining results. The algorithm is based on the definition of time sensitive data block [7], that is, the set of complex data which are periodically (e.g., daily, monthly) added to a stream. The pattern base includes relational patterns (i.e., patterns possibly involving several database relations) which are frequent on at least one data block of a time window ending at current time. The time window is defined as a user-defined number of the lastly income blocks. Novelty patterns are those patterns in the base whose frequency on the last block significantly changes with respect to an “homogeneous” region of frequencies computed in the remaining window blocks.

The paper is organized as follows. Section 2 presents preliminary concepts and defines novelty patterns. The algorithm Mr-NoDeS is described in Section 3. Section 4 reports an application of the proposed algorithm on an Internet packet stream. Lastly, some conclusions are drawn.

2 Preliminary Concepts and Definitions

In the traditional streaming model, the input data elements $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n, \dots$ arrive sequentially, item by item, as continuous fixed-length vectors of attribute values \mathbf{a}_i . This attribute-value representation appears unnatural and inadequate in several real world data stream applications where data elements are complex data which consist of several objects possibly belonging to heterogeneous data types. These objects may play different roles, hence it is necessary to distinguish

between the set S of reference (or target) objects, which are the main subject of analysis, and the sets $R_k, k = 1, \dots, M$, of task-relevant (or non-target) objects, which are related to the former and can contribute to account for the variation. In the relational data model, both reference objects and task-relevant object can be naturally modeled as distinct relations (or tables) of a relational database D . Let H be the schema of D , H includes the definition of a (target) relation T_S which stores the properties (or attributes) of objects in S as well as an arbitrary number of additional (non-target) relations T_{R_k} , where each T_{R_k} stores attributes of objects in R_k . A reference object $s \in S$ defines a unit of analysis $D[s]$. Task-relevant objects which are somehow related to the reference object contribute to defining the unit of analysis without being the main subject of the analysis. The “structure” of units of analysis, that is, the relationships between reference and task-relevant objects, is expressed in the schema H by foreign key constraints (FK). Foreign keys make it possible to navigate the data schema and retrieve all the task-relevant objects in D which are related to a reference object. In this way, a unit of analysis is retrieved by simply navigating the database structure without requiring any additional pretreatment of the complex data arriving in the stream.

Definition 1 (Unit of analysis). *A unit of analysis $D[s]$ consists of a reference object $s \in T_S$ and the finite set of task-relevant objects stored in some T_{R_k} which are related to s according to foreign key constraints of H .*

This notion of unit of analysis is coherent with the individual-centered representation [1], which has both theoretical (PAC-learnability) and computational advantages (smaller hypothesis space and more efficient search). In a streaming model, units of analysis are associated with time points.

Definition 2 (Complex data stream). *Let $t_1, t_2, \dots, t_i, \dots$ a sequence of time points and let \prec the order relationship defined among them. A complex data stream is a continuous flow DS of units of analysis associated with the time points, i.e., $DS = \{\langle D[s_1], t_1 \rangle, \langle D[s_2], t_2 \rangle, \langle D[s_n], t_n \rangle, \dots\}$, where $t_i \prec t_{i+1}$.*

Time intervals define data blocks in a complex data stream [7].

Definition 3 (Data block). *Given a time point t and a time period p , a basic data block B is the set of units of analysis $D[s_i]$ such that their associated time t_i is in $[t - p + 1, t]$, i.e. $t_i \in [t - p + 1, t]$.*

The number of units of analysis in a basic data block B_i is denoted as $|B_i|$.

Given a time period p , a complex stream can be partitioned into consecutive data blocks B_1, B_2, \dots such that B_i includes the units of analysis observed in the time interval $[t_0 + (i - 1) \cdot p, t_0 + i \cdot p]$. Since we are interested to capture evolutions (and novelties) from block to block, we extend the notion of time window to data blocks.

Definition 4 (Time window). *Let w be a window size. Then the time window $W(i, w)$ associated with each B_i is the set of basic blocks B_{i-w+1}, \dots, B_i .*

Mining only data in a time window is a natural choice in data stream mining. Indeed, data distribution in a stream may change in time and we are interested in discovering a model which is descriptive of the recently income data only. In this work, the discovered model is intended as a base of “relational” patterns. In order to formally define a relational pattern, we introduce the concepts of key predicate, structural predicate and property predicate.

Definition 5 (Key predicate). *The key predicate associated with the target table T_S in H is a unary predicate $p(t)$ such that p denotes the table T_S and the term t is a variable that represents the primary key of T_S .*

Definition 6 (Property predicate). *A property predicate is a binary predicate $p(t, c)$ associated with the attribute Att of the table T_i . The name p denotes the attribute Att , the term t is a variable representing the primary key of T_i and c is a constant which represents a value belonging to the range of Att in T_i .*

Definition 7 (Structural predicate). *A structural predicate is a binary predicate $p(t, s)$ associated with the foreign key constraint FK from the table T_i to the table T_j in H . The name p denotes FK , while the terms t and s are two variables which represent foreign key in T_i and the primary key in T_j according to FK .¹*

A relational pattern is defined as follows:

Definition 8 (Relational pattern). *A relational pattern P over the schema H is a conjunction of predicates $p_0(t_0^1), p_1(t_1^1, t_1^2), p_2(t_2^1, t_2^2), \dots, p_m(t_m^1, t_m^2)$, where $p_0(t_0^1)$ is the key predicate associated with the table T_S and $p_i(t_i^1, t_i^2), i = 1, \dots, m$, is either a structural predicate or a property predicate over H .*

An example of relational pattern is reported in Example [II](#)

Example 1. Let us consider a stream of connections incoming a firewall, where each connection is a sequence of consecutive packets. An example of relational pattern P is the following:

connection(C), protocol(C, tcp), contain_packet(C, D), contain_packet(C, E),
D ≠ E, next(D, E)

where *connection* is a key predicate, *protocol* and *time* are property predicates, *contain_packet* is a structural predicate.

The support of a relational pattern P on a block B_i is computed as follows:

$$s_i(P) = \frac{|\{D[s] | \langle D[s], t \rangle \in B_i, \exists \theta : P\theta \subseteq D[s]\}|}{|\{D[s] | \langle D[s], t \rangle \in B_i\}|}, \tag{1}$$

where θ is a substitution of variables into constants and $P\theta$ denotes the application of the substitution θ to the pattern P . Therefore, we define a relational

¹ In database theory, a foreign key constraint allows certain attributes (foreign key) in one table to refer to attributes (primary key) in another table. In our formalization, both the primary key and the foreign key of a constraint are mapped into two variables which univocally identify the pairs of tuples related by the constraint.

pattern P as *frequent* with respect to a minimum support threshold $minSupp$ if a block B_i exists, such that $s_i(P) \geq minSupp$.

A novelty pattern on a time window can be formally defined as follows.

Definition 9 (Novelty pattern). *Let (1) $W(i, w) = \langle B_{i-w+1} B_{i-w+2} \dots, B_i \rangle$ be a time window with length w and an ending block B_i ; (2) P be a pattern and $\langle s_{i-w+1}, s_{i-w+2}, \dots, s_i \rangle$ the list of support values of P on each data block of $W(i, w)$; (3) $\Theta_P : [0, 1] \rightarrow \Psi$ be a discretization function which associates a support value of P in the interval $[0, 1]$ with a discrete values $\psi \in \Psi$. Then, P is a novelty pattern for the time window $W(i, w)$ if and only if $\Theta(s_{i-w+1}(P)) = \dots = \Theta(s_{i-1}(P)) \neq \Theta(s_i(P))$.*

3 The Algorithm

The algorithm Mr-NoDeS is a two step data stream algorithm that is triggered each time a p sized data block arrives in the stream. The algorithm is designed to record only data blocks falling in a w sized time window. In the first step, the relational pattern base $M(i, w)$ is updated each time a data block B_i arrives, while in the second phase patterns in $M(i, w)$ are filtered out in order to keep only those patterns which represent novelty patterns within the time window $W(i, w)$. Details on relational pattern discovery, pattern base maintenance and novelty pattern detections are reported in the next subsections.

3.1 Relational Pattern Discovery

The pattern discovery is performed by exploring level-by-level the lattice of relational patterns ordered according to a generality relation (\geq) between patterns. Given two patterns P_1 and P_2 , $P_1 \geq P_2$ denotes that P_1 (P_2) is more general (specific) than P_2 (P_1). The search proceeds from the most general pattern and iteratively alternates the candidate generation and candidate evaluation phases as in the levelwise method [13]. Candidate generation assumes that the space of pattern is structured according to the θ -subsumption generality order [15].

Definition 10 (θ -subsumption generality order). *Let P_1 and P_2 be two relational patterns. P_1 is more general than P_2 under θ -subsumption, denoted as $P_1 \geq_\theta P_2$, if and only if a substitution θ exists such that $P_2\theta \subseteq P_1$.*

Example 2. Let us consider the patterns: (P_1) connection(C). (P_2) connection(C), packet(C,P). (P_3) connection(C), service(C,'http'). These patterns are ordered as follows: $P_1 \geq_\theta P_2$, $P_1 \geq_\theta P_3$.

The θ -subsumption generality order satisfies the monotonicity property with respect to support (i.e., a specialization of an infrequent pattern cannot be frequent) and defines a quasi-ordering which can be searched according to a downward refinement operator which computes the refinements for a relational pattern [11]. The downward refinement operator ρ' used in this work is defined below.

Definition 11 (Downward refinement operator). Let P be a relational pattern. Then $\rho'(P) = \{P \cup \{p(t_1, t_2)\} | p(t_1, t_2) \text{ is a structural or property predicate that shares at least one term with one predicate already occurring in } P\}$.

ρ' is a refinement operator under θ -subsumption, i.e., $P \geq_{\theta} Q$ for all $Q \in \rho'(P)$. Due to the monotonicity property of θ -subsumption generality order with respect to support, ρ' allows for a level-wise exploration of the quasi-ordered set of relational patterns. Indeed, the implemented algorithm starts from a set \wp containing only the most general pattern, i.e. the pattern that contains only the key predicate, and then updates \wp by repeatedly applying ρ' to all patterns in \wp . In generating each level of the quasi-ordered set, the candidate pattern search space is represented as a set of enumeration trees (SE-trees) [18]. The idea is to impose an ordering on atoms such that all patterns in the search space are enumerated. Practically, a node g of a SE-tree is represented as a group comprising: the head ($h(g)$), i.e. the pattern enumerated at g , and the tail ($t(g)$) that is the ordered set consisting of all atoms which can be potentially appended to g by ρ' in order to form a pattern enumerated by some sub-node of g . A child g_c of g is formed by taking an atom $q \in t(g)$ and appending it to $h(g)$. Therefore, $t(g_c)$ contains all atoms in $t(g)$ that follows q (see Figure 1). In the case q is a structural predicate, $t(g_c)$ contains both atoms in $t(g)$ that follows q and new atoms directly linkable to q according to ρ' not yet included in $t(g)$. Given this child expansion policy, without any pruning of nodes or pattern, the SE-tree enumerates all possible patterns and prevents the generation and evaluation of candidate equivalent under θ -subsumption to some other candidate.

As pruning criterion, the monotonicity property of the generality order \geq_{θ} with respect to the support value is exploited to avoid refinement of infrequent relational patterns. An additional pruning criterion stops the search when a maximum number of literals (*MaxNumLiterals*) have been added to a pattern.

3.2 Pattern Base Maintenance

The pattern base $M(i, w)$ is the set of relational patterns which are frequent on at least one basic block of the time window $W(i, w)$. A pattern $P \in M(i, w)$ is associated with a support list $\langle s_{i-w+1}(P), s_{i-w+2}(P) \dots, s_i(P) \rangle$, that is, the list of support values computed for P on each data block of $W(i, w)$. A pattern base $M(i, w)$ is mined starting from $M(i-1, w)$ when the data block B_i arrives in the stream. Operations to maintain the pattern base are inserting frequent patterns,

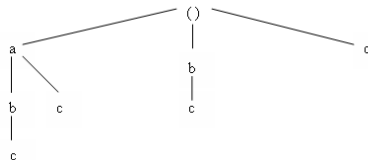


Fig. 1. The enumeration tree to search the patterns a, b, c, ab, ac, bc, abc

deleting infrequent patterns and updating the support list of a pattern already belonging to the base. We can distinguish between three cases based upon the serial number i of the data block B_i which arrives in the stream.

($i = 1$). The pattern base $M(1, w)$ is mined from scratch: $M(1, w) = \{P_i | s_1(P_i) \geq \text{minSupp}\}$.

($i = 2, \dots, w$). Relational patterns which are frequent on B_i are discovered and used to construct M_i^+ , that is the set of patterns P which are frequent on B_i , but do not belong to $M(i - 1, w)$. $M(i, w)$ is constructed by adding patterns of M_i^+ to $M(i - 1, w)$. For each pattern $P \in M(i - 1, w)$, the associated support list $\langle s_1(P), \dots, s_{i-1}(P) \rangle$ is updated by adding $s_i(P)$. Differently, for each pattern $P \in M_i^+$, the entire support list $\langle s_1(P), \dots, s_i(P) \rangle$ is built from scratch.

($i > w$). The sliding time window moves from B_{i-w}, \dots, B_{i-1} to B_{i-w+1}, \dots, B_i and the data block B_{i-w} is removed from memory. Relational patterns which are frequent on B_i are discovered and used to construct both M_i^+ and M_i^- .

1. M_i^+ that is the set of patterns P which are frequent on B_i but do not belong to $M(i - 1, w)$;
2. M_i^- that is the set of patterns P which are infrequent on B_i , belong to $M(i - 1, w)$ but are infrequent on $B_{i-w+1}, \dots, B_{i-1}$.

$M(i, w)$ is then constructed from $M(i - 1, w)$ by adding patterns of M_i^+ and removing patterns of M_i^- . The support list of each pattern $P \in M(i, w)$ is updated or created from scratch. An update operation is performed when the pattern P is already included in $M(i - 1, w)$ ($P \in M(i, w) - M_i^+$). In this case, the support list associated to P is updated by removing $s_{i-w}(P)$ and by adding $s_i(P)$. A creation operation is performed when $P \in M_i^+$ and the entire support list $\langle s_{i-w+1}(P), \dots, s_i(P) \rangle$ is built.

3.3 Time-Window Based Novelty Pattern Detection

Mr-NoDeS post-processes the pattern base $M(i, w)$ in order to identify novelty patterns for B_i . The function Θ_P is a discretization function based on a density based clustering algorithm. Several clustering algorithms have been designed in the literature. In this paper we use the density-base clustering algorithm DBSCAN [16] which is devised to discover arbitrary-shaped clusters which are discovered without providing a-priori the number of clusters. The complexity is quadratic with respect to the size w of the window ($\mathbf{O}(w^2)$). Clusters are intended as dense, timely consecutive, areas. For each pattern P , the cluster construction starts from a data block b (seed) and constructs the neighborhood $N(b)$ that includes b and the data block incoming before b as well as the data block incoming after b in the stream. The neighborhood is labeled as a cluster c only if it satisfies the condition of dense region. Density is estimated by means of the standard deviation of the support values of P falling in $N(b)$. Standard deviation must not exceed a user-defined threshold (maxStd). The cluster is then expanded by merging partially overlapping neighborhoods. The merge is performed only

in the case that the cluster which is output of the merge operation satisfies the condition of dense region. To avoid to evaluate the possible expansion of a cluster by merging an heterogeneous neighborhood, Mr-NoDeS evaluates only the merge of neighborhoods including support values whose standard deviation does not exceed a local user-defined threshold (*localMaxSTD*). If a cluster cannot be further expanded, a new seed is selected and a new cluster is constructed. The strategy adopted to select the seed is the sequential one. The cluster c is labeled with the interval $[minC, maxC]$ where $minC$ ($maxC$) is the minimum (maximum) support value falling in c .

Each time, the clustering algorithm segments the time window $W(i, w)$ of P in only two clusters, namely $c1$ and $c2$, such that $c1$ includes the support values for data blocks $B_{i-w+1}, \dots, B_{i+1}$ and $c2$ includes the support value for the data block B_i , then P is marked as a novelty pattern for B_i over $W(i, w)$. In other words, patterns whose support value passes from a cluster to another, are marked as novelty patterns.

4 The Application

Mr-NoDeS is applied to detect anomaly patterns in the Internet packet stream incoming the firewall of the Department of Informatics in Bari from June 1st to June 28th, 2004. Units of analysis are time-stamped ingoing connections (reference objects) which are composed by several packets (task-relevant objects).

Dataset Description. A connection is described by the identifier (integer); the protocol (nominal); the starting time (integer); the IP destination (nominal); the service (nominal); the number of packets (integer); the average packet time distance (integer); the time length (integer); the source nation code (nominal); the source nation time zone (integer). Each packet is described by the identifier (integer) and the starting time (number) of the packet within the connection. The interaction between consecutive packets is described by the time distance. Numeric attributes are discretized through an equal-width discretization that partitions each range of values into 10 bins.

Analysis of results. In these experiments, the data block size is 24 hours, while starting point is at 00:00 on June 1st, 2004. Novelty pattern discovery is triggered each time a new data block arrives in the stream by setting $minSupp = 0.1$, $MaxNumLiterals = 5$, $maxSTD = 0.02$, $localMaxSTD = 0.05$ and $w = 3, 4, 5, 6$. The number of anomaly patterns is plotted in Figure 2. Interestingly, the number of patterns extracted for each time window is rather large. This is due to the high number of similar extracted patterns. In fact, in most of cases, Mr-NoDeS extracts the patterns that are related each other according to the θ -subsumption generality order (one is the specialization of the other). However, the number of discovered novelty patterns significantly decreases for $w = 6$, where the average number of patterns extracted for each data block is 59.48. This makes it possible to manually analyze patterns. In addition, by observing the smoothing of peaks in the number of novelty patterns per

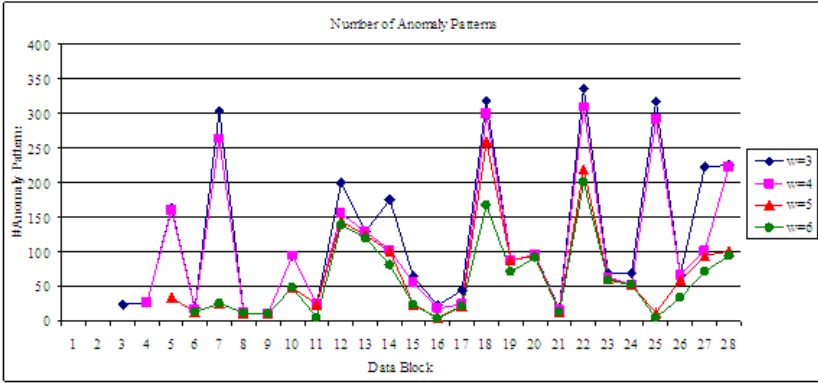


Fig. 2. Number of anomaly patterns discovered on each data block by varying w

data blocks we observe that the cardinality of anomaly pattern base presents a high variance over the different data blocks when $w = 3$, while this variance is somehow mitigated by increasing values of w . This would help the user to identify and analyze critical days, when attacks may have occurred. There are several critical data blocks (days) when $w = 3$ and less when $w = 6$. In particular, days where the number of extracted novelty patterns is greater than 200 decreases from $B_7, B_{11}, B_{18}, B_{22}, B_{25}, B_{27}, B_{28}$ when $w = 3$ to B_{22} when $w = 6$. An example of novelty pattern P_1 detected for the data block B_{22} (June 22th) by using $w = 5$ is “ $conn(C), packet(C, P), proto(C, udp)$ ”, where $s_{18}(P_1)=0.0420, s_{19}(P_1)=0.078, s_{20}(P_1)=0.095$ and $s_{21}(P_1)=0.0422$ are automatically clustered in a single region labeled with $[0.0420, 0.095]$, while $s_{22}(P)=0.71$ is clustered alone. This pattern describes as an anomaly on June 22th, 2004 the sharp increase of percentage of udp connections incoming the firewall. An example of novelty pattern P_2 which takes into account the relational nature of data is extracted on June 20th, 2004 with $w = 5$, that is, “ $conn(C), packet(C, P), packToPack(P, Q), dist(P, Q, [0, 0.280]), service(C, unknown)$ ”. P_2 has a support value of 0.213 on the June 20th 2004 (B_{20}), while its support is clustered in the region $[0.0, 0.0]$ in the previous days of the window $W[20, 5]$. This pattern detects as anomalous the high number of connections C which use an *unknown* service and include at least two packets P and Q , where P is sent after Q with a time distance that is in the interval $[0, 280]$ ms.

5 Conclusions

In this paper, we present a multi-relational data mining algorithm to discover novelty patterns from evolving data blocks of complex data streams. Complex data are stored in several relations of a relational database. A relational pattern base is maintained for the stream data falling in the current a time window. A clustering algorithm is employed to detect sharp change of support values. The algorithm is applied for the anomaly detection in an Internet Packet stream.

Acknowledgment

This work is partial fulfillment of objective of ATENEO-2008 project “Scoperta di conoscenza in domini relazionali” and Strategic Project PS121: “Telecommunication Facilities and Wireless Sensor Networks in Emergency Management”.

References

1. Blockeel, H., Sebag, M.: Scalability and efficiency in multi-relational data mining. SIGKDD Explorations Newsletter 5(1), 17–30 (2003)
2. Brenna, L., Demers, A., Gehrke, J., Hong, M., Ossher, J., Panda, B., Riedewald, M., Thatte, M., White, W.: Cayuga: a high-performance event processing engine. In: International Conference on Management of Data, pp. 1100–1102. ACM, New York (2007)
3. Domingos, P., Hulten, G.: Mining high-speed data streams. In: the 6th International Conference on Knowledge Discovery and Data Mining, KDD 2000, pp. 71–80. ACM, New York (2000)
4. Džeroski, S., Lavrač, N.: Relational Data Mining. Springer, Heidelberg (2001)
5. Gaber, M.M., Zaslavsky, A., Krishnaswamy, S.: Mining data streams: a review. SIGMOD Record 34(2), 18–26 (2005)
6. Gama, J.: Issues and challenges in learning from data streams. In: Kargupta, H., Han, J., Yu, P.S., Motwani, R., Kumar, V. (eds.) Data Mining and Knowledge Discovery Series on Next Generation of Data Mining, pp. 209–222. Chapman and Hall, CRC Press, Taylor and Francis Group (2009)
7. Ganti, V., Gehrke, J., Ramakrishnan, R.: Mining data streams under block evolution. SIGKDD Explorations 3(2), 1–10 (2002)
8. Guha, S., Koudas, N., Shim, K.: Data-streams and histograms. In: the 33th Symposium on Theory of Computing, STOC 2001, pp. 471–475. ACM, New York (2001)
9. <http://www.streambase.com/>
10. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: the 7th International Conference on Knowledge Discovery and Data Mining, KDD 2001, pp. 97–106. ACM, New York (2001)
11. Lisi, F.A., Malerba, D.: Inducing multi-level association rules from multiple relations. Machine Learning 55(2), 175–210 (2004)
12. Ma, J., Perkins, S.: Online novelty detection on temporal sequences. In: the 9th International Conference on Knowledge Discovery and Data Mining, KDD 2003, pp. 613–618. ACM, New York (2003)
13. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery 1(3), 241–258 (1997)
14. Mitchell, T.: Machine Learning. McGraw Hill, New York (1997)
15. Plotkin, G.D.: A note on inductive generalization. Machine Intelligence 5, 153–163 (1970)
16. Sander, J., Ester, M., Kriegel, H.-P., Xu, X.: Density-based clustering in spatial databases: The algorithm gdbscan and its applications. Data Mining and Knowledge Discovery 2(2), 169–194 (1998)
17. Spinosa, E.J., de Carvalho, A.P.d.L.F., Gama, J.: Cluster-based novel concept detection in data streams applied to intrusion detection in computer networks. In: The Symposium on Applied Computing, SAC 2008, pp. 976–980. ACM, New York (2008)
18. Zhang, X., Dong, G., Kotagiri, R.: Exploring constraints to efficiently mine emerging patterns from large high-dimensional datasets. In: Knowledge Discovery and Data Mining, pp. 310–314 (2000)

On Computational Creativity, ‘Inventing’ Theorem Proofs

Marta Fraňová and Yves Kodratoff

Equipe Inférence et Apprentissage
Laboratoire de Recherche en Informatique, UMR8623
CNRS & Université Paris Sud
Bât. 490, 91405 Orsay, France
mf@lri.fr, yk@lri.fr

Abstract. We provide a precise illustration of what can be the idea of “computational creativity”, that is, the whole set of the methods by which a computer may simulate creativity. This paper is centered on the relationship between computational creativity and theorem proving. The basic tool for this kind of computational creativity is what we call an ‘asset generator’ a specification of which is given in section 5, followed by a short description of our methodology for the generation of assets in theorem proving. In a sense, our ‘asset generation methodology’ relies essentially on making explicit the logician’s good sense while performing a recursion constructive proof. Our contribution is making explicit this good sense and making a systematic methodology of it.

1 Introduction

The goal of this paper is presenting the sketch of a methodology of computational creativity, in the field of theorem proving. As an illustration, consider the problem of a recursive definition of the exponential, which will be the current one of this paper. The classical definition of the exponential is: $a^1 = a$, $a^{n+1} = a * a^n$. It is then trivial to find and program a recursive definition, $\text{expl}(n,a)$, which is recursive on the exponent, it is:

Base case: $\text{expl}(1,a) = a$

Recursive case : $\text{expl}(n+1,a) = a * \text{expl}(n,a)$

It follows that any model of the real world that uses the exponential and this definition will compute variations of a^n by varying n instead of a . In the cases where the data are given by variations of ‘ a ’, not of ‘ n ’, and for a number of different reasons it might be advantageous to be able to program as close as possible to the data, by using a definition recursive on ‘ a ’ rather than on ‘ n ’. We meet here a typical ‘recursive problem’: knowing the definition of $\text{expl}(n,a)$ where ‘ n ’ is the exponent of ‘ a ’, how can we invent a definition of $\text{expl}(n,a)$ where ‘ n ’ is still the exponent of ‘ a ’, but the recurrence is on ‘ a ’ instead of ‘ n ’? Human ingenuity may be able to solve this problem (which is not so easy to solve as we shall see). This paper will show how it is possible for this particular problem to mimic the creative reasoning in a computable

way. We have no place here to show the whole solution given in Fraňová (2009). We will nevertheless use parts of the proof as detailed examples of our methodology.

We gather these heuristics in three groups: the generalization/particularization methods, increasing the amount of background knowledge used and the discovery of new knowledge (the ‘assets’) relative to the domain. We can imagine that ‘real’ creativity belongs to the discovery of assets. One of our aims is to show that computational creativity merges the three approaches.

2 Generalization/Particularization

The problem of finding a proper generalization is also met in theorem proving (see Bundy (2001)), for instance, when none of the variables of the formula to prove is a good candidate to become induction variable. We shall illustrate with more details in section 4.1 the way we react to this situation. The position we adopt here is that this situation is a hint to suggest that a generalization is necessary for the proof to go on.

We must at first acknowledge that spotting possible generalizations (namely, terms that are repeated within the formula to prove) is quite trivial. The difficult problem is the one spotting a non absurd one. This section will describe as rapidly the method we suggest in order to provide an informal proof of the validity of a generalization. This problem is extremely difficult and has been worked upon some 30 year ago by one of us (Kodratoff, 1979), (Arsac et al., 1982). The solution we proposed at the time can hold in this simple sentence: “Try to obtain recurrence relations among the variables and terms of by matching each pair of items in the sequence, until you obtain constant substitutions of the form ‘ t_i is substituted by t_{i+1} and $t_i = t_{i+1}$ ’, that is, all substitutions between the i -th and the $(i+1)$ -th items are constant.” We shall explain, by using the following example, that we do not try to solve this problem but a much simpler one, namely: “When observing an infinite sequence of formulas, what are the variables and terms that are already constant when comparing the 1st and the 2nd terms of the sequence?”

As explained in section 4.3, the building of a definition of exponential recursive relative to the variable under the exponent (called ‘ a ’ in the introduction) generates the following intermediary lemma:

$$L2: \forall m \forall x \forall e \exists z, x * (\exp1(m,x) + e) + \underline{(\exp1(m,x) + e)} = x * \exp1(m,x) + z.$$

Motivation for undertaking a generalization:

We shall explain with more details in section 4.1 that we are motivated to perform a generalization because we observe that our definition of the function ‘+’, given in the recursive case as A4: $(n+1) + y = (n + y)+1$, does not allow us a recurrence on the second variable (called ‘ y ’ in A4). Thus, choosing the induction variable ‘in position of y ’ in L2, underlined above, will not be useful for the remaining of the proof. However, evaluating repetitively the recursive definition of $\exp1$, namely A2: $\exp1(n+1,a) = a * \exp1(n,a)$, will provide an infinite sequence of lemmas to prove. As we said, evaluating the limit of this infinite sequence is very difficult. We simply want to find the terms that undergo a constant substitution between the 1st and 2nd items of this sequence. These terms are good candidates for being generalized by the same variable. Obviously, this does not prove that we can generalize them, it simply eliminates the false generalizations that would lead to a lemma that would lead to a failure.

For the sake of convenience, let us write the first item of the sequence at hand as: $t_1 = x * (\text{expl}(m,x) + e) + 1 * (\text{expl}(m,x) + e) = x * \text{expl}(m,x) + z$. Applying axiom A4 for $m := m+1$ gives the second term: $t_2 = x * (x * \text{expl}(m,x) + e) + (x * \text{expl}(m,x) + e) = x * x * \text{expl}(m,x) + z$. Matching them provides an obvious set of substitutions: $x \leftarrow x * x$, $\text{expl}(m,x) \leftarrow \text{expl}(m,x)$, $e \leftarrow e$, $1 \leftarrow x$, $\text{expl}(m,x) \leftarrow \text{expl}(m,x)$, $e \leftarrow e$, $x \leftarrow x * x$, $\text{expl}(m,x) \leftarrow \text{expl}(m,x)$, $z \leftarrow z$.

It follows that the general form of t_n will be

$$t_n = f1(x) * (\text{expl}(m,x) + e) + (f2(x) * \text{expl}(m,x) + e) = f1(x) * \text{expl}(m,x) + z.$$

as can be checked by observing the other items of the sequence. Note also that using the equation $1 * y = y$ is suggested by the need to have constant substitutions starting from the first item.

We observe that the term $\text{expl}(m,x)$ undergoes a constant substitution, it can thus be generalized to one variable w . In the contrary case, we would have to generalize it to several different variables. Finally, we will study the generalized form of L2, L3: $\forall w \forall x \forall e \exists z, x * (w + e) + (w + e) = x * w + z$. L3 is an example of an “invented” theorem.

An example of a particularization in theorem proving is provided in Fraňová et al. (2009).

3 Increasing Domain Knowledge

We must at first make a clear difference between domain knowledge (also called ‘background knowledge’) and the invention of assets. The assets are a special kind of domain knowledge which is invented during the proof and without which the proof would be impossible. Inversely, what is usually called domain knowledge is the bulk of all what is ‘well-known’ in the field. Domain knowledge is believed to be always an asset but it is also well-known that an excess of knowledge may totally clog the proofs because the order in which the knowledge is applied becomes significant. Determining the good ordering of application of domain knowledge may often become a problem by itself, and one more difficult to solve than the initial problem.

In the domain of formal theorem proving, let us give three extremely simple examples of domain knowledge that will be necessary to achieve the construction of a new recursive definition of the exponential we are here looking for.

3.1 Some Basic Knowledge

Case 1. How happens that computing ‘ z ’ defines the function we are looking for? This follows from a basic property of formal theorem proving. Let X be the vector of input variables, and Z the vector of output variables of the relation $R(X,Z)$. Suppose we prove that $\forall X \exists Z R(X,Z)$. Then, the function that realizes the computation of Z is called the Skolem function of X , $Sf(X)$, and is defined by: $Sf(X) = Z$ and $\forall X R(X,Sf(X))$. In the following, we shall stress the computational quality of the Skolem function by naming it “aux-i” since it defines an auxiliary function needed to solve our problem.

Performing a proof by recursion consists in the analysis of what are called “base case” and “general case” and to apply to the general case what is called the “induction hypothesis.” Since we restrict ourselves here to the natural numbers, the base case is $x = 1$ since the simplest formula we start from includes no conditions on x . The general case is $x = n+1$ where ‘+1’ is the successor function. The induction hypothesis (“if the formula is assumed to be true for n , then we can prove that it is true for $n+1$ ”) writes $\exists e \ R(n,e) \Rightarrow \exists z \ R(n+1,z)$ or else, using a Skolem function, $R(n,Sf(n)) \Rightarrow R(n+1,Sf(n+1))$.

Case 2. Trivial recurrence

In order to simplify notations, suppose that we study a binary function $f(n,y)$ where ‘ n ’ is the induction variable. Suppose further that we find out that $f(1,y) = y + a$ and $f(n+1,y) = f(n,y)$, where a is a constant. Then recurrence is removed since we know that $\forall n \ f(n,y) = y + a$. If $f(n+1,y) = f(n,y)+1$, then $\forall n \ f(n,y) = n + (y + a)$.

3.2 Jump Operator

Suppose we are in a situation where, in some way, a generalized ‘less than’, \angle , operator can be defined. If we know that $x \angle y$, it is then always possible to state that $\exists z, x + z = y$, and the proof on this existential theorem will provide us the value of z . Here are three trivial but important cases where we apply this ‘jump operator’.

Case 1. Since the exponential function is increasing, we know the $\forall x \ \forall y \ \exp1(x,y) \angle \exp1(x,y+1)$. Using the jump operator, we know that $\forall x \ \forall y \ \exists z \ \exp1(x,y+1) = \exp1(x,y) + z$ is true and that proving this theorem will enable us to build a suitable ‘ z ’.

Case 2. Another important, if not obvious, case of application of the ‘jump operator’ occurs when we observe that applying the induction hypothesis is not possible in a formula that is an equation. Our solution is to generate an intermediary lemma. Let us write the induction hypothesis as $h_i(n,X) = h_r(n,X')$, where X and X' are vector variables. The general case is given by $c_i(n+1,X) = c_r(n+1,X')$. When the induction hypothesis cannot be directly applied to the generated case, we generate the intermediary lemma $\forall n \ \forall X \ \exists z \ c_i(n+1,X) = h_i(n,X) + z$.

3.3 What and When Is It Necessary to Evaluate?

The reader may have already noted that our methodology tries to avoid a systematic use of all the properties of the functions defined by the axioms. The reason of this restraint is that such a use may dissimulate generalizations for completing the proof or may complicate *ad absurdum* the application of the induction hypothesis. This is why we do apply systemically the definition axioms and the induction hypothesis. For the last, we even go up to ‘inventing’ an intermediary lemma to prove that the induction hypothesis may be applied.

Inversely, we ‘restrain’ as much as possible the evaluation of the well-known properties of the functions defined by the axioms. Actually, we did not find a case, until now, where our methodology cannot skip this evaluation. We can always introduce intermediary lemmas that will, in a sense, prove again the property needed

exactly when it is needed. The price to pay for a blind application of this choice is a cumbersome accumulation of intermediary lemmas. This is why we use a strategy of lazy evaluation where the properties of the functions are used only when an intermediary lemma is created and that this last one can be proved by a straightforward evaluation.

4 Discovering New Knowledge about the Domain (The Assets) and Failure Analysis in PSFS

4.1 Failure Analysis in PSFS

The generation of assets is based on failure analysis. This is a whole domain of research and we shall here give a simple example of failure analysis, namely, spotting the variables that can or cannot be used as induction variables.

In section 2.1, our motivation for attempting a generalization is that no variable is suitable to become the induction variable. This is quite obvious when we observe a recursive definition. For instance, if we define the general case of the addition by A4: $(n+1) + y = (n + y) + 1$, then ‘n’ is the variable suitable to become the induction variable, not y. This is slightly less obvious when we consider function embedded into another one.

For example, in the term $Ex: (n + y) + (m + z)$, ‘n’ is indeed suitable to become the induction variable. Inversely, if we attempt to use ‘m’ as the induction variable, the general case will include the study of $m := (m+1)$. Thus Ex will become $(n + y) + ((m+1) + z)$, and by evaluation using A4, $(n + y) + ((m + z) + 1)$, and we gain nothing more that could help us to apply the induction hypothesis. The only solution we will have is to generate a new intermediary lemma enabling us to apply the induction hypothesis. Obviously, when we cannot apply the induction hypothesis there is no hope to solve the recursive problem at hand! Thus, ‘m’ is not suitable to become the induction variable.

This explains one of our choices, as presented in section 2.1. When no variable is suitable for becoming the induction variable, we try to generalize the formula under study. It may happen that no generalization is obvious or, as it is often the case, the generalization also shows no variable suitable for becoming the induction variable. Since we have no other choice, we accept to choose one induction variable, knowing ahead of time that this leads us to introduce yet another intermediary lemma.

In order to generate assets during our proofs, we need to add two steps to the usual recursion proofs. The one is the management of a kind of stack into which we pile up the conditions needed for solving the problem at hand. We call this process “**introducing abstract arguments.**” The second one is a heuristic helping us to generate lemmas such that their proof enables to go on in the proving process. The generation of these special assets is called “**generation of intermediary lemmas.**”

4.2 Introducing the ‘Abstract Arguments’

We introduce a new type of argument in the predicates a feature of which has to be proven true, we call **abstract arguments**. They are denoted by ξ (or ξ' etc.) in the following.

Problem 1. Building the abstract argument

It replaces one of the arguments of the base theorem. The first step is choosing which of the arguments of the base theorem will be replaced by an abstract argument, ξ . This argument is known and, in a usual proof, its characteristics are used in order to prove the base theorem. In our approach, we ‘forget’ for some time these characteristics and we concentrate on studying the features ξ should have so as insuring that the theorem with a substituted argument is true.

In the following, and for the sake of avoiding a too general wording, suppose for example that the formula to prove has two arguments, that is to say that we need to prove that $F(t_1, t_2)$ is true, where F is the base theorem. Suppose further that we have chosen to work with $F(\xi, t_2)$. We shall then look for the features shown by all the ξ such that $F(\xi, t_2)$ is true.

At first, we have to choose which argument will be made abstract. There are two ways to introduce an abstract argument, and we thus start with either $F(t_1, \xi)$ or $F(\xi, t_2)$ since, obviously, $F(\xi, \xi')$, an *a priori* possible choice, would hide all the characteristics of t_1 and t_2 .

Supposing we are able to find the features of ξ such that (say) $F(t_1, \xi)$ is true, for all the ξ showing these features, $F(t_1, \xi)$ is true. In other words, calling ‘cond’ these features and C the set of the ξ such that $\text{cond}(\xi)$ is true, we define C by $C = \{\xi \mid \text{cond}(\xi)\}$. We can also say that we try to build a ‘cond’ such that the theorem: $\forall \xi \in C, F(t_1, \xi)$ is true. It is reasonable to expect that this theorem is much more difficult to prove than $F(t_1, t_2)$. We thus propose a ‘detour’ that will enable us to prove the theorems that cannot be directly proven, without this ‘detour’. Using the characteristics of C and the definition axioms in order to perform evaluations, and also using the induction hypothesis, we shall build a form of ξ such that $F(t_1, \xi)$. Even though it is still ‘ ξ ’ and only for the sake of clarity, let us call ξ_C one of these forms. It is thus such that $F(t_1, \xi_C)$. We are still left with a hard work to perform: Choose the ‘good’ ξ_C in the set C and modify it (possibly using the induction hypotheses), in such a way that ξ_C and t_2 will be made identical, which finally completes the proof.

In Fraňová et al. (2009), we explain how our methodology deals with proving that, for the natural numbers, $2 < 4$. Here F is the predicate ‘ $<$ ’, $t_1 = 2$ and $t_2 = 4$. In this example, we replace the second argument by ξ and we study the characteristics of ξ such that $2 < \xi$. As already said, this is a generalization of the base theorem. Note also that we temporarily forget the base theorem and that we focus on the study of the ξ such that $2 < \xi$.

Problem 2. How to use the abstract argument

Suppose we started with $F(t_1, \xi)$. Since we wish to build solutions enabling us to prove the theorem, the construction process will include checking the likeness of ξ and t_2 . The general rule we use during these likeness checks is quite obvious. Suppose that t_2 has the form $f(p, q)$. Then ξ cannot be matched but with a function as $f(x, y)$ where x and y are variables. Thus, we reach the problem of proving $\exists u \exists v, \xi = f(u, v)$. More generally, the failure of a matching between ξ and t_2 leads us to introduce existentially quantified variables that insure the success of the matching of ξ and t_2 . The automation of this reasoning step can become very complex and shows many different cases, but each case is quite general and is trivial to solve. For example, suppose that during an evaluation step, we realize that in order to prove the theorem, we have to identify ξ and $s(u)$, then

we will make the hypothesis that there indeed exists a ‘u’ such that $\xi = s(u)$. This relation is looked upon as a condition on ξ : ξ is such that $\exists u, \xi = s(u)$.

4.3 The Generation of Intermediary Lemmas

It is not enough to introduce the existential lemmas we noticed just above. Besides the variables necessary to insure the matching of ξ and t_2 , the base theorem included other variables, in particular in t_2 . If these variables were universally quantified then what we call an intermediary lemma needs also a universal quantification of these variables. Moreover, the base theorem we need to prove looks like $\forall x G(x, \xi)$. By the manipulations ξ undergoes, we discover that the relation G' has to be true, so that $\exists u G'(x, u)$. Since x is universally quantified during all the manipulations we did, the intermediary lemma is thus $\forall x \exists u G'(x, u)$, in which the quantifiers are obviously put in this order. Depending on the conditions on ξ , the intermediary lemmas are not of the same form in the choice of the variables to quantify. We are not yet able to provide a method to generate a complete list of all the possible forms. The discovery of the most suitable forms still relies entirely on human creativity. We shall see, however, that each form can be applied to many particular cases.

Here is an example of the generation of an intermediary lemma, which illustrates how a failure case is dealt with during the building of our version of the exponential.

In order to invent a new recursive definition of the exponential, we need, among others, to prove the following theorem:

$$\forall n \forall x \exists z \text{expl}(n, x+1) = \text{expl}(n, x) + z.$$

Suppose that the general case of the axioms defining the multiplication and the classical form of the exponential are A2: $\text{expl}(n+1, a) = a^* \text{expl}(n, a)$, and A6: $(n+1)^*y = (n^*y) + y$. Let us study the general case and apply the induction hypothesis. Let $n = m+1$, the induction hypothesis is:

$$\begin{aligned} \exists e \text{expl}(m, x+1) &= \text{expl}(m, x) + e. \text{ Using it, we have to prove that} \\ \exists z \text{expl}(m+1, x+1) &= \text{expl}(m+1, x) + z = x^* \text{expl}(m, x) + z, \text{ (by A2).} \end{aligned}$$

Let us treat the equality as any other operator and replace the left side of the equation (i.e. the second argument of the equation, the argument which contains an existentially quantified variable) by the abstract argument ξ . We thus have $\text{expl}(m+1, x+1) = \xi$. We thus shall now study the properties of $C_\xi = \{\xi \mid \xi = \text{expl}(m+1, x+1)\}$ and, once this study is completed, check whether the other side of the equality, namely ‘ $x^* \text{expl}(m, x) + z$ ’, belongs to C_ξ . Let us evaluate ξ . By A2, we obtain $(x+1)^* \text{expl}(m, x+1) = \xi$. By A6, $x^* \text{expl}(m, x+1) + \text{expl}(m, x+1) = \xi$. Thus $C_\xi = \{\xi \mid \xi = x^* \text{expl}(m, x+1) + \text{expl}(m, x+1)\}$. Applying the induction hypothesis, we obtain: $C_\xi = \{\xi \mid \xi = x^*(\text{expl}(m, x) + e) + (\text{expl}(m, x) + e)\}$. The question to ask now is: does ‘ $x^* \text{expl}(m, x) + z$ ’ belongs to C_ξ ? This is not at all obvious, unless there exists a z such that $x^*(\text{expl}(m, x) + e) + (\text{expl}(m, x) + e) = x^* \text{expl}(m, x) + z$. Thus, the analysis of the properties of ξ leads us to discover that we need to prove the intermediary lemma, the one we generalized in section 2.

$$L2: \forall m \forall x \forall e \exists z, x^*(\text{expl}(m, x) + e) + (\text{expl}(m, x) + e) = x^* \text{expl}(m, x) + z.$$

This lemma is generalized as explained in section 2. This generalization still has no variable suitable to become the induction variable. We thus know that attempting to use the induction hypothesis will need the generation of a new intermediary lemma. The proof goes on. You will find in Fraňová (2009) a complete solution for the problem of finding a new recursive definition of exponential leading the following definition:

$$\begin{aligned} \text{exp2}(n,1) &= 1 \\ \text{exp2}(n,x+1) &= \text{exp2}(n,x) + \text{aux2}(n,x) \\ \text{aux2}(1,x) &= 1 \\ \text{aux2}(n+1,x) &= \text{aux3}(\text{exp1}(n,x),x, \text{aux2}(n,x)) \\ \text{aux3}(w,0,e) &= w + (e + e) \\ \text{aux3}(w,n+1,e) &= e + \text{aux3}(w,n,e) \end{aligned}$$

You will find in Fraňová et al. (2009), a more complex example, namely an original result for the Ackermann function.

5 Conclusion, the Technical and the Philosophical

Most of the techniques illustrated here: the ‘abstract argument’ technique, the ‘intermediary lemma’ and even our ‘asset generator’ may seem a result of simple logical good sense, more than a real forward step in theorem proving. We partly agree since we are convinced that, as a matter of fact, most humans who prove theorems go through one or the other of these techniques. The new result we claim is to have put them in a coherent methodology based on the ‘abstract argument’ technique, from which all other techniques are called when they are needed. As we underlined when presenting it, it is clear that our methodology asks for a fair amount of work since many ‘intermediary’ lemmas can be generated depending on the matching we want to prove to be possible, and each of them has to be disproved before attempting to prove another one. Creativity is expensive; we have to acknowledge this fact. Relations with Program Synthesis have been already provided in Franova et. al, 1993. However, we still have to study the relations between the costs presented here and the efforts generally needed in computational creativity viewed as a general research field.

From a more philosophical point of view, this paper makes it clear that ‘creativity’ is, but is not only, what most people claim it is: ‘leaving aside the ordinary roads to wander on new ways’. It is true that we did so: When evaluation techniques fail to give a result, perform your own “Copernicus revolution” by putting evaluation of the properties away and concentrate on the evaluation of the axioms and on the application of the induction hypothesis. This paper shows also clearly that this is far from enough.

One supplementary condition is isolating the parts of domain knowledge, usually deemed as of secondary importance, and make a systematic use of these parts (as exemplified in section 3). Even the use of ‘abstract arguments’ is so often trivial that it can be seen as an obvious domain knowledge. Note however that when its use is not trivial, it amounts to state that some failed matchings can be ‘repaired’ by proving the equivalence between the two terms that failed to be matched. In other words, it is really “forcing the matching” without an improper use of the words, a far from trivial statement.

The second supplementary condition stems from the fact that unknown ways always lead to unknown dangers. In the case of theorem proving, the danger is that new problems are generated, which could be more difficult to solve than the initial problem. This happens in our case when we generate an infinite sequence of lemmas. We could have stated that it was ‘simply’ necessary to compute the limit of this sequence, a problem with no known general solution. As you have seen, we have been able to transform this problem in the much simpler one of spotting the terms that will be kept constant during the generation of the sequence. These terms are good candidates for being changed into a variable, and undergoing a generalization.

References

- Arsac, J., Kodratoff, Y.: Some Techniques for Recursion Removal from Recursive Functions. *ACM Transactions on Programming Languages and Systems* 4(2), 295–322 (1982)
- Bundy, A.: The Automation of Proof by Mathematical Induction. In: Robinson, A., Voronkov, A. (eds.) *Handbook of Automated Reasoning*, vol. I, pp. 845–912. North-Holland, Amsterdam (2001)
- Fraňová, M.: A Construction of Several Definitions Recursive over the Variable under the Exponent for the Exponent function. *Rapport de Recherche No.1519, L.R.I., Université de Paris-Sud, Orsay, France (June 2009)*
- Fraňová, M., Kodratoff, Y., Gross, M.: Constructive Matching Methodology: Formally Creative or Intelligent Inductive Theorem Proving? In: Komorowski, J., Raś, Z.W. (eds.) *ISMIS 1993. LNCS (LNAI)*, vol. 689, pp. 476–485. Springer, Heidelberg (1993)
- Fraňová, M., Kodratoff, Y.: La “créativité calculatoire” et les heuristiques créatives en synthèse de prédicats multiples. In: Ganascia, J.-G., Gançarski, P. (eds.) *Extraction et gestion des connaissances: EGC 2009, Revue des Nouvelles Technologies de l’Information, RNTI-E-15, Cépades*, pp. 151–162 (2009)
- Kodratoff, Y.: A class of functions synthesized from a finite number of examples and a LISP program scheme. *International J. of Computational and Information Science* 8, 489–521 (1979)
- Peter, R.: *Recursive Functions*. Academic Press, New York (1967)

Revisiting Constraint Models for Planning Problems

Roman Barták¹ and Daniel Toropila^{1,2}

¹ Charles University, Faculty of Mathematics and Physics, Malostranské nám. 2/25,
118 00 Praha 1, Czech Republic

{Roman.Bartak, Daniel.Toropila}@mff.cuni.cz

² Charles University, Computer Science Center, Ovocný trh 5,
116 36 Praha 1, Czech Republic

Abstract. Planning problems deal with finding a sequence of actions that transfer the initial state of the world into a desired state. Frequently such problems are solved by dedicated algorithms but there exist planners based on translating the planning problem into a different formalism such as constraint satisfaction or Boolean satisfiability and using a general solver for this formalism. The paper describes how to enhance existing constraint models of planning problems by using techniques such as symmetry breaking (dominance rules), singleton consistency, nogoods, and lifting.

Keywords: Planning, constraint models, symmetry breaking, lifting.

1 Introduction

Planning is an important aspect of rational behavior and has been a fundamental topic of artificial intelligence since its beginning. Many approaches exist for solving planning problems; one of them is translation of the problem to a different formalism such as constraint satisfaction and using solving techniques developed for this formalism. In this paper we propose several improvements of the constraint model for solving classical AI (artificial intelligence) planning problems. The original base model was proposed in [1] and it used a straightforward encoding of the planning problem. Later, we proposed three enhancements of this model [2]. In this paper we revise one of these enhancements (symmetry breaking); we add one more improvement (nogood learning) and do incremental experimental comparison of influence of all these enhancements on the efficiency of the base model. Two of the proposed improvements are based on techniques used in constraint satisfaction. In particular, we use singleton consistency to prune more of the search space and we use dominance rules to break plan permutation symmetries in the problem. The third improvement called lifting has been introduced in planning to decrease the branching factor during regression (backward) search. Finally, our last improvement, nogood learning, exploits the knowledge gained during the search in order to efficiently prune the search space.

The paper is organized as follows. We will first describe the base constraint model used in our research. This is the best model from [1] that was obtained by reformulating constraints from the original model proposed in [3]. After that, four enhancements of this model will be described and finally these enhancements will be experimentally compared using several problems from the International Planning Competition.

2 Constraint Models for Planning Problems

One of the difficulties of planning is that the length of the plan, that is, the sequence of used actions, is unknown in advance so some dynamic technique which can produce plans of “unrestricted” length is required. As shown in [4], the problem of shortest-plan planning can be translated to a series of SAT problems, where each SAT instance encodes the problem of finding a plan of a given length. Similarly, this problem can be encoded as a constraint satisfaction problem (CSP). Then the following iterative technique can be used to find the shortest plan. A specific constraint model is first used to encode the problem of finding a plan of length n (starting with $n = 0$). Then the search for the plan is performed. In case of success, the plan found is returned, otherwise the encoding for the problem of finding a plan of length $n+1$ is constructed (by extending the previous encoding with the new layer of variables and constraints, not building it from scratch). The whole process is repeated until the plan is found or computation runs out of time or another termination condition applies.

2.1 Existing Approaches

There exist several constraint models for the problem of finding a plan of length n . All of them share the idea of using a set of variables describing the states of the world and a set of variables describing selected actions. In [1] we described these models using the *multi-valued variable representation* (SAS+), as mentioned in [5] or [6]. World state is modelled using v multi-valued variables, instantiation of which exactly specifies a particular state. The actions are then changing the values of state variables. Each action consists of the preconditions specifying required values of certain state variables and the effects of setting the values of certain state variables (Fig. 2).

A CSP modelling the problem of finding a plan of length n consists of $n+1$ sets of the above-mentioned multi-valued variables, having 1st set denoting the initial state and k^{th} set denoting the state after performing $k-1$ actions, for $k \in \langle 2, n+1 \rangle$, and of n variables indicating the selected actions. Hence, we have $v(n+1)$ state variables V_i^s and n action variables A^j , where i ranges from 0 to $v-1$, j ranges from 0 to $n-1$, and s ranges from 0 to n (Fig. 1). The particular constraint models differ mainly in the set of constraints specifying the relations between the variables. These constraints connect two adjacent sets of state variables through the corresponding action variable between them, describing thus change between states if a particular action is selected.

In this paper, we use the ideas proposed in CSP-PLAN [3] and then re-formulated in [1] where authors also show that the CSP-PLAN-based model provides the best performance among the reviewed models.

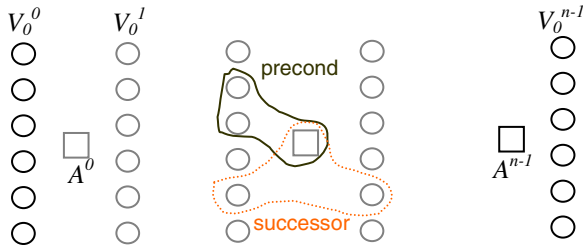


Fig. 1. Base decision variables and constraints modeling sequential plans

First, CSP-PLAN models the action preconditions using a set of constraints – for a given layer s we connect state variable layer $V_i^s, i \in \langle 0, v-1 \rangle$ with action variable A^s :

$$A^s = act \rightarrow \text{Pre}(act)^s, \forall act \in \text{Dom}(A^s). \tag{1}$$

Rather than encoding the effects of actions (similarly to (1)) and the frame axiom separately as in the straightforward constraint model from [7], the CSP-PLAN model uses *successor state constraints* originally described in [8]. The successor state constraints merge the effect constraints and the frame axioms together as follows: for each possible assignment of state variable $V_i^s = val, val \in \text{Dom}(V_i^s)$, we have a constraint between it and the same state variable assignment $V_i^{s-1} = val$ in the previous layer. The constraint says that state variable V_i^s takes value val if and only if some action assigned this value to the variable V_i^s , or equation $V_i^{s-1} = val$ held in the previous layer and no action changed the assignment of variable V_i . Formally:

$$V_i^s = val \leftrightarrow A^{s-1} \in C(i, val) \vee (V_i^{s-1} = val \wedge A^{s-1} \in N(i)). \tag{2}$$

In the formula above, $C(i, val)$ denotes the set of actions containing $V_i = val$ among their effects, and $N(i)$ denotes the set of actions that do not affect V_i .

2.2 Base Model

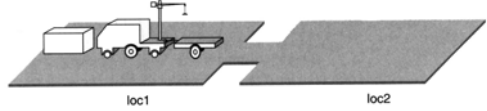
In [1] we identified several problems of the above model, namely the large number of constraints and weak domain filtering. The large number of constraints means that the consistency procedure runs longer as it needs to check consistency of all constraints. Weak domain filtering is mainly due to a disjunctive character of the above logical constraints (implication is a syntactic sugar for disjunction) – due to efficiency issues, most existing constraint solvers do not achieve full arc consistency for disjunctive constraints. Moreover, there are many constraints with the same scope (a set of constrained variables), which also contributes to weak domain filtering (each constraint is processed separately in AC).

To overcome these problems, in [1] we adopted the approach of substituting some of the above-mentioned propositional formulae using the constraints with extensionally defined set of admissible tuples (sometimes also called *combinatorial constraints*). The idea was to union the scope of “similar” constraints and to define the admissible tuples in extension rather than using a formula. Such types of constraints

are frequently called *table constraints*, because the set of admissible tuples is given in a table-like structure (Fig. 2). The experiments showed that the reformulated version of CSP-PLAN leads to better efficiency [1]. Thus, we decided to use it as the base model for further enhancements.

Domain
 DWR domain with two locations (*loc1,loc2*), a robot capable of loading and unloading containers by itself (*r*), and one container (*c*)

State Variables
rloc ∈ {*loc1,loc2*} ;; robot's location
cpos ∈ {*loc1,loc2,r*} ;; container's position



- Actions**
- 1: *move(r, loc1, loc2)*
 ;; robot *r* at location *loc1* moves to location *loc2*
 Precond: *rloc* = *loc1*
 Effects: *rloc* ← *loc2*
 - 2: *move(r, loc2, loc1)*
 ;; robot *r* at location *loc2* moves to location *loc1*
 Precond: *rloc* = *loc2*
 Effects: *rloc* ← *loc1*
 - 3: *load(r, c, loc1)*
 ;; robot *r* loads container *c* at location *loc1*
 Precond: *rloc* = *loc1, cpos* = *loc1*
 Effects: *cpos* ← *r*
 - 4: *load(r, c, loc2)*
 ;; robot *r* loads container *c* at location *loc2*
 Precond: *rloc* = *loc2, cpos* = *loc2*
 Effects: *cpos* ← *r*
 - 5: *unload(r, c, loc1)*
 ;; robot *r* unloads container *c* at location *loc1*
 Precond: *rloc* = *loc1, cpos* = *r*
 Effects: *cpos* ← *loc1*
 - 6: *unload(r, c, loc2)*
 ;; robot *r* unloads container *c* at location *loc2*
 Precond: *rloc* = *loc2, cpos* = *r*
 Effects: *cpos* ← *loc2*

Table for precondition constraint

A^s	$rloc^s$	$cpos^s$
1	loc1	{loc1,loc2,r}
2	loc2	{loc1,loc2,r}
3	loc1	loc1
4	loc2	loc2
5	loc1	r
6	loc2	r

rules for successor state constraint

A^s	$rloc^s$	$rloc^{s+1}$	A^s	$cpos^s$	$cpos^{s+1}$
2	{loc1,loc2}	loc1	5	{loc1,loc2,r}	loc1
1	{loc1,loc2}	loc2	6	{loc1,loc2,r}	loc2
{3,4,5,6}	loc1	loc1	{3,4}	{loc1,loc2,r}	r
{3,4,5,6}	loc2	loc2	{1,2}	loc1	loc1
			{1,2}	loc2	loc2
			{1,2}	r	r

Fig. 2. Example of constraint model using combinatorial constraints (domain taken from [7])

2.3 Base Search Strategy

Constraint modeling is an important step, but it is also necessary to specify a search strategy for instantiating the variables. One can use generic labeling techniques, for example based on dom heuristics (select the variable with the smallest domain first), but our first experiments showed that this is not efficient for the proposed constraint model. First, one should realize that it is enough to instantiate just the action variables A^s because when their values are known, then the values of remaining variables, in particular the state variables, are set by means of constraint propagation. Of course, we assume that the values for state variables V_i^0 modeling the initial state were set and similarly the state variables V_i^n in the final layer were set according to the goal (the final state is just partially specified so some state variables in the final layer are un-instantiated at the beginning).

We utilized a regression planning approach in the search strategy meaning that we instantiate the action variables in the decreasing order from A^{n-1} to A^0 . This is called a fixed variable ordering in constraint satisfaction. For each action variable we assume only actions that contribute to (sub)goal in the next state layer – these actions are called relevant in [7]. The actions (values) in the action variable are explored in the

order of appearance in the plan – the action that appeared later in the plan is tried first for instantiation.

3 Model Enhancements

As shown in [1], the reformulation of constraints from the logical form to the combinatorial form reduced significantly the runtime for solving planning problems. The question, which we started to answer in [2] and are further answering in this paper, is whether it is possible to improve efficiency by including more advanced solving techniques. In this section we describe four enhancements motivated by the existing techniques from planning and constraint satisfaction.

3.1 Lifting

The search strategy used for the base model resembles the labeling technique in constraint satisfaction. At each step, we select an action that contributes to the current goal. Assume that there are 100 locations and part of the goal is that the robot must be at a specific location. When selecting the move action to satisfy this goal, we actually determined also the location from which the robot goes. This might be too restrictive, because we are building the plan from the end (backward planning), so we may find later that it is not possible to get the robot to the required start location. Hence, we need to backtrack to the choice point, where the move action was selected, and try a different one. Notice that we have 100 options at that choice point and the decision must be done without information which start location is the best one.

We propose to postpone the decision to the point when more information is available. This is called *lifting* in the planning community [7]. This idea can be easily realized by modification of the search strategy. Rather than selecting a particular move action, we reduce the domain of the action variable to all move actions that lead to a given location. In terms of constraint satisfaction, we split the domain into two parts: one with the compatible move actions and one with the remaining actions that can be used at that step of the plan. As the search proceeds, the domain of the action variable may be further reduced via maintaining arc consistency (the later chosen actions may determine possible start locations for the move action). Still, it may happen that when reaching the initial state, some actions are not decided yet. In such a situation, we apply the standard labeling procedure to instantiate the action variables.

Let us now describe the process of lifting more formally. Let $\text{PrecVars}(a)$ be the state variables appearing in the precondition of action a and $\text{EffVars}(a)$ be the state variables changed by action a (these variables appear in effects of a). We say that actions a and b have the same scope if and only if $\text{PrecVars}(a) = \text{PrecVars}(b)$ and $\text{EffVars}(a) = \text{EffVars}(b)$. Let the base search procedure selects action a to be assigned to variable A^s ; in other words we split the search space by resolving the disjunction $A^s = a \vee A^s \neq a$. In the lifted version, we are resolving the following disjunction:

$$A^s \in \text{SameScope}(a) \vee A^s \notin \text{SameScope}(a), \quad (3)$$

where $\text{SameScope}(a) = \{ b \mid b \text{ has the same scope as } a \}$.

3.2 Dominance Rules (a.k.a. Symmetry Breaking)

Recall, that we are looking for sequential plans, that is, for a sequence of actions. Assume that we have two actions a_1 and a_2 such that these actions do not interfere, for example, the move action of a robot and the load action of a different robot. If we have a valid plan where a_1 is right before a_2 then a plan where we swap both actions is also valid. This feature, called *plan permutation symmetry* [9], can be exploited during search in the following way. Assume that at some stage of search we selected a_1 to be right before a_2 at some position of the plan. If this decision leads to a failure (no complete plan was found) then it is not necessary to explore plans where a_2 is right before a_1 at the same position because such plans will also be invalid. This feature can be used to prune the search space by omitting exploration of symmetrical plans. Another way to re-solve the very same problem is allowing parallel actions like in Graphplan [10] or using partial order plans [11].

First, we need to define formally what it means that two actions do not interfere. Recall that our motivation is that for a sequence of actions where a_1 is right before a_2 these two actions can be swapped without influencing validity of the plan (a_2 would be right before a_1). Swapping of actions a_1 and a_2 can be realized if for any state s the following condition holds: $\gamma(\gamma(s, a_1), a_2) = \gamma(\gamma(s, a_2), a_1)$, where $\gamma(s, a)$ is a state obtained by applying action a to state s . Such situation happens if actions a_1 and a_2 are *independent* [7]. However, it is sufficient for our purpose to use a condition that is more relaxed than the action independence in order to achieve stronger pruning.

Before we formally introduce the condition of action interchangeability, we need to describe two relations between actions. We say that effects of action a *clash* with preconditions of action b if action a assigns value val to some state variable V_i such that the assignment $V_i = val$ is inconsistent with the preconditions of action b . Also, we say that the effects of action a don't *override* the effects of action b if both actions a and b set the same value for each state variable $V \in (\text{EffVars}(a) \cap \text{EffVars}(b))$.

Now, using the multi-valued state representation, we suggest following conditions for determining the ability to swap two actions without influencing plan validity. For a sequence of actions where a_1 is right before a_2 , a_1 and a_2 can be swapped if:

$$\begin{aligned} \text{EffVars}(a_1) \cap \text{PrecVars}(a_2) &= \emptyset, \text{ (} a_1 \text{ does not provide preconditions for } a_2\text{),} \\ \text{Effects}(a_2) \text{ don't clash with Preconds}(a_1), &\text{ (placing } a_2 \text{ before } a_1 \text{ is consistent),} \\ \text{Effects}(a_1) \text{ don't override Effects}(a_2). & \end{aligned} \quad (4)$$

Note that this definition is weaker than the definition of action independence from [7] (i.e., independence implies interchangeability, but not vice versa), and that the relation is not symmetrical – the ability to swap actions a and b does not imply the ability to swap b and a .

Once we know how to recognize the swappable actions, we propose to include the following dominance rule to the search procedure. We choose an arbitrary ordering of actions such that action a_i is before action a_{i+1} in the ordering (this ordering has nothing in common with the ordering of actions in the plan). Assume that action a_i has been assigned to state variable A^s . Then, when selecting the action for state variable A^{s-l} (recall, that we are building the plan from the end), we only consider actions a_j for which at least one of the following conditions holds: either a_j and a_i are not swappable, or $j > i$. In other words, if actions a_i and a_j are swappable and $i < j$ then we

explore only the plans where a_i can be right before a_i but not vice versa. This way, we prevent the solver from exploring permutations of interchangeable actions which decreases the size of the search space.

Note that the above dominance rule can be combined with lifting presented in the previous section thanks to the definition of action interchangeability. When using lifting, we do not assign a particular action to the action variable but we restrict the domain of the action variable to a set of actions with the same scope. Hence, even if a particular action is not yet selected but a set of actions is used for A^s , we can still take action a_i with the smallest i in the domain of A^s (after domain splitting) to participate in the dominance rule.

3.3 Singleton Consistency

So far, we discussed the improvements of the search strategy. Another way to improve efficiency of constraint solving is incorporating a stronger consistency technique. Singleton arc consistency (SAC) [12] would be a good candidate because it is easy to implement on top of arc consistency. However, it is computationally expensive to maintain SAC during search or even to make the problem SAC before search. Nevertheless, we can apply the idea of SAC in a restricted form. Recall, that in SAC we assign a value to some variable and propagate this information using AC to other variables. If a failure is detected then the respective value can be removed from the domain of the variable.

Assume that we failed to find a plan of length n and hence a new layer is added to the constraint model. New action variable A^n (recall, that A^0 is the first action) is introduced and connected with the state variables (new state variables V_i^n are also introduced). After posting the new constraints involving this action variable, the problem is made AC. This removes some actions from the domain of variable A^n but according to our observations many actions that cannot be assigned to A^n still remain in the domain. To remove some of them, we propose to exploit the idea of SAC in the following way. We take all actions from the domain of A^n that do not appear in the domain of A^{n-1} . These are the newly introduced actions and we want to validate if these actions can be used at $(n+1)$ -th step of the plan. Let a be such a newly introduced action. We assign a to variable A^n and try to find action b for A^{n-1} that supports $a - b$ provides certain precondition of a (in other words, we instantiate A^{n-1} by b). If this is not possible, we can remove a from the domain of A^n , because a can never be assigned to A^n . This process is performed for every newly introduced action. The hope is that we can eliminate actions that would otherwise be tried during search.

Let us describe the above process more formally. Let P be a constraint model describing the problem of finding a plan of length $n+1$ and a be an action that appears in the domain of A^n but not in the domain of A^{n-1} (a newly introduced action). If there is no action b such that $V_i = v$ is among its effects, $V_i = v$ is among preconditions of a , and $Pl_{A^n=a, A^{n-1}=b}$ is arc consistent then a can be removed from the domain of A^n ($Pl_{x_i=a}$ is a CSP derived from P by reducing the domain of variable x_i to $\{a\}$). The reason for filtering out action a is that there is no plan of length n giving the preconditions of a (note that the n -th layer is the first layer where action a appeared so the precondition cannot be fully provided by actions before layer $n-1$ because otherwise a

would already appeared at layer $n-1$). Hence action a cannot be used at the $(n+1)$ -th position of any plan.

3.4 Nogood Learning

The last (but not least) improvement we have incorporated is the use of so-called *nogoods*. This well-known technique, often mentioned in connection with dependency-directed backtracking or backjumping, helps the planner to leverage from the failures it has encountered during the search, and uses them to avoid the same failures later, saving thus valuable time the search procedure would have spent otherwise for exploring the same failure again. In order to beware that case, we have to do a single thing: memorize the reason of the failure – memorize a nogood.

Extending our search strategy to support nogood learning is reasonably straightforward since all the necessary information is already provided throughout the search process. As described above, the search starts with the goals to be satisfied and continues backwards by selecting an action A^n that satisfies some of the goals. The preconditions of the selected action are then merged with the unsatisfied goals in order to create new goals for the next step of the search (note that the next step of the search moves to the layer $n-1$). Thus, encountering a failure in fact means that the set of goals at a given layer is unsatisfiable (there is no other action that we could try to apply in order to satisfy the required goals) and that the next time we can avoid trying to satisfy the same set of goals at that layer – a nogood is recorded for future reference as a set of unsatisfiable goals at certain layer.

Moreover, a stronger pruning strategy can be employed when using the nogood records. Not only we do not have to try satisfying the set of goals if it has already been recorded as a nogood, but we can abandon our attempts to satisfy any set of goals that is *more demanding* than any of the nogoods stored for corresponding layer. Formally, set of goals G is more demanding than set of goals H , if $G \subseteq H$. That means a set of goals can be skipped in case it subsumes any of previously recorded nogoods.

Unfortunately, we can use the definition above only in case we use the base search strategy without lifting, i.e., when a set of goals corresponds to a set of state variables that are required to have certain value:

$$\text{Goals} \approx V_1=\text{val}_1 \wedge V_2=\text{val}_2 \wedge \dots \wedge V_k=\text{val}_k. \quad (5)$$

The employment of lifting technique described in section 3.1 requires us to use a different goals representation, where each of state variables can be assigned not only with the certain value, but with any of the values from the set of allowed values for the given variable:

$$\text{Goals} \approx V_1 \in \text{Vals}_1 \wedge V_2 \in \text{Vals}_2 \wedge \dots \wedge V_k \in \text{Vals}_k, \quad (6)$$

where Vals_i represents a set of allowed values for variable V_i .

So we must be careful when defining a *more demanding* set of goals for pruning the search space: Set of goals G is more demanding than set of goals H , if for each state variable $V_i \in H$ also $V_i \in G$ and $\text{Vals}_i(G) \subseteq \text{Vals}_i(H)$. $\text{Vals}_i(X)$ denotes the set of allowed values for variable V_i within the set of goals X (i.e., the domain of that variable).

4 Experimental Comparison

To evaluate the proposed enhancements we implemented them in the *clpfd* library of SICStus Prolog 4.0.5 and compared them using selected planning problems from past International Planning Competitions (STRIPS versions). Namely, we used Gripper, Logistics, Mystery (IPC1), Blocks, Elevator (IPC2), Depots, Zenotravel, DriverLog (IPC3), Airport, PSR (IPC4), and Pipesworld, Rovers, TPP (IPC5). The experiments ran on Intel Xeon CPU E5335 2.0 GHz processor with 8GB RAM under Ubuntu Linux 8.04.2 (Hardy Heron). The reported runtime includes the time to generate the constraint model (prepare the tables) and the time to find the shortest (optimal) plan.

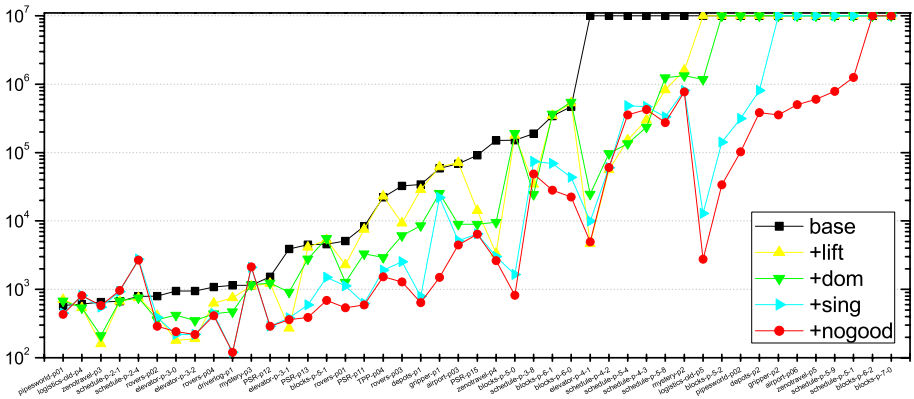


Fig. 3. Comparison of runtimes (logarithmic scale) for selected problems from IPC 1-5

Figure 3 shows the comparison of runtimes (in milliseconds; using 30 min. time limit) to find a shortest plan for all four enhancements when we tried to increasingly combine the proposed methods with the goal to strengthen their power (*+nogood* line denotes the planner which integrates lifting, dom. rules, sing. consistency, and nogoods). We sorted the planning problems increasingly using the runtime of the base model. In comparison with the base model, all enhancements produce some speedup (note that the logarithmic scale is used in the graphs) though it is difficult to exactly state which method brings the biggest improvement.

What we can also notice in the graph is that for some problems, the runtime is worse than for the base model. This is due to the computational complexity of singleton consistency that outweighs the positive effect of search space reduction in these problems. Nevertheless, there is a clear evidence that with the increasing hardness of the problems, the proposed enhancements pay off. Although the graph shows results for only 46 selected problems (due to the picture clarity), in fact we ran our experiments using 86 problems (from easy to semi-difficult instances), out of which the base planner solved 47 problems, while our best version (*+nogood*) solved 63.

5 Conclusions

The paper proposed four enhancements of the constraint models for solving sequential planning problems. The common feature of these enhancements is an attempt to reduce the search space large size of which is a major obstacle when solving planning problems. Our preliminary experiments showed that these enhancements contribute to much better efficiency of planning though their individual contribution is not fully comparable. When all enhancements are used together, they significantly outperform (orders of magnitude) the base model, especially when the problems become hard.

The goal of the proposed constraint models is to provide an efficient framework for sequential planning that can be extended to cover more complex state transitions. In particular, the constraints can describe any relation so we can go beyond the logical formulas and use the arithmetic formulas in the preconditions and effects.

Finally, we would like to acknowledge the support of this research by the Czech Science Foundation under the contracts no. 201/07/0205 and 201/09/H057.

References

1. Barták, R., Toropila, D.: Reformulating Constraint Models for Classical Planning. In: 21st International Florida AI Research Society Conference (FLAIRS 2008), pp. 525–530. AAAI Press, Menlo Park (2008)
2. Barták, R., Toropila, D.: Enhancing Constraint Models for Planning Problems. In: Návrát, P., Chudá, D. (eds.) *Znalosti 2009*, pp. 47–58. Vydavateľstvo Slovenskej technickej univerzity (2009)
3. Lopez, A., Bacchus, F.: Generalizing GraphPlan by Formulating Planning as a CSP. In: 18th International Joint Conference on Artificial Intelligence (IJCAI 2003), pp. 954–960. Morgan Kaufmann, San Francisco (2003)
4. Kautz, H., Selman, B.: Planning as satisfiability. In: 10th European Conference on Artificial Intelligence (ECAI 1992), pp. 359–363 (1992)
5. Bäckström, C., Nebel, B.: Complexity results for SAS+ planning. *Computational Intelligence* 11(4), 625–655 (1995)
6. Helmert, M.: The Fast Downward Planning System. *Journal of Artificial Intelligence Research* 26, 191–246 (2006)
7. Ghallab, M., Nau, D., Traverso, P.: *Automated Planning: Theory and Practice*. Morgan Kaufmann, San Francisco (2004)
8. Reiter, R.: *Knowledge in Action: Logical Foundation for Specifying and Implementing Dynamic Systems*. MIT Press, Cambridge (2001)
9. Long, D., Fox, M.: Plan Permutation Symmetries as a Source of Planner Inefficiency. In: 22nd Workshop of UK Planning and Scheduling Special Interest Group, PlanSIG-22 (2003)
10. Blum, A., Furst, M.: Fast planning through planning graph analysis. *Artificial Intelligence* 90, 281–300 (1997)
11. Vidal, V., Geffner, H.: Branching and Pruning: An Optimal Temporal POCL Planner based on Constraint Programming. In: 19th National Conference on Artificial Intelligence (AAAI 2004), pp. 570–577 (2004)
12. Debruyne, R., Bessière, C.: Some Practicable Filtering Techniques for the Constraint Satisfaction Problem. In: 15th International Joint Conference on Artificial Intelligence (IJCAI), pp. 412–417. Morgan Kaufmann, San Francisco (1997)

Interval-Valued Fuzzy Formal Concept Analysis

Yassine Djouadi^{1,2} and Henri Prade¹

¹ IRIT, Université Paul Sabatier,
118 Route de Narbonne, 31062 Toulouse Cedex 09, France
djouadi@irit.fr, prade@irit.fr

² University of Tizi-Ouzou, BP 17, RP, Tizi-Ouzou, Algeria

Abstract. Fuzzy formal concept analysis is concerned with formal contexts expressing scalar-valued fuzzy relationships between objects and their properties. Existing fuzzy approaches assume that the relationship between a given object and a given property is a matter of degree in a scale L (generally $[0,1]$). However, the extent to which “object o has property a ” may be sometimes hard to assess precisely. Then it is convenient to use a sub-interval from the scale L rather than a precise value. Such formal contexts naturally lead to interval-valued formal concepts. The aim of the paper is twofold. We provide a sound minimal set of requirements for interval-valued implications in order to fulfill the fuzzy closure properties of the resulting Galois connection. Secondly, a new approach based on a generalization of Gödel implication is proposed for building the complete lattice of all interval-valued formal concepts.

1 Introduction

Formal concept analysis [1] (FCA for short) deals with a particular kind of analysis of data based on an object-property relationship called formal context. In the classical setting, a formal context consists of a crisp binary relation \mathcal{R} between a set of objects and a set of properties. This relation is usually represented as a table with rows corresponding to objects, columns corresponding to properties (or conversely). From a formal context, one can construct $(\{\text{objects}\}, \{\text{properties}\})$ pairs, representing the extension and the intension of a formal concept.

Human knowledge often uses gradual properties. Then, the relationship $\mathcal{R}(o, a)$ becomes a matter of degree. Fuzzy set theory has been already considered to deal with such a case. Since the first paper from Burusco et al. [2], many approaches have been proposed: Pollandt [3], Belohlavek [4], Ayouni et al. in [5], etc. The reader is referred to [6] for a complete survey. The common starting point of these approaches is the notion of a fuzzy formal context (L -fuzzy context). In this case, each entry $\mathcal{R}(o, a)$ of a fuzzy formal context is assigned a truth value from the scale $L=[0,1]$ which estimates to what extent object o has property a .

In practice, the elicitation of such degrees is not always easy. It may be more comfortable to allow the use of intervals, especially when the evaluation of the degree is subject to some variation, or is partially known. For example, it may be difficult, for a teacher, to assess the level of a student for whom he has only

incomplete information. Then, it is natural to use an interval for assessing this level. This observation motivates the study of fuzzy FCA where the contexts are associated with intervals rather than precise scalar values. It results in interval-valued fuzzy formal contexts (for short IVFF contexts). To our knowledge, the only existing work on IVFF context is due to Burusco et al [7]. They also propose in [8] an application in order to predict missing information in IVFF contexts.

The development of an interval-valued fuzzy FCA theory raises two questions: the definition of a proper fuzzy closure operator and the related issue of the choice of an appropriate interval-valued implication connective.

For this purpose, the proposed approach characterizes a minimal requirement set of algebraic properties for preserving the fuzzy closure properties in the interval-valued setting, such closure properties being necessary for defining formal concepts on the basis of a Galois connection. Moreover, this paper more particularly studies the use of an interval-valued extension of Gödel implication, because of its graduality semantics as highlighted in [9].

The paper is organized as follows. Section 2 advocates the need for generalizing crisp and fuzzy formal contexts. Section 3 gives the mathematical background on FCA and especially discusses the problem of choosing an implication operator underlying the definition of the Galois connection. After a brief presentation of interval-valued fuzzy sets (for short IVFS) theory given in section 4, families of IVFS implication operators are characterized w.r.t. the closure properties. In the next section, an extension of Gödel implication is proposed whose closure satisfaction is proved. Properties of the concepts lattice are established. They will be used in the construction algorithm for building this lattice efficiently.

2 Why Interval-Valued Fuzzy Formal Contexts

In the classical FCA setting, formal contexts are given under the form of a table that formalizes the relationship between objects and properties. A table entry indicates whether an object satisfies the property (this is usually denoted by a cross mark), or not (it is often indicated by the absence of mark). For instance, Table 1 in its left part presents a binary relation that indicates whether a given student sufficiently masters or not given courses in *English*, *Law*, and *Mathematics* (respectively abbreviated in *eng*, *law*, and *mat*).

It may occur that one cannot state whether a property is satisfied or not by an object. For example, $Masters(Sophie, law)$ is represented by a question mark that is, e.g. *Sophie* was always absent for the test in law. Such incompletely informed contexts are referred to as incomplete contexts [10]. A possibilistic interpretation of these contexts is also proposed in [11].

It is worth pointing out that the fuzzy extension of binary formal contexts may in fact convey different semantics. In a first interpretation, the values in the table (which are scalars in L) should be understood as providing a refinement of the crosses. Namely, they represent to which extent an object has a property, while in the classical model, this relationship was not a matter of degree. Note that in this view, we do not refine the absence of a property for an object (the

Table 1. Left: binary incomplete formal context. Right: fuzzy formal context.

Binary	<i>eng</i>	<i>law</i>	<i>mat</i>	Fuzzy	<i>eng</i>	<i>law</i>	<i>mat</i>
<i>Peter</i>	×	×		<i>Peter</i>	0.9	0.7	0.2
<i>Sophie</i>	×	?	×	<i>Sophie</i>	0.8	?	0.5
<i>Michael</i>		×	×	<i>Michael</i>	0.3	1	0.8
<i>Nahla</i>		×		<i>Nahla</i>	0.4	0.6	0.1

blank is always replaced by the bottom element 0 of L). This view will be referred to as the *unipolar scale interpretation* in the following. Another fuzzification of the table, maybe more in the standard spirit of fuzzy logic would be to replace both the *cross* marks and the *blank* marks by values in the scale L ($L = [0, 1]$), where the mid-point of L is the pivoting value between the situations where the object has more the property than it does not, and the converse situation where having the negation of the property is more prominent. This means that in this view, a fuzzy formal concept should be learnt together with its negation. This view corresponds to a *bipolar scale interpretation*. In this paper, we privilege the *unipolar scale interpretation*, which fits with the graduality of the increasing intensity of the satisfaction of a property. Moreover this gradual interpretation is the one currently used for encoding many-valued attribute contexts in fuzzy FCA (e.g. [12]). For example, in the right part of Table 1, Peter masters English at 90%. It is also worth noticing that L -fuzzy contexts have no way to represent situations of total ignorance (e.g. (*Sophie*, *law*)).

It is not always easy to precisely assess the degree to which an object satisfies a gradual property as advocated in the introduction. Table 2 gives an example of an IVFF context where some entries are sub-intervals of L . In this example, entries are understood as: Peter masters Law at least at 50% and at most at 70%. An interesting feature of IVFF contexts is that they provide the ability to encode total ignorance. For instance, Sophie’s mastering level in law is now encoded by the interval $[0,1]$.

Table 2. Example of an interval-valued fuzzy formal context

Interval-Valued	<i>eng</i>	<i>law</i>	<i>mat</i>
<i>Peter</i>	[0.9,1]	[0.5,0.7]	[0.0,0.2]
<i>Sophie</i>	[0.8,1]	[0,1]	[0.5,0.5]
<i>Michael</i>	[0.3,0.6]	[1,1]	[0.8,0.8]
<i>Nahla</i>	[0.2,0.4]	[0.6,1]	[0.0,0.1]

3 Formal Concept Analysis: Background

Formal concept analysis has been introduced by Wille and Ganter [1, 13]. Formally, let \mathcal{O} be a set of objects ($\mathcal{O} = \{o_1, o_2, \dots, o_n\}$) and \mathcal{A} be a set of properties ($\mathcal{A} = \{a_1, a_2, \dots, a_m\}$). A binary formal context is a triple $\mathcal{K} := (\mathcal{O}, \mathcal{A}, \mathcal{R})$ which

is completely defined by a crisp binary relation \mathcal{R} ($\mathcal{R} \subseteq \mathcal{O} \times \mathcal{A}$). We shall denote in the rest of the paper, by uppercases A, B, \dots (resp. X, Y, \dots), the subsets of \mathcal{A} (resp. \mathcal{O}). For a set of objects X we define the set X^* of properties that are satisfied by all objects in X . Similarly, for a set of properties A , we define the set A^* of objects that satisfy all properties in A as follows:

$$X^* = \{a \mid \forall o \in X, (o, a) \in \mathcal{R}\}, \quad A^* = \{o \mid \forall a \in A, (o, a) \in \mathcal{R}\} \quad (1)$$

A *formal concept* is a pair $\langle X, A \rangle$ where X (called extent-concept) and A (called intent-concept) are such that $X^* = A$ and $A^* = X$. The set $\mathcal{C}(\mathcal{K})$ of all formal concepts of \mathcal{K} , equipped with a partial order (denoted \preceq) defined as: $\langle X_1, A_1 \rangle \preceq \langle X_2, A_2 \rangle$ iff $X_1 \subseteq X_2$ (or, equivalently, $A_2 \subseteq A_1$), forms a complete lattice, called the *concept lattice* of \mathcal{K} . The stars in the expressions of X^* and A^* denote the so-called Galois derivation operators. An interesting feature of Galois connection is that a “closure” property of the relation is implicitly conveyed.

3.1 Fuzzy Settings

Fuzzy Galois derivation operators originally proposed in [2] are generally based on an implication operator (denoted I), since Expression 1 expresses inclusions. Their definition is given as:

$$X^*(a) = \bigwedge_{o \in \mathcal{O}} \left(I(X(o), \mathcal{R}(o, a)) \right), \quad A^*(o) = \bigwedge_{a \in \mathcal{A}} \left(\mathcal{I}(A(a), \mathcal{R}(o, a)) \right) \quad (2)$$

Fuzzy FCA poses the problem of the satisfaction of the closure properties [6] defined below. Depending on the choice of an implication operator these properties may be satisfied or not.

Definition 1. A fuzzy closure operator on a set \mathcal{U} is a mapping $\xi : L^{\mathcal{U}} \rightarrow L^{\mathcal{U}}$ satisfying $\forall U, V \in L^{\mathcal{U}}$:

- (CL1): $U \subseteq \xi(U)$
- (CL2): $U \subseteq V \implies \xi(U) \subseteq \xi(V)$
- (CL3): $\xi(\xi(U)) = \xi(U)$

where \subseteq denotes the standard fuzzy set inclusion (defined by the pointwise inequality of the membership functions).

3.2 Choice of the Implication

Burusco et al. [2] use a strong implication of the form $\neg p \vee q$, while Belohlavek [4] proposed to use a residuated implication in order to ensure fuzzy closure properties. Residuated implications are such that $I(p \rightarrow q) = 1$ iff $p \leq q$. In this paper, we favor the use of a particular implication, i.e. the Gödel implication defined by $I(p \rightarrow q) = q$ if $p > q$. Indeed, this implication is qualitative in nature, which agrees with the fact that the values in formal contexts are also qualitatively assessed. Moreover, it is well-known that Gödel implication encodes gradual rules [9], and is generally speaking in the spirit of the gradual interpretation discussed in Section 2, since the implication increases with q until q reaches the threshold p .

4 Interval-Valued Fuzzy Galois Connections

This section first presents interval-valued fuzzy sets, and details the choice of an implication operator in this setting before discussing closure operators.

An IVFS U in a universe \mathcal{U} is defined by $U = \{ \langle u, \underline{\mu}_U(u), \overline{\mu}_U(u) \rangle, u \in \mathcal{U} \}$ where $\underline{\mu}_U, \overline{\mu}_U : \mathcal{U} \rightarrow L$ are the lower and the upper membership functions s.t. $0 \leq \underline{\mu}_U(u) \leq \overline{\mu}_U(u) \leq 1$. Naturally, if $\forall u \in \mathcal{U}, \underline{\mu}_U(u) = \overline{\mu}_U(u)$ then, U is an ordinary fuzzy set. Let $\mathfrak{L}[L]$ denote the set of all closed intervals in L defined as: $\mathfrak{L}[L] = \{ x \mid x = [\beta_1, \beta_2], \beta_1, \beta_2 \in [0, 1]^2 \text{ and } \beta_1 \leq \beta_2 \}$. The set of all IVFS on \mathcal{U} can be considered as $\mathfrak{L}[L]$ -fuzzy sets (in the sense of Goguen [14]) according to the lattice $\mathfrak{J} = (\mathfrak{L}[L], \preceq_{\mathfrak{J}})$ where the smallest element is $0_{\mathfrak{J}} = [0, 0]$ and the largest element is $1_{\mathfrak{J}} = [1, 1]$. A partial order relation $\preceq_{\mathfrak{J}}$ is defined as:

$$[\beta_1, \beta_2] \preceq_{\mathfrak{J}} [\gamma_1, \gamma_2] \iff \beta_1 \leq \gamma_1 \text{ and } \beta_2 \leq \gamma_2 \tag{3}$$

while the meet (\wedge) and join (\vee) operators are defined for each pair $([\beta_1, \beta_2], [\gamma_1, \gamma_2])$, as:

$$[\beta_1, \beta_2] \wedge [\gamma_1, \gamma_2] = [\min(\beta_1, \gamma_1), \min(\beta_2, \gamma_2)] \tag{4}$$

$$[\beta_1, \beta_2] \vee [\gamma_1, \gamma_2] = [\max(\beta_1, \gamma_1), \max(\beta_2, \gamma_2)] \tag{5}$$

The definition of logical connectives in the framework of interval-valued fuzzy logic arises quite naturally from their counterpart in L -fuzzy logic. Implication operators are thus generalized [15, 16] and defined as follows.

Definition 2. *An interval-valued fuzzy implication is any mapping $\mathcal{I} : \mathfrak{L}[L] \times \mathfrak{L}[L] \rightarrow \mathfrak{L}[L]$ satisfying:*

- (I1) : $\mathcal{I}(x, y) \preceq_{\mathfrak{J}} \mathcal{I}(x', y)$ for $x' \preceq_{\mathfrak{J}} x$
- (I2) : $\mathcal{I}(x, y) \preceq_{\mathfrak{J}} \mathcal{I}(x, y')$ for $y \preceq_{\mathfrak{J}} y'$
- (I3) : $\mathcal{I}(0_{\mathfrak{J}}, 0_{\mathfrak{J}}) = \mathcal{I}(0_{\mathfrak{J}}, 1_{\mathfrak{J}}) = \mathcal{I}(1_{\mathfrak{J}}, 1_{\mathfrak{J}}) = 1_{\mathfrak{J}}$
- (I4) : $\mathcal{I}(1_{\mathfrak{J}}, 0_{\mathfrak{J}}) = 0_{\mathfrak{J}}$

An IVFF context, denoted also $\mathfrak{L}[L]$ -fuzzy context, is a tuple $\mathfrak{K} = (\mathfrak{L}[L], \mathcal{O}, \mathcal{A}, \mathfrak{R})$ where the interval-valued fuzzy relation \mathfrak{R} is defined as $\mathfrak{R} : \mathcal{O} \times \mathcal{A} \rightarrow \mathfrak{L}[L]$. Let us recall that formal concepts are dual pairs of closed subsets. In order to induce all formal concepts related to IVFF contexts, we have to define fuzzy Galois operators that satisfy the closure property w.r.t. Definition 1 now equipped with the interval-valued inclusion: $U_1 \subseteq_{\mathfrak{J}} U_2 \iff \forall u \in \mathcal{U}, U_1(u) \preceq_{\mathfrak{J}} U_2(u)$.

We may think that a $\mathfrak{L}[L]$ -residuated algebra can be obtained as a straightforward extension of L -residuated algebra [4] for which many properties continue to hold, and among them the closure properties. However, such extended algebra restricts the class of permitted implication operators. For this purpose, we propose to consider a minimal requirement set under which the closure property of the Galois operators is satisfied, as guaranteed by the following theorem.

Definition 3. *A $\mathfrak{L}[L]$ -fuzzy set $X \in \mathfrak{L}[L]^{\mathcal{O}}$ (dually, $A \in \mathfrak{L}[L]^{\mathcal{A}}$) is a closed $\mathfrak{L}[L]$ -fuzzy set if it is equal to its closure: $X = X^{**}$ (dually, $A = A^{**}$).*

Theorem 1. *The Galois derivation operator $(.)^{**}$ defined on $\mathfrak{L}[L]^{\mathcal{O}}$, dually on $\mathfrak{L}[L]^{\mathcal{A}}$, is a closure operator if the property $x \preceq_{\mathfrak{J}} \mathcal{I}(\mathcal{I}(x, y), y)$ is satisfied $\forall x, y \in \mathfrak{L}[L]$.*

Proofs are omitted in this paper due to the lack of space.

Following Birkhoff [17], it results that the sets of extent-concepts X and the set of intent-concepts A , verifying $X^* = A$ and $A^* = X$, are dually isomorphic complete lattices w.r.t the relation $\subseteq_{\mathfrak{J}}$. The set $\mathfrak{C}(\mathfrak{K})$ of all formal concepts equipped with a partial order in the sense of the inclusion relation $\subseteq_{\mathfrak{J}}$ is also a complete lattice denoted $\mathfrak{B}(\mathfrak{K})$.

5 Concept Lattice Based on Extended Gödel Implication

For fuzzy contexts, derivation operators depend on the choice of a fuzzy implication operator. Like for L -fuzzy implication, different classes of interval-valued fuzzy implication operators exist in the literature [16, 18]. Let us recall that IVFF contexts considered in this paper are concerned with graduality semantics. For the reasons already stated in Section 3.3, Gödel implication is thus a natural candidate for the extension to interval-valued (IV for short) setting.

5.1 Interval-Valued Gödel Implication

The existing proposals [16, 18] for IV fuzzy extension of Gödel implication operators amount to adapt the definition of residuation to interval-valued structures, and for this purpose to choose an IV fuzzy extension of conjunction operators, and are not much motivated by interpretative issues. With in mind the unipolar scale interpretation semantics, and the idea of imprecise information about the fulfilment degrees of relationship in IVFF contexts, provided that the implication satisfies minimal requirements s.t. the fuzzy closure properties hold, the extension of Gödel implication to $\mathfrak{L}[L]$ -fuzzy setting arises quite naturally from its counterpart in classical L -fuzzy setting and is given as:

Definition 4. *The extension of the Gödel implication to the interval-valued setting is defined as:*

$$\mathcal{I}_{\mathcal{G}}(x, y) = \begin{cases} 1_{\mathfrak{J}} & \text{if } x \preceq_{\mathfrak{J}} y \\ y & \text{otherwise} \end{cases} \quad (6)$$

It can be easily noticed that the above expression reduces to Gödel implication when intervals reduce to scalar values. Moreover, if we consider the IV counterpart of Expression 2, it sounds natural to consider that the IVFS of properties possessed by objects in X is defined by an imprecise degree y that reflects the degrees appearing in the context table for these objects, when X is an ordinary subset (since $0_{\mathfrak{J}} \preceq_{\mathfrak{J}} y \preceq_{\mathfrak{J}} 1_{\mathfrak{J}}$). When iterating (2), X (resp. A) is replaced by A^* (resp. X^*) which is an IVFS. Then, (6) yields $1_{\mathfrak{J}}$ as soon as y exceeds the now imprecise threshold x both in terms of lower and upper bounds (in agreement with (3)).

According to Theorem 1, we have first to verify that the above definition satisfies the closure properties. Given $x, y \in \mathcal{L}[L]$ (s.t. $x = [x_1, x_2], y = [y_1, y_2]$) four different situations may be considered depending on whether $[x_1, x_2] \preceq_{\mathcal{I}} [y_1, y_2]$ or not. It is also important to point out that the relation $\preceq_{\mathcal{I}}$ is a partial order and in situations #2 and #4 no order may be established between $[x_1, x_2]$ and $[y_1, y_2]$. The following table states that for each situation, the closure property holds, namely $x \preceq_{\mathcal{I}} \mathcal{I}_G(\mathcal{I}_G(x, y), y)$.

#	Ranking	$\mathcal{I}_G(\mathcal{I}_G(x, y), y)$	$x \preceq_{\mathcal{I}} \mathcal{I}_G(\mathcal{I}_G(x, y), y)$
1	$x_1 \leq y_1$ and $x_2 \leq y_2$	$[y_1, y_2]$	yes
2	$x_1 \leq y_1$ and $y_2 < x_2$	$[1, 1]$	yes
3	$y_1 < x_1$ and $y_2 < x_2$	$[1, 1]$	yes
4	$y_1 < x_1$ and $x_2 \leq y_2$	$[1, 1]$	yes

5.2 Concept Lattice Characterization

There are few works related to L -fuzzy concepts lattice construction (namely [19] and [20]). A $\mathcal{L}[L]$ -fuzzy extension of the concepts lattice construction process will be of a significantly higher time complexity than their L -fuzzy counterpart. A straightforward extension becomes then inappropriate. In order to reduce the complexity, we provide some characterization of the $\mathcal{L}[L]$ -fuzzy concepts lattice. Propositions 1 and 2 give the upper and lower bounds of the $\mathcal{L}[L]$ -lattice which are used to start and stop the construction algorithm.

Definition 5. A $\mathcal{L}[L]$ -fuzzy set G is called generator of an intent-concept A if $G^{**} = A$ and it is said minimal if $\nexists G'$ s.t. $G' \subsetneq G$ and $(G')^{**} = A$.

Proposition 1. Each element U (intent-concept or extent-concept) of the lattice $\mathfrak{B}(\mathfrak{R})$ is bounded:

$$\tilde{0}_3^{**}(u) \preceq_{\mathcal{I}} U(u) \preceq_{\mathcal{I}} \tilde{1}_3^{**}(u)$$

where $\forall u \tilde{0}_3^{**}(u) = 0_3^{**}$ (dually $\tilde{1}_3^{**}(u) = 1_3^{**}$)

Let A^\wedge denote the $\mathcal{L}[L]$ -fuzzy set on \mathcal{A} whose membership corresponds to the smallest element in each column of the $\mathcal{L}[L]$ -formal context defined as $A^\wedge(a_j) = \bigwedge_{o \in \mathcal{O}} \mathfrak{R}(o, a_j)$. The bottom element of the lattice is given by:

Proposition 2. The infimum element, denoted S_A , of the intent-concept lattice is given as:

$$S_A = A^\wedge$$

Each intent-concept owns a set of generators. Determining this set for each intent-concept, unlike existing approaches (e.g. [20], [7]), may be of real interest for data mining applications, like generating fuzzy association rules. The following lemma is used to initialize, in the construction algorithm, the set of generators and further the set of potential generators. Whereas, the lemma 2 characterizes the lower bound of this set, knowing that the upper bound is the closure itself.

Lemma 1. Let the set A_k^\wedge denotes a $\mathfrak{L}[L]$ -fuzzy set on \mathcal{A} defined as:

$$A_k^\wedge(a_j) = \begin{cases} \bigwedge_{o_i \in \mathcal{O}} \mathfrak{R}(o_i, a_j) & \text{if } j = k \\ 0_{\mathfrak{J}} & \text{otherwise} \end{cases}$$

Then $A_k^\wedge, k = 1, \dots, m$, is a generator of the fuzzy intent-concept S_A .

Lemma 2. Let A and B two $\mathfrak{L}[L]$ -fuzzy sets such that $A \subseteq_{\mathfrak{J}} B$. If $B \subseteq_{\mathfrak{J}} A^{**}$ then $B^{**} = A^{**}$.

5.3 Construction Algorithm

The proposed algorithm assumes a discretization of the scale L which obviously implies a discretization of the lattice $\mathfrak{L}[L]$ and makes easier the understanding of the algorithm. Typically, we take $L = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. The function *Compute_Closure*(g) is used to compute the closure and to store the generators. The following notations and functions are also used:

- ICS*: Intent Concept Set, containing all intent concepts.
- GS*(c): Generators Set, containing all generators of the intent concept c .
- DOM*(a_j): returns the domain of the attribute a_j which is a subset of $\mathfrak{L}[L]$.
- g*(a_j): returns the membership degree of the element a_j for a given IVFS g .
- SUCC*($g(a_j)$): returns all upper bounds of a_j , such that $SUCC(g(a_j)) = \{y \in DOM(a_j) \mid g(a_j) \prec_{\mathfrak{J}} y \text{ and } \nexists y' \text{ s.t. } g(a_j) \prec_{\mathfrak{J}} y' \prec_{\mathfrak{J}} y\}$.
- JOIN*(U): returns the smallest upper bound of a given set U ($U \subseteq \mathfrak{L}[L]$).

Algorithm. *Compute_Closure*(g)

Begin

- 1: **If** $g^{**} \in ICS$
- 2: **Then** $G(g^{**}) \leftarrow G(g^{**}) \cup \{g\}$
- 3: **Else** $ICS \leftarrow ICS \cup \{g^{**}\};$
- 4: $G(g^{**}) \leftarrow g;$

5: **End If**

6: return $g^{**};$

End

The algorithm *GEN_INTENT_CONCEPTS* takes as input an IVFF context and returns the set *ICS* of all intent concepts. Extents concepts are obtained using the IV counterpart of Expression 2.

For example, $(\langle eng, .3, .6 \rangle, \langle law, 1, 1 \rangle, \langle mat, .8, .8 \rangle)$ is an intent concept and its generator is $(\langle eng, .2, .4 \rangle, \langle law, 1, 1 \rangle, \langle mat, .8, .8 \rangle)$. Typically, the smallest intent concept is $(\langle eng, .2, .4 \rangle, \langle law, .0, .7 \rangle, \langle mat, .0, .1 \rangle)$.

Algorithm. GEN_INTENT_CONCEPTS

Begin

```

1:  $ICS \leftarrow \{A^\wedge\};$  /* Initialization of  $ICS$  with the singleton  $A^\wedge$  */
2:  $G(A^\wedge) \leftarrow \{A_k^\wedge, k = 1..m\};$  /* Initializing generators of  $A^\wedge$  */
3:  $j \leftarrow 1; g \leftarrow A^\wedge;$ 
4: While  $(j \leq m)$  and  $(g \preceq_{\mathcal{I}} 1_{\mathcal{I}}^{**})$  Do
5: Begin
6: While  $g(a_j) \in DOM(a_j)$  Do
7: Begin
8: If  $SUCC(g(a_j))$  is a singleton
9: Then
10:  $g(a_j) \leftarrow SUCC(g(a_j));$  /* $g(a_k)$  s.t.  $k \neq j$  remains unchanged*/
11:  $g \leftarrow Compute\_Closure(g);$ 
12: else
13: For each  $e \in SUCC(g(a_j))$  Do
14:  $g(a_j) \leftarrow e;$ 
15:  $g \leftarrow Compute\_Closure(g);$ 
16: End For
17:  $g \leftarrow JOIN(SUCC(g(a_j)));$ 
18: End If
19: End While
20:  $j \leftarrow j + 1;$ 
21: End While
End

```

6 Conclusion

This paper extends the framework of fuzzy formal concept analysis by allowing interval values degrees to be used. This framework provides the ability to handle incomplete and partial information including missing values which may occur in real world formal contexts. We have also provided a minimal requirement property under which, the fuzzy closure property of Galois derivation operators holds. Beyond the construction of the concepts lattice, we have provided a way to determine the associated generators of each intent-concept. This may be useful for practical data-mining applications like fuzzy association rules generation. Many issues remain to explore. Are there other implications operators of interest for FCA setting? Some particular cases of IVFF contexts may be worth studying such as intervals of the form $[\alpha, 1]$, or the restriction of the context to disjoint intervals. Besides, in case of many different intervals appearing in the context, we may think of replacing some of them by slightly larger intervals covering them (making thus the information a bit less precise), in order to diminish the

number of distinct intervals of the context. Then what will be the impact on the set of existing formal concepts?

References

- [1] Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) *Ordered Sets*, pp. 445–470. Reidel, Dordrecht (1982)
- [2] Burusco, A., Fuentes-Gonzalez, R.: The study of the L-fuzzy concept lattice. *Mathware & Soft Computing* 3, 209–218 (1994)
- [3] Pollandt, S.: *Fuzzy begriffe*. Springer, Heidelberg (1997)
- [4] Belohlávek, R.: Fuzzy Galois connections. *Math. Logic Quart* 45, 497–504 (1999)
- [5] Ayouni, S., BenYahia, S.: Extracting compact and information lossless set of fuzzy association rules. In: *IEEE, Inter. Conf. on Fuzzy Systems*, London, pp. 1444–1449 (2007)
- [6] Belohlávek, R., Vychodil, V.: What is a fuzzy concept lattice. In: *Proc. CLA 2005*, Olomouc, Czech Republic, pp. 34–45 (2005)
- [7] Burusco, A., Fuentes-Gonzalez, R.: The study of the interval-valued contexts. *Fuzzy Sets and Systems* 121(3), 439–452 (2001)
- [8] Alcade, C., Burusco, A., Fuentes-Gonzalez, R., Zubia, I.: Treatment of L-fuzzy contexts with absent values. *Information Sciences* 179(2), 1–15 (2009)
- [9] Dubois, D., Prade, H.: Gradual inference rules in approximate reasoning. *Information Sciences* 61, 103–122 (1992)
- [10] Burmeister, P., Holzer, R.: Treating incomplete knowledge in formal concepts analysis. In: Ganter, B., Stumme, G., Wille, R. (eds.) *Formal Concept Analysis. LNCS (LNAI)*, vol. 3626, pp. 114–126. Springer, Heidelberg (2005)
- [11] Dubois, D., Dupin de Saint Cyr, F., Prade, H.: A possibility-theoretic view of formal concept analysis. *Fundamenta Informaticae* 75(1-4), 195–213 (2007)
- [12] Messai, N., Devignes, M.D., Napoli, A., Tabbone, M.S.: Many-valued concept lattices for conceptual clustering and information retrieval. In: *ECAI 2008*, 18th European Conference on Artificial Intelligence, Patras, Greece, pp. 722–727 (2008)
- [13] Ganter, B., Wille, R.: *Formal Concept Analysis*. Springer, Heidelberg (1999)
- [14] Goguen, J.A.: L-fuzzy sets. *J. Math. Anal. Appl.* 18, 145–174 (1967)
- [15] Van Gasse, B., Cornelis, C., Deschrijver, G., Kerre, E.E.: Triangle algebras: A formal logic approach to interval-valued residuated lattices. *Fuzzy Sets and Systems* 159(9), 1042–1060 (2008)
- [16] Alcade, C., Burusco, A., Fuentes-Gonzalez, R.: A constructive method for the definition of interval-valued fuzzy implication operators. *Fuzzy Sets and Systems* 153(2), 211–227 (2005)
- [17] Birkhoff, G.: *Théorie et applications des treillis*. *Annales de l’IHP* 11(5), 227–240 (1949)
- [18] Cornelis, C., Deschrijver, G., Kerre, E.E.: Implication in intuitionistic fuzzy and interval-valued fuzzy set theory: construction, classification, application. *International Journal of Approximate Reasoning* 35, 55–95 (2004)
- [19] Belohlávek, R., Vychodil, V.: Graded LinClosure and its role in relational data analysis. In: Yahia, S.B., Nguifo, E.M., Belohlavek, R. (eds.) *CLA 2006. LNCS (LNAI)*, vol. 4923, pp. 139–154. Springer, Heidelberg (2008)
- [20] Xie, C., Yi, L., Du, Y.: An algorithm for fuzzy concept lattices building with application to social navigation. In: *ISKE 2007, International Conference on Intelligent Systems and Knowledge Engineering*, China (2007)

Application of Meta Sets to Character Recognition

Bartłomiej Starosta*

Polish-Japanese Institute of Information Technology,
ul. Koszykowa 86, 02-008 Warsaw, Poland
Tel.: +48-22-5844528
barstar@pjwstk.edu.pl
<http://www.pjwstk.edu.pl/~barstar>

Abstract. A new approach to character recognition problem, based on meta sets, is introduced and developed. For the given compound character pattern consisting of a number of character samples accompanied by their corresponding quality degrees, and for the given testing character sample, the main theorem of the paper gives means to evaluate the correlation between the testing sample and the compound pattern. It also enables calculation of similarity degrees of the testing sample to each pattern element. The quality degrees and the correlation are expressed by means of membership degrees of meta sets representing samples in the meta set representing the compound pattern. The similarity degrees are expressed as equality degrees of these meta sets.

The meta set theory is a new alternative to the fuzzy set theory. By the construction of its fundamental notions it is directed to efficient computer implementations. This paper presents an example of application of the theory to a real-life problem.

Keywords: Meta set, character recognition, quality degree, fuzzy set.

1 Motivations

The theory of meta sets is a new set theory with non-binary (“fuzzy”) membership relation. It uses a language similar to the language of the classical set theory ([1]). However, there is an infinite, countable number of membership, non-membership, equality and inequality relational symbols to enable expressing various degrees of satisfaction of the relations. It is worth noting, that algebraic operations for meta sets satisfy Boolean algebra axioms ([3]). The meta set theory is meant to be an alternative to the fuzzy set theory ([2]). Although it is better fitted within the classical set theory than the fuzzy set theory – in particular, elements of meta sets are also meta sets – it was designed so as to enable efficient computer (or even hardware) implementations. For the detailed treatment of the idea of meta set the reader is referred to [3] and [4]. This paper introduces only the concepts directly relevant to character recognition.

* Corresponding author.

The theory is new and under development. We demonstrate here the first example of its application to a real-life problem. We stress, that our main goal is to manifest the fact that the concept of meta set properly describes “fuzzy” relations and is applicable to real problems. To clarify the presentation we concentrate on a very particular, simplified case. Further in this section we explain the general idea of our approach. Section 2 introduces basic definitions and proves the main theorem. Section 3 reveals the idea of application of meta sets to character recognition.

The abstract concepts presented here are practically tested by means of an experimental computer program based on the implementation of meta sets operations. The program enables defining character patterns, supplying testing samples and evaluating similarity degrees. The results seem to be consistent with human intuition with respect to similarity of characters or letters.

1.1 The General Idea

Let us consider a number of different (possibly hand-written) samples of some letter, e.g. 'L', and let us denote them with the symbols $\pi^1, \pi^2, \dots, \pi^n$. Further, let us assign to each sample π^i a quality degree P^i which measures how close is the sample to the ideal. The quality degrees are to be supplied by an expert or a user. They represent his or her point of view on the shape of the letter. If we manage to represent each sample π^i as a meta set, 1 and each degree P^i as a set of nodes of the binary tree, 2 then the set $\pi = \bigcup_{i=1}^{i=n} \{ \pi^i \} \times P^i$ may be treated as another meta set. This meta set expresses our opinion on how the letter 'L' should look like, based on the compound pattern comprised of a number of estimated samples. We will match new samples of the letter against this pattern to measure their quality degree or similarity to the pattern π .

Let us then supply another sample of the letter 'L' and let us represent it as a meta set σ , similarly to the pattern elements π^i . We may ask what is the membership degree of the sample σ to the pattern π . The Theorem 1 gives the answer to this question. It also allows for evaluation of equality degrees of σ to each π^i , which express similarity of the supplied sample to each pattern element. Since these degrees are meant to express character resemblance, then the higher the membership (equality) degree – the greater resemblance of the sample and the pattern (pattern element).

2 The Theory of Meta Sets

We now establish some basic terms and notation. Then we define fundamental meta set theory concepts and prove their most important properties relevant to character recognition. This section ends with the main theorem which is applied in the next section.

¹ See Sect. 3.

² See Definition 1.

2.1 Fundamental Definitions

We use the symbol \mathbb{T} for the infinite binary tree with the root $\mathbb{1}$ which is its largest element. The nodes of the tree \mathbb{T} might be considered as finite binary sequences, the root $\mathbb{1}$ being the empty sequence. A branch in the tree \mathbb{T} is a maximal chain (a maximal set of pairwise comparable nodes). It might be represented as an infinite binary sequence. The n -th level of the tree \mathbb{T} , denoted by \mathbb{T}_n , is the set of all binary sequences of the same length n , for instance $\mathbb{T}_1 = \{0, 1\}$, $\mathbb{T}_2 = \{00, 01, 10, 11\}$, whereas $\mathbb{T}_0 = \{\mathbb{1}\}$.

We now define the fundamental notion of meta set. A meta set might be perceived as a crisp set whose elements are accompanied by sets of nodes of the binary tree. These sets of nodes express the membership degrees of elements to the set. Since elements are also meta sets, then their elements are also accompanied by nodes of \mathbb{T} , and so on, recursively. The recursion stops at the empty set \emptyset by the Axiom of Foundation in the Zermelo-Fraenkel set theory ([\[1\]](#)).

Definition 1. *A meta set is a crisp set which is either the empty set \emptyset or which has the form:*

$$\tau = \{ \langle \sigma, p \rangle : \sigma \text{ is a meta set, } p \in \mathbb{T} \} .$$

Here $\langle \cdot, \cdot \rangle$ denotes an ordered pair.

Thus, from the classical set theory point of view, a meta set is a relation between a crisp set of other meta sets and a set of nodes of the tree \mathbb{T} . The nodes measure the membership degree of an element to the set in such way, that greater nodes (in the tree ordering) represent higher membership degrees; the root $\mathbb{1}$ designates the full, classical membership. For two comparable nodes $q > p$, the greater one q supplies more membership information and therefore, the smaller one p does not influence the overall membership degree of an element to the meta set. On the other hand, if p and q are incomparable, then they both independently contribute to the membership degree (cf. Lemmas 1 and 2 in [\[4\]](#)).

A meta set may also be perceived as a "fuzzy" family of crisp sets thanks to the interpretation technique introduced by the following definition. Each member of the family represents some specific, particular point of view on the meta set.

Definition 2. *Let τ be a meta set and let \mathcal{C} be a branch in the binary tree \mathbb{T} . The interpretation of the meta set τ , given by the branch \mathcal{C} , is the following crisp set:*

$$\tau_{\mathcal{C}} = \{ \sigma_{\mathcal{C}} : \langle \sigma, p \rangle \in \tau \wedge p \in \mathcal{C} \} .$$

Thus, branches in \mathbb{T} allow for producing crisp sets out of the meta set. The family $\{ \tau_{\mathcal{C}} : \mathcal{C} \text{ is a branch in } \mathbb{T} \}$ consists of interpretations of the meta set τ . Properties of these interpretations induce properties of the meta set. The family might be considered "fuzzy" because some interpretations might occur "more frequently" than others, depending on the structure of τ . The frequency allows for introducing the membership degree of each interpretation set in the whole family. This way, a meta set perceived as a "fuzzy" family of crisp sets, resembles

a fuzzy set. Of course, this informal reasoning may be made rigorous. Note, that all interpretations of the empty meta set \emptyset are equal to itself: $\emptyset_{\mathcal{C}} = \emptyset$.

We now define basic set-theoretic relations for meta sets.

Definition 3. *Let $p \in \mathbf{T}$ and let τ, μ be meta sets. We say that μ belongs to τ under the condition p , if for each branch \mathcal{C} containing p we have $\mu_{\mathcal{C}} \in \tau_{\mathcal{C}}$. We use the symbol ϵ_p for the relation of being a member under the condition p : $\mu \epsilon_p \tau$.*

In other words, $\mu \epsilon_p \tau$, whenever the (crisp) membership relation holds for interpretations of τ and μ determined by p . Similarly we define conditional equality relations.

Definition 4. *Let $p \in \mathbf{T}$ and let τ, μ be meta sets. We say that μ is equal to τ under the condition p , if for each branch \mathcal{C} containing p we have $\mu_{\mathcal{C}} = \tau_{\mathcal{C}}$. We use the symbol \approx_p for the relation of being equal under the condition p : $\mu \approx_p \tau$.*

The relations $\epsilon_{\mathbb{1}}$ and $\approx_{\mathbb{1}}$ designate full membership and equality, like the standard relations for crisp sets. If $\tau \approx_{\mathbb{1}} \mu$, then all their interpretations are pairwise equal: $\tau_{\mathcal{C}} = \mu_{\mathcal{C}}$ for any \mathcal{C} . Similarly for the $\epsilon_{\mathbb{1}}$ relation. On the other hand, if $p < q < \mathbb{1}$, then \approx_p means “less equal” than \approx_q , and ϵ_p denotes smaller degree of membership than ϵ_q .

2.2 Basic Properties of Meta Sets

Let us state some basic properties of meta sets needed further. We start with the simplest example of conditional equality.

Lemma 1. *Let $p, q \in \mathbf{T}_n$ be nodes of the binary tree from the n -th level and let $\tau = \{\langle \emptyset, p \rangle\}$ and $\sigma = \{\langle \emptyset, q \rangle\}$ be meta sets. If $p = q$, then $\forall r \in \mathbf{T}_n \tau \approx_r \sigma$. If $p \neq q$, then $\forall r \in \mathbf{T}_n r \notin \{p, q\} \Leftrightarrow \tau \approx_r \sigma$.*

Proof. First, assume that $p = q$ and let \mathcal{C}_p be a branch in \mathbf{T} containing p . By the Definition 2 we see that $\tau_{\mathcal{C}_p} = \{\emptyset\} = \sigma_{\mathcal{C}_p}$. Since the crisp equality holds for interpretations given by p , then by the Definition 4 also $\tau \approx_p \sigma$. If $n \neq 0$, then there exist other nodes in the level n , so let $r \in \mathbf{T}_n$, $r \neq p$ and let \mathcal{C} be a branch containing r (of course, it cannot contain p). Clearly $\tau_{\mathcal{C}} = \emptyset = \sigma_{\mathcal{C}}$, so the crisp equality holds for these interpretations too, and consequently $\tau \approx_r \sigma$. Therefore, $\forall r \in \mathbf{T}_n \tau \approx_r \sigma$.

To prove the second part (which makes sense for $n > 0$) assume $p \neq q$ and let $r \in \mathbf{T}_n$ be such that $r \neq p$ and $r \neq q$ (such r exist for $n > 1$). If \mathcal{C} is a branch containing r , then similarly as before we have $\tau_{\mathcal{C}} = \emptyset = \sigma_{\mathcal{C}}$, and therefore $\tau \approx_r \sigma$. On the other hand (for $n > 0$), if $r = q$ and \mathcal{C}_q is a branch containing q , then $\tau_{\mathcal{C}_q} = \emptyset$ (since \mathcal{C}_q cannot contain p), whereas $\sigma_{\mathcal{C}_q} = \{\emptyset\}$, so $\tau_{\mathcal{C}_q} \neq \sigma_{\mathcal{C}_q}$, and by the Definition 4 we have $\neg \tau \approx_q \sigma$. Similarly for $r = p$ and a branch $\mathcal{C}_p \ni p$: $\tau_{\mathcal{C}_p} = \{\emptyset\} \neq \emptyset = \sigma_{\mathcal{C}_p}$, so $\neg \tau \approx_p \sigma$ holds too. \square

Note, that we do not define here the relation $\not\approx_p$, so $\neg \tau \approx_p \sigma$ is not equivalent to $\tau \not\approx_p \sigma$. The formula $\neg \tau \approx_p \sigma$ simply means that there exists a branch \mathcal{C} containing p such, that $\tau_{\mathcal{C}} \neq \sigma_{\mathcal{C}}$. In some particular cases we can say even more; the following proposition handles one of such simple cases.

Proposition 1. *Let $S, R \subset \mathbb{T}_n$ be not empty, $q \in \mathbb{T}_n$ and let $\sigma = \{\emptyset\} \times S$ and $\rho = \{\emptyset\} \times R$ be meta sets. If $\neg\sigma \approx_q \rho$, then for any branch \mathcal{C} containing q holds $\sigma_{\mathcal{C}} \neq \rho_{\mathcal{C}}$.*

Proof. According to the Definition [4](#), $\sigma \approx_q \rho$ only if for each branch \mathcal{C} containing q holds $\sigma_{\mathcal{C}} = \rho_{\mathcal{C}}$. Thus, if $\neg\sigma \approx_q \rho$, then there must exist a branch $\bar{\mathcal{C}}$ such, that $\sigma_{\bar{\mathcal{C}}} \neq \rho_{\bar{\mathcal{C}}}$. However, if \mathcal{C}' and \mathcal{C}'' are any branches containing q , then $\sigma_{\mathcal{C}'} = \sigma_{\mathcal{C}''}$ and also $\rho_{\mathcal{C}'} = \rho_{\mathcal{C}''}$, because $\mathcal{C}' \cap S = \mathcal{C}'' \cap S \subset \{q\}$ and similarly $\mathcal{C}' \cap R = \mathcal{C}'' \cap R \subset \{q\}$ (branches are chains in \mathbb{T} , whereas S and R are antichains, so their intersections may contain at most one element). Thus, because all interpretations of σ given by branches containing q are equal (similarly for τ) and for some interpretation \mathcal{C} holds $\sigma_{\mathcal{C}} \neq \tau_{\mathcal{C}}$, then this must hold for all branches. In other words, if $\neg\sigma \approx_q \rho$, then for all branches \mathcal{C} containing q holds $\sigma_{\mathcal{C}} \neq \rho_{\mathcal{C}}$. \square

The next lemma generalizes the Lemma [1](#) to arbitrary subsets of \mathbb{T}_n . It enables evaluation of the equality degree, in other words – similarity degree, of two character samples.

Lemma 2. *Let $P, Q \subset \mathbb{T}_n$ be not empty and let $\tau = \{\langle \emptyset, p \rangle : p \in P\}$ and $\sigma = \{\langle \emptyset, q \rangle : q \in Q\}$. If $R = P \cap Q \cup (\mathbb{T}_n \setminus P) \cap (\mathbb{T}_n \setminus Q)$, then the following implications hold:*

$$r \in R \Rightarrow \tau \approx_r \sigma \quad , \tag{1}$$

$$r \in \mathbb{T}_n \setminus R \Rightarrow \neg\tau \approx_r \sigma \quad . \tag{2}$$

Proof. Assume that $r \in P \cap Q$. If \mathcal{C} is a branch containing r , then clearly $\tau_{\mathcal{C}} = \{\emptyset\} = \sigma_{\mathcal{C}}$, and therefore $\tau \approx_r \sigma$. If $r \in (\mathbb{T}_n \setminus P) \cap (\mathbb{T}_n \setminus Q)$ and \mathcal{C} is a branch containing r , then $\tau_{\mathcal{C}} = \emptyset = \sigma_{\mathcal{C}}$, so $\tau \approx_r \sigma$ holds too. This proves [\(1\)](#).

To prove [\(2\)](#) note, that

$$\begin{aligned} \mathbb{T}_n \setminus R &= \mathbb{T}_n \setminus (P \cap Q \cup (\mathbb{T}_n \setminus P) \cap (\mathbb{T}_n \setminus Q)) \quad , \\ &= (\mathbb{T}_n \setminus P \cap Q) \cap (\mathbb{T}_n \setminus (\mathbb{T}_n \setminus P) \cap (\mathbb{T}_n \setminus Q)) \quad , \\ &= (\mathbb{T}_n \setminus P \cap Q) \cap (P \cup Q) \quad , \\ &= ((\mathbb{T}_n \setminus P) \cup (\mathbb{T}_n \setminus Q)) \cap (P \cup Q) \quad , \\ &= (\mathbb{T}_n \setminus P) \cap Q \cup (\mathbb{T}_n \setminus Q) \cap P \quad . \end{aligned}$$

If $r \in (\mathbb{T}_n \setminus P) \cap Q$, and \mathcal{C} is a branch containing r , then $\tau_{\mathcal{C}} = \emptyset$ and $\sigma_{\mathcal{C}} = \{\emptyset\}$, so $\neg\tau \approx_r \sigma$. Similarly, if $r \in (\mathbb{T}_n \setminus Q) \cap P$, then $\tau_{\mathcal{C}} = \{\emptyset\}$ and $\sigma_{\mathcal{C}} = \emptyset$, so $\neg\tau \approx_r \sigma$. Thus, for $r \in \mathbb{T}_n \setminus R$ we obtain $\neg\tau \approx_r \sigma$. \square

The last lemma is the meta set version of the obvious fact known from the crisp set theory: $x = y \wedge y \in z \Rightarrow x \in z$.

Lemma 3. *If $p \in \mathbb{T}$ and τ, σ, λ are meta sets such, that $\tau \approx_p \sigma$ and $\sigma \epsilon_p \lambda$, then also $\tau \epsilon_p \lambda$.*

Proof. If \mathcal{C} is an arbitrary branch containing p , then by the assumptions $\tau_{\mathcal{C}} = \sigma_{\mathcal{C}}$ and $\sigma_{\mathcal{C}} \in \lambda_{\mathcal{C}}$. Therefore, also $\tau_{\mathcal{C}} \in \lambda_{\mathcal{C}}$, what implies $\tau \epsilon_p \lambda$. \square

We now state the main theorem which allows for calculation of the membership degree of a supplied sample to the compound pattern. The membership degree measures the quality of the sample i.e., its similarity to the defined pattern. In the following theorem the meta set σ represents testing character sample, each π^i for $i = 1 \dots n$ represents a compound pattern element and ρ is the compound character pattern built up of elements π^i .

Theorem 1. *Let $P^i, R^i, S \subset \mathbb{T}_n$ for $i = 1 \dots k$ be not empty. Let $\sigma = \{\emptyset\} \times S$, $\pi^i = \{\emptyset\} \times P^i$ for $i = 1 \dots k$ and $\rho = \bigcup_{i=1}^k \{\pi^i\} \times R^i$ be meta sets. For the sets $Q^i = S \cap P^i \cup (\mathbb{T}_n \setminus S) \cap (\mathbb{T}_n \setminus P^i)$, $i = 1 \dots k$, and $U = \bigcup_{i=1}^k Q^i \cap R^i$, the following holds:*

$$\begin{aligned}
 q \in Q^i &\Rightarrow \sigma \approx_q \pi^i, & (3) \\
 q \in \mathbb{T}_n \setminus Q^i &\Rightarrow \neg \sigma \approx_q \pi^i. & (4) \\
 u \in U &\Rightarrow \sigma \epsilon_u \rho, & (5) \\
 u \in \mathbb{T}_n \setminus U &\Rightarrow \neg \sigma \epsilon_u \rho, & (6)
 \end{aligned}$$

Proof. The Lemma 2 proves (3) and (4).

To prove (5) and (6) let $R = \bigcup_{i=1}^k R^i$ and let $\bar{Q}^i = \mathbb{T}_n \setminus Q^i$. We may split each R^i into two parts: $R^i = (R^i \setminus Q^i) \cup (R^i \cap Q^i) = R^i \cap \bar{Q}^i \cup R^i \cap Q^i$. Therefore,

$$R = \bigcup_{i=1}^k R^i \cap \bar{Q}^i \cup \bigcup_{i=1}^k R^i \cap Q^i = \bigcup_{i=1}^k (R^i \cap \bar{Q}^i) \cup U.$$

Let $u \in \mathbb{T}_n$ and let \mathcal{C} be a branch containing u .

First, let $u \in R$. If $u \in U$, then $u \in R^i \cap Q^i$ for some $i \in \{1 \dots k\}$. By (3) this implies $\sigma \approx_u \pi^i$, since $u \in Q^i$. By the construction of ρ ($u \in R^i$, so $\langle \pi^i, u \rangle \in \rho$) and by the Definition 3 we have $\pi^i \epsilon_u \rho$. Thus, by the Lemma 3 we obtain $\sigma \epsilon_u \rho$, what proves (5). If $u \notin U$ (but still $u \in R$), then let $I \subset \{1 \dots k\}$ be the set of all those i , that $u \in R^i \cap \bar{Q}^i$. Since $u \in \mathcal{C}$ and for each $i \in I$ the intersection $R^i \cap \mathcal{C}$ contains at most one element which is u , then by the Definition 2: $\rho_{\mathcal{C}} = \{\pi_{\mathcal{C}}^i : R^i \cap \mathcal{C} \neq \emptyset\} = \{\pi_{\mathcal{C}}^i : u \in R^i\}$. Note, that $\{i : u \in R^i\} = \{i : u \in R^i \cap \bar{Q}^i\} \cup \{i : u \in R^i \cap Q^i\} = I$, since $u \notin U$. Thus, $\rho_{\mathcal{C}} = \{\pi_{\mathcal{C}}^i : i \in I\}$ and for $i \in I$ holds $\pi_{\mathcal{C}}^i \in \rho_{\mathcal{C}}$. However by (4) and by the Proposition 1 we get $\sigma_{\mathcal{C}} \neq \pi_{\mathcal{C}}^i$ for $i \in I$. Since $\sigma_{\mathcal{C}}$ is different than all the members of $\rho_{\mathcal{C}}$, then $\sigma_{\mathcal{C}} \notin \rho_{\mathcal{C}}$ for any $\mathcal{C} \ni u$, and consequently $\neg \sigma \epsilon_u \rho$.

If $u \in \mathbb{T}_n \setminus R$, then $\rho_{\mathcal{C}} = \emptyset$, therefore also $\neg \sigma \epsilon_u \rho$. This proves (6) and the whole theorem, since either $u \in U$ or $u \in R \setminus U$ or $u \in \mathbb{T}_n \setminus R$. □

The equality degrees as well as the membership degree are subsets of the n -th level of the tree \mathbb{T} . Since there are 2^n nodes on this level, then we may easily map these degrees to rational numbers from the unit interval, dividing the cardinality of the subset representing a degree by 2^n . This mapping allows for evaluating obtained results by means of numbers, what is more human friendly.

3 Character Recognition

Characters might be depicted using rectangular matrices of *width* × *height* cells. For simplicity we consider here a very particular case when *width* · *height* = 2^{*n*} for some *n*. The general case involves additional techniques which are beyond the scope of this paper.

We will explain the method for encoding character samples as meta sets and the method for encoding quality degrees of samples as membership degrees. Then, applying the Theorem 1 we will be able to calculate the quality degree of a new testing sample and its similarity to pattern elements.

3.1 Characters as Meta Sets

To represent a character sample as a meta set, first we must establish a mapping of cells of the matrix to nodes of the *n*-th level of the tree \mathbb{T} . We focus here on 4 × 4 matrices in order to simplify formulas and figures; in practical applications we would rather use matrices comprised of 32 or 64 cells. We map cells of the matrix to nodes from \mathbb{T}_4 as on the Fig. 1 (other mappings are acceptable too).

0000	0001	0010	0011
0100	0101	0110	0111
1000	1001	1010	1011
1100	1101	1110	1111

Fig. 1. Mapping of cells of the 4 × 4 matrix to nodes in \mathbb{T}_4

By marking appropriate cells we may draw a character on the matrix. For instance the Fig. 2 represents two versions of the letter 'L': marked cells contain nodes from \mathbb{T}_4 (depicted as binary sequences), whereas unmarked ones are empty.

0000			
0100			
1000			
1100	1101	1110	

0000			
0100			
1000			
1100	1101	1110	1111

Fig. 2. Two versions of the letter 'L' represented on the matrix: π^1 and π^2

Once the mapping is established we define the meta set representing the given character sample to be the crisp set of ordered pairs whose first element is the empty meta set \emptyset and the second element is a node from \mathbb{T}_4 corresponding to a marked cell. For instance, the character on the left matrix from the Fig. 2 is represented by the following meta set π^1 :

$$\pi^1 = \{ \langle \emptyset, 0000 \rangle, \langle \emptyset, 0100 \rangle, \langle \emptyset, 1000 \rangle, \langle \emptyset, 1100 \rangle, \langle \emptyset, 1101 \rangle, \langle \emptyset, 1110 \rangle \} .$$

Similarly we define π^2 representing 'L' from the right matrix of the Fig. 2:

$$\pi^2 = \{ \langle \emptyset, 0000 \rangle, \langle \emptyset, 0100 \rangle, \langle \emptyset, 1000 \rangle, \langle \emptyset, 1100 \rangle, \langle \emptyset, 1101 \rangle, \langle \emptyset, 1110 \rangle, \langle \emptyset, 1111 \rangle \} .$$

The meta sets π^1 and π^2 are different views on the letter 'L'. Probably one of them is better and another worse. Therefore, we assign them quality degrees in form of sets of nodes or – in other words – in form of membership degrees in some meta set ρ which will represent the compound pattern, and whose domain is comprised of π^1 and π^2 . In this paper we assume that the sets of nodes representing quality degrees are subsets of the same level of \mathbb{T} that is mapped to cells of the matrix (here it is \mathbb{T}_4). This implies that membership degrees are directly proportional to cardinalities of sets of nodes (it is not true in general, cf. [3], [4]). Thus, the larger cardinality of the set, the greater membership degree, and consequently, the better quality. Assuming that π^1 represents the letter 'L' better than π^2 we construct the meta set ρ as follows:

$$\rho = \{ \pi^1 \} \times \{ 0000, 0001, 0010, 0011 \} \cup \{ \pi^2 \} \times \{ 1110, 1111 \} .$$

We have chosen here some arbitrary sets of nodes, small enough to make formulas simple, yet they express the fact, that the π_1 is better than π_2 .

The sets of nodes representing quality degrees should reflect our perception of the quality of the letters. The better one is accompanied by a larger subset of \mathbb{T}_4 , representing greater membership degree. The best one, if it existed, should have the whole \mathbb{T}_4 as the representant of its quality. This rule leaves some indeterminacy. It is possible to express equal degrees of membership by different sets, which differently influence the result. The precise selection of one of the equivalent subsets is subject to experimentation, similarly to the internal structure of a neural network, which usually cannot be determined a priori by some rule, but has to be tuned experimentally to achieve the best result. On the other hand, it is possible to formulate and prove some laws which simplify selection of the subsets best suited for the given task.

The ratio of the number of different nodes paired with π^1 to the number of all nodes in \mathbb{T}_4 is the numerical value of the quality degree of π^1 , similarly for π^2 . For π^1 this ratio is 1/4 and for π^2 it is 1/8, so the fact that the former resembles the letter 'L' better than the latter is properly reflected. Note, that we have given π^1 and π^2 small ratings in order to simplify formulas; in practice, these ratings should be close to 1, since these samples resemble the letter 'L' quite well.

3.2 Evaluating Quality of a Testing Sample

Let σ be a meta set representing a testing sample supplied by a user, like the one from the Fig. 3: $\sigma = \{ \langle \emptyset, 0100 \rangle, \langle \emptyset, 1000 \rangle, \langle \emptyset, 1100 \rangle, \langle \emptyset, 1101 \rangle, \langle \emptyset, 1110 \rangle \}$.

To what measure does this sample match our view of the letter 'L', or – in other words – what is the degree of membership of σ to ρ ? And what are the equality degrees of σ and each π^i (they express the resemblance of the supplied sample to each pattern element)? By the Theorem 1 we may calculate the answer

0100			
1000			
1100	1101	1110	

Fig. 3. A testing sample σ of the letter 'L', supplied by a user

as follows. The constructions of σ , ρ and each π^i imply that the sets S , P^i and R^i from the Theorem 1 have the following contents:

$$\begin{aligned}
 S &= \{ 0100, 1000, 1100, 1101, 1110 \} , \\
 P^1 &= \{ 0000, 0100, 1000, 1100, 1101, 1110 \} , \\
 P^2 &= \{ 0000, 0100, 1000, 1100, 1101, 1110, 1111 \} , \\
 R^1 &= \{ 0000, 0001, 0010, 0011 \} , \\
 R^2 &= \{ 1110, 1111 \} .
 \end{aligned}$$

Therefore,

$$\begin{aligned}
 Q^1 &= S \cap P^1 \cup (\mathbb{T}_4 \setminus S) \cap (\mathbb{T}_4 \setminus P^1) = S \cup (\mathbb{T}_4 \setminus P^1) = \mathbb{T}_4 \setminus \{ 0000 \} , \\
 Q^2 &= S \cap P^2 \cup (\mathbb{T}_4 \setminus S) \cap (\mathbb{T}_4 \setminus P^2) = S \cup (\mathbb{T}_4 \setminus P^2) = \mathbb{T}_4 \setminus \{ 0000, 1111 \} , \\
 U &= Q^1 \cap R^1 \cup Q^2 \cap R^2 = \{ 0001, 0010, 0011 \} \cup \{ 1110 \} ,
 \end{aligned}$$

and finally, for any $p \in \mathbb{T}_4$ we obtain $\sigma \epsilon_p \rho \Leftrightarrow p \in U$ and $\sigma \approx_p \pi^1 \Leftrightarrow p \in Q^1$ and $\sigma \approx_p \pi^2 \Leftrightarrow p \in Q^2$.

As we see, σ represents the letter 'L' equally well as π^1 and better than π^2 , since U has the same number of elements as R^1 and more than R^2 . The numerical ratio for the membership degree of σ in ρ equals $1/4$, like it is the case for π^1 . The sets Q^1 and Q^2 measure similarity of σ to π^1 and π^2 respectively. Since Q^1 includes Q^2 , it follows that σ resembles π^1 better than π^2 .

In general, the obtained results strongly depend not only on the cardinality of R^1 and R^2 , but rather on the nodes they contain. If we change the contents of the sets R^i , but preserve their cardinalities, the results might be completely different. This variability enables supposing additional semantics on the quality degrees R^i , besides the linear ordering of their cardinalities. An example of such semantics is stressing some important areas of the matrix diminishing at the same time the relevance of other cells, i.e., pixels in characters which define the compound pattern. It is done simply by choosing elements of R^i from some crucial area of the matrix, like, for instance, the cells that might contain the dot over the letter 'i'.

The core problem in grading the compound pattern elements, i.e. defining the sets R^i , is to achieve a result (the set U for a supplied sample) consistent with human intuition – a human perception of similar characters. This problem is partially open and is subject to investigations.

4 Conclusions and Further Work

We have described the method for using meta sets to grade similarity of characters for the particular case of characters represented on matrices comprised of 2^n cells. The general case requires additional facts from the meta set theory and it is to be published soon. Note that the presented mechanism seems to be applicable not only to characters (like letters), but to any type of data expressible by means of sets of finite binary sequences, when the problem involves evaluating a degree to which some predefined compound data pattern is matched by a supplied sample.

The existing computer program already handles the general case of arbitrary rectangular matrices. Initial tests confirm that the presented mechanism is able to properly reflect human notion of similarity of character patterns. Particularly – when supplied with accurate pattern data – the program reasonably grades samples which are not member of the compound pattern set (ρ), i.e., it interpolates human view on similarity of characters.

We stress the fact that the theory of meta sets, especially in the form introduced in [3], is directed towards computer implementations and applications. It is because of possible to achieve efficiency of algorithms realizing fundamental relations and operations. The current implementation has a testing character and is to be replaced by a final product in the future. We expect interesting results and computer applications once the final implementation is ready. The theory is under development, new papers on this subject are under preparation; this one includes only the facts that are substantial for the discussed problem.

References

1. Kunen, K.: Set Theory, An Introduction to Independence Proofs. Studies in Logic And Foundations of Mathematics, vol. 102. North-Holland Publishing Company, Amsterdam (1980)
2. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)
3. Starosta, B., Kosiński, W.: Meta Sets. Another Approach to Fuzziness. In: Views on Fuzzy Sets and Systems from Different Perspectives. Philosophy and Logic, Criticisms and Applications. Studies in Fuzziness and Soft Computing, vol. 243, pp. 509–522. Springer, Heidelberg (2009)
4. Starosta, B., Kosiński, W.: Meta Sets. A New Concept of Intuitionistic Fuzzy Sets. In: Intuitionistic Fuzzy Sets: Recent Advances. Volume of Studies in Fuzziness and Soft Computing. Springer, Heidelberg (2009)

A General Framework for Revising Belief Bases Using Qualitative Jeffrey’s Rule

Salem Benferhat¹, Didier Dubois², Henri Prade², and Mary-Anne Williams³

¹ CRIL-CNRS, UMR 8188, Faculté Jean Perrin, Université d’Artois,
rue Jean Souvraz, 62307 Lens, France

² IRIT - Université Paul Sabatier, 118 route de Narbonne
31062 Toulouse cedex 09, France

³ Innovation and Enterprise Research Laboratory, University of Technology,
Sydney NSW 2007, Australia

Abstract. Intelligent agents require methods to revise their epistemic state as they acquire new information. Jeffrey’s rule, which extends conditioning to uncertain inputs, is currently used for revising probabilistic epistemic states when new information is uncertain. This paper analyses the expressive power of two possibilistic counterparts of Jeffrey’s rule for modeling belief revision in intelligent agents. We show that this rule can be used to recover most of the existing approaches proposed in knowledge base revision, such as adjustment, natural belief revision, drastic belief revision, revision of an epistemic by another epistemic state. In addition, we also show that that some recent forms of revision, namely improvement operators, can also be recovered in our framework.

1 Introduction

The information available to intelligent agents is often uncertain, inconsistent and incomplete. It is then crucially important to define tools to manage it in response to the acquisition of new, possibly conflicting, information. The term ‘information’ covers a broad range of entities such as knowledge, perceptions, beliefs, expectations, preferences, or causal relations. It can describe the agent’s view of the world, its actions and its understanding of changes. During the past twenty years, many approaches have been proposed to address the problem of belief change from the axiomatic point of view (e.g., [12], [6]), from the semantics point of view (e.g., [22], [4], [21]) and from the computational point of view [18], [2] (see also [10] for a deep discussion on different forms of revision applying to different kinds of information).

Due to lack of space, this paper only focuses on the semantics of belief revision in the framework of possibility theory. The basic object in possibility theory is a possibility distribution, which is a mapping from the set of classical interpretations to a totally ordered structure, usually the interval $[0, 1]$.

The revision of a possibility distribution can be viewed as a so-called “transmutation” [16] that modifies the ranking of interpretations so as to give priority to the input information. In particular, two forms of possibilistic revision

(based on minimum and product respectively) are investigated as counterparts to Jeffrey’s rule of revision in probability theory. These two forms of possibilistic revision consist in modifying a possibility distribution π with a set of weighted, mutually exclusive formulas, denoted by $\mu = \{(\phi_i, a_i), i = 1, \dots, n\}$, where the propositional formulas ϕ_i ’s induce a partition of a set of interpretations. This set μ will be called a partial epistemic state, and expresses a set of constraints stating that the *possibility* degree of ϕ_i is equal to a_i . This is not to be confused with standard possibilistic logic formulas. Each of the two forms of possibilistic revision comes down to modifying the possibility distribution π such that each formula ϕ_i will be plausible to the prescribed degree a_i . The new degrees a_i ’s may be either constants determined for example by an expert, or a function defined for instance w. r. t. the original possibility degree associated with ϕ_i .

This paper first extends natural properties described in [2] in order to take into account the new form of the input, namely a partial epistemic state. Then we present two definitions of possibilistic revision operators that naturally extend the two well-known forms of conditioning that have been defined in the possibility theory framework. We also compare possibilistic revision with the counterpart of Jeffrey’s rule of conditioning. In its second half, the paper shows that most of existing belief revision operators can be recovered by one of the two forms of possibilistic revision. But first in order to establish the new results, we need to restate the necessary background on possibility theory.

2 Possibilistic Representations of Epistemic States

Let L be a finite propositional language with formulas ϕ , or ψ . \models denotes the (semantical) classical consequence relation. Ω is the set of classical interpretations, and $[\phi]$ is the set of classical models of ϕ .

There are several common representations of epistemic states such as : well ordered partitions of Ω , probabilistic epistemic states, Grove’s systems of spheres, Spohn’s Ordinal Conditional Functions (OCF), etc. Throughout this paper we use a general representation of a total preorder, namely a possibility distribution π , which is a mapping from Ω to the interval $[0,1]$.

A possibility distribution can be used for representing any total preorder of possible worlds. But all representations do not have the same expressive power. The purely ordinal one (a plausibility relation) differs from the qualitative encoding of a possibility distribution (on a totally ordered scale) as the former cannot express impossibility. We will identify operators that require the full power of the $[0, 1]$ scale, which is much more expressive.

Given an interpretation $\omega \in \Omega$, $\pi(\omega)$ represents the degree of compatibility of ω with the available information (or beliefs) about the real world. $\pi(\omega) = 0$ means that the interpretation ω is impossible, and $\pi(\omega) = 1$ means that nothing prevents ω from being the real world. Interpretations ω where $\pi(\omega) = 1$ are considered to be expected (they are not at all surprising). When $\pi(\omega) > \pi(\omega')$, ω is preferred to ω' as a candidate for being the real state of the world. The less $\pi(\omega)$, the less plausible ω , or the more different it is to the current world. A

possibility distribution π is said to be *normal* if $\exists \omega \in \Omega$, such that $\pi(\omega) = 1$, in other words if at least one interpretation is a fully plausible candidate for being the actual world.

Given a possibility distribution π , the possibility degree of formula ϕ is defined as:

$$II(\phi) = \max\{\pi(\omega) : \omega \in [\phi]\}.$$

It evaluates the extent to which ϕ is consistent with the available information expressed by π . Note that $II(\phi)$ is evaluated under the assumption that the situation where ϕ is true is as normal as it can be (since $II(\phi)$ reflects the maximal plausibility of a model of ϕ).

Given a possibility distribution π , we define a belief set [12], denoted by $Bel(\pi)$, as a set of accepted beliefs [11], [1], obtained by considering all sentences that are more plausible than their negation, namely:

$$Bel(\pi) = \{\phi : II(\phi) > II(\neg\phi)\}.$$

Namely, $Bel(\pi)$ is a classical theory whose models are the interpretations having the highest degrees in π . When π is normalized, models of $Bel(\pi)$ are interpretations that are completely possible, namely $[Bel(\pi)] = \{\omega : \pi(\omega) = 1\}$. The sentence ϕ belongs to $Bel(\pi)$ when ϕ holds in all the most normal or plausible situations (hence ϕ is expected, or accepted as being true).

Lastly, given a formula ϕ , two different types of conditioning [9] have been defined in possibility theory (when $II(\phi) > 0$):

- In an ordinal setting, we have:

$$\begin{aligned} \pi(\omega \mid_m \phi) &= 1 \text{ if } \pi(\omega) = II(\phi) \text{ and } \omega \models \phi \\ &= \pi(\omega) \text{ if } \pi(\omega) < II(\phi) \text{ and } \omega \models \phi \\ &= 0 \text{ if } \omega \notin [\phi]. \end{aligned} \tag{1}$$

This is the definition of *minimum-based conditioning*.

- In a numerical setting, we get:

$$\begin{aligned} \pi(\omega \mid \cdot \phi) &= \frac{\pi(\omega)}{II(\phi)} \text{ if } \omega \models \phi \\ &= 0 \text{ otherwise} \end{aligned} \tag{2}$$

This is the definition of *product-based conditioning*.

These two definitions of conditioning satisfy an equation of the form

$$\forall \omega, \pi(\omega) = \pi(\omega \mid \phi) \oplus II(\phi),$$

which is similar to Bayesian conditioning, where \oplus is min and the product respectively. The rule based on the product is much closer to genuine Bayesian conditioning than the qualitative conditioning defined from the minimum which is purely based on the comparison of levels; product-based conditioning requires more of the structure of the unit interval. Besides, when $II(\phi) = 0, \pi(\omega \mid_m \phi) = \pi(\omega \mid \cdot \phi) = 1, \forall \omega$, by convention.

3 Iterated Semantic Revision in Possibilistic Logic

Belief revision results from the effect of accepting a new piece of information called the input information. In this paper, it is assumed that the current epistemic state or generic knowledge (represented by a possibility distribution), and the input information, do not play the same role. The input takes priority over information in the epistemic state. This asymmetry is expressed by the way the belief change problem is stated, namely the new information alters the epistemic state and not conversely. This asymmetry will appear clearly at the level of belief change operations. This situation is different from the one of information fusion from several sources, where no epistemic state dominates *a priori*. In this context, the use of symmetrical rules is natural especially when the sources are equally reliable.

3.1 Jeffrey’s Rule for Revising Probability Distributions

In probability theory, there is a natural method for reasoning in the presence of new pieces of uncertain information. This is achieved using Jeffrey’s rule [13], which is proposed for revising probability distributions based on the probability kinematics principle whose objective is minimizing change. In this method, generic knowledge is represented as a probability distribution. Jeffrey’s rule [13] provides an effective means to revise a probability distribution p to p' given an input with probability bearing on a set of *mutually exclusive* and *exhaustive* events ϕ_i . Note that when speaking of events, ϕ is short for $[\phi]$. The input information is given in the form of pairs (ϕ_i, a_i) with:

$$P'(\phi_i) = a_i. \tag{3}$$

Jeffrey’s method relies on the assumption that, while the probability on a prescribed subalgebra of events is enforced by the input information, the conditional probability of any event $\psi \subseteq \Omega$ given any uncertain event ϕ_i in this subalgebra is the same in the original and the revised distributions. Namely,

$$\forall \phi_i, \forall \psi, P(\psi|\phi_i) = P'(\psi|\phi_i). \tag{4}$$

The underlying interpretation of revision implied by the constraint of Equation 4 is that the revised probability distribution p' must not change conditional probability degrees of any event ϕ given uncertain events ϕ_i . In the probabilistic framework, applying Bayes rule then marginalization allows revision of the possibility degree of any event ψ in the following way:

$$P'(\psi) = \sum_{\phi_i} P'(\phi_i) * \frac{P(\psi, \phi_i)}{P(\phi_i)}. \tag{5}$$

This technique (known as Jeffrey’s rule of conditioning) yields the unique distribution that satisfies (3) and (4) (see [5]).

3.2 Two Forms of Possibilistic Revision Based on Jeffrey’s Rule

The possibilistic counterpart of Jeffrey’s rule was introduced in [7] (see also [8]), without emphasizing the probability kinematics condition (4) however. There are two natural ways to define a possibilistic revision based on Jeffrey’s rule, which naturally extend the two forms of conditioning that exist in possibility theory.

Note that most existing works on belief revision (both from semantics and axiomatics perspectives) assume that the input information is either a propositional formula, or an epistemic state (namely a possibility distribution).

Defining a possibilistic revision based on Jeffrey’s rule allows us to define a general framework is a possibility distribution that bears on a partition of the set of interpretations. Namely, the input is of the form $\mu = \{(\phi_i, a_i) \mid i = 1, m\}$ where the ϕ_i ’s are pairwise mutually exclusive formulas, and represent a partition of set of interpretations Ω (namely, $\forall \phi_i, \phi_j, [\phi_i] \cap [\phi_j] = \emptyset$ and $\bigcup_{i=1, m} [\phi_m] = \Omega$). The only requirement is that there exists at least one a_j such that $a_j = 1$. In the following, μ will be called a partial epistemic state. It is partial in the sense that letting $\Pi'(\phi_i) = a_i$ (Π' is defined from π') does not amount to the full specification of π' over the models of ϕ_i .

Let us first discuss some natural properties of the revision of a possibility distribution π and a new input information $\mu = \{(\phi_i, a_i) \mid i = 1, m\}$ to a new possibility distribution denoted by $\pi' = \pi(\cdot | \mu)$. Natural properties for π' are:

- A₁**: π' should be normalized,
- A₂**: $\forall (\phi_i, a_i) \in \mu, \Pi'(\phi_i) = a_i$.
- A₃**: $\forall \omega, \omega' \in [\phi_i]$ if $\pi(\omega) \geq \pi(\omega')$ then $\pi'(\omega) \geq \pi'(\omega')$,
- A₄**: If for all $\phi_i, \Pi(\phi_i) = a_i$ then $\forall \omega \in [\phi_i] : \pi(\omega) = \pi'(\omega)$,
- A₅**: If $\pi(\omega) = 0$ then $\pi'(\omega) = 0$.

A₁ means that the new epistemic state is consistent. **A₂** confirms that the input (ϕ, a) is interpreted as a constraint which forces π' to satisfy:

$$\Pi'(\phi_i) = a_i.$$

A₃ means that the new possibility distribution should preserve the previous relative order (in the wide sense) between models of each ϕ_i . A stronger version of **A₃** can be defined:

$$\mathbf{A}'_3 : \forall \omega, \omega' \in [\phi_i] \text{ then: } \pi(\omega) > \pi(\omega') \text{ iff } \pi'(\omega) > \pi'(\omega'),$$

A'₃ clearly extends **CR₁**, **CR₂** proposed in [6]. **A₄** means that when all new beliefs ϕ_i are accepted at their prescribed levels a_i then revision does not affect π . **A₅** stipulates that impossible worlds remain impossible after revision. Note that there are no further constraints which relate models of different ϕ_i in the new epistemic state.

However in the qualitative and quantitative cases, the previous properties **A₁**–**A₅** do not guarantee a unique definition of conditioning. **A₃** suggests that the possibilistic revision process can be achieved using several parallel changes

with a sure input: First, apply a conditioning (using equations **(1)** or **(2)**) on each ϕ_i and in order to satisfy **A₂**, the distribution $\pi(\cdot \mid \neg\phi)$ is denormalized so as to satisfy $\Pi'(\phi_i) = a_i$. Therefore, revising with μ can be achieved using the following definition:

$$\forall \phi_i \in \mu, \forall \omega \models \phi_i, \pi(\omega \mid \mu) = a_i \oplus \pi(\omega \mid_{\oplus} \phi_i) \tag{6}$$

where \oplus is either min or the product, depending on whether conditioning is based on the product or the minimum operator. When $\oplus =$ product (resp. min) the possibilistic revision will be simply called product-based (resp. minimum-based) conditioning with partial epistemic states.

The new possibility degrees of models of ϕ_i depend on the relative position of the *a priori* possibility degrees of ϕ_i , and the prescribed posterior possibility degree of ϕ_i :

- If $\Pi(\phi_i) \geq a_i$ and when $\oplus = \text{min}$, all interpretations that were originally more plausible than a_i , are forced to level a_i , which means that some strict ordering between models of ϕ_i may be lost. Hence **A₃'** is clearly not satisfied, and the result does not coincide with the ordinal revision rule. When $\oplus = \text{product}$, all plausibility levels are proportionally shifted down (to the level a_i).
- If $\Pi(\phi_i) < a_i$ the best models of ϕ_i are raised to level a_i . Moreover, when $\oplus = \text{product}$, the plausibility levels of other models are proportionally shifted up (to level a_i).

3.3 Relationships with Jeffrey’s Kinematics Properties

Another way to define possibilistic revision is to simply define counterparts to Jeffrey’s rule axioms **[13]**. Namely, given an initial possibility distribution π and a partial epistemic state $\mu = \{(\phi_i, a_i) \mid i = 1, m\}$ we need to find possibility distributions π' that satisfy:

1. $\Pi'(\phi_i) = a_i$.
2. $\forall \phi_i, \forall \psi, \Pi(\psi \mid_{\oplus} \phi_i) = \Pi'(\psi \mid_{\oplus} \phi_i)$,

where \oplus is either a minimum or a product. When \oplus is the product then we can show that the possibilistic revision given by **(6)** is the unique possibility distribution that satisfies condition 1 and 2. However, it is not the case when \oplus is the minimum, where in general condition 2 is not satisfied.

4 Recovering Existing Belief Revision Frameworks

4.1 Standard Possibilistic Conditioning and Adjustment

Clearly, possibilistic revision with partial epistemic states generalizes possibilistic conditioning with a propositional formula ϕ . Indeed, applying possibilistic revision given by **(6)** with a partial epistemic state $\mu = \{(\phi, 1), (\neg\phi, 0)\}$ gives exactly the same results if one applies equation **(1)** on ϕ when $\oplus = \text{min}$ (resp. **(2)** for $\oplus = \text{product}$). Similarly, possibilistic revision with uncertain input, which corresponds to adjustment (see **[2]**), is a particular case of possibilistic revision with a partial epistemic state, where the input is of the form $\mu = \{(\phi, 1), (\neg\phi, a)\}$.

4.2 Natural Belief Revision

Let $<_{initial}$ be a complete pre-order on the set of epistemic states. Let ϕ be a new piece of information. We denote by $<_N$ the result of applying natural belief revision of $<_{initial}$ by ϕ . Natural belief revision of $<_{initial}$ by ϕ proposed in [4], also hinted by Spohn [20], proceeds to minimal change of $<_{initial}$ by considering the most plausible models of ϕ in $<_{initial}$ to become the most plausible interpretations in $<_N$. More precisely, $<_N$ is defined as follows:

- $\forall \omega \in \min(\phi, <_{initial}), \forall \omega' \in \min(\phi, <_{initial}), \omega =_N \omega'$
- $\forall \omega \in \min(\phi, <_{initial}), \forall \omega' \notin \min(\phi, <_{initial}), \omega <_N \omega'$
- $\forall \omega \notin \min(\phi, <_{initial}), \forall \omega' \notin \min(\phi, <_{initial}), \omega <_N \omega'$ iff $\omega <_{initial} \omega'$.

To recover natural belief revision, first associate with $<_{initial}$ a compatible positive possibility distribution¹ $\pi_{initial}$, on $[0, 1]$, defined by:

$$\forall \omega, \omega' \in \Omega, \pi_{initial}(\omega) > \pi_{initial}(\omega') \text{ iff } \omega <_{initial} \omega'.$$

Such $\pi_{initial}$ always exists. Then let a be such that $1 > a > \max\{\pi(\omega) : \pi(\omega) \neq 1\}$. Then let $[\phi^*] = \min(\phi, <_{initial})$, and define $\pi_{<_N}(\cdot) = \pi_{input}(\cdot |_{\mu})$ where $\mu = \{(\phi^*, 1), (\neg\phi^*, a)\}$, $\pi_{input}(\cdot |_{\mu})$ is the result applying possibilistic revision given by equation (6) with $\oplus = \min$ and replacing ϕ by ϕ^* . Then we can show that $\pi_{<_N}$ indeed encodes natural belief revision, namely:

$$\forall \omega, \omega' \in \Omega, \pi_{<_N}(\omega) > \pi_{<_N}(\omega') \text{ iff } \omega <_N \omega'.$$

4.3 Drastic Belief Revision

Papini [19] (see also [17], [14]), has considered a stronger constraint (also hinted by Spohn [20]) by imposing that each model of ϕ should be strictly preferred to each countermodel of $\neg\phi$, and moreover the relative ordering between models (resp. countermodels) of ϕ should be preserved. More formally, let us denote by $<_D$ be result of applying drastic belief revision of $<_{initial}$ by ϕ . $<_D$ is defined as follows:

- $\forall \omega, \omega' \in [\phi], \omega <_D \omega' \text{ iff } \omega <_{initial} \omega'$.
- $\forall \omega, \omega' \notin [\phi], \omega <_D \omega' \text{ iff } \omega <_{initial} \omega'$.
- $\forall \omega \in [\phi], \forall \omega' \notin [\phi], \omega <_D \omega'$.

Note that this is a special case of Jeffrey’s rule for possibility orderings. To recover drastic belief revision, first associate with $<_{initial}$ a compatible positive possibility distribution π_{input} , as defined above. Let $\Delta(\phi) = \min\{\pi(\omega) : \omega \models \phi\}$, and a such that $a < \Delta(\phi)$.

Then define $\pi_{<_D}(\cdot) = \pi_{input}(\cdot |_{\mu})$ where $\mu = \{(\phi, 1), (\neg\phi, a)\}$, $\pi_{input}(\cdot |_{\mu})$ is the result applying possibilistic revision given by equation (6) with $\oplus = \text{product}$. Then we can show that $\pi_{<_D}$ indeed encodes drastic belief revision, namely:

$$\forall \omega, \omega' \in \Omega, \pi_{<_D}(\omega) > \pi_{<_D}(\omega') \text{ iff } \omega <_D \omega'.$$

¹ A possibility distribution π is said to be positive if $\forall \omega, \pi(\omega) > 0$.

4.4 Revising Generic Knowledge by Generic Knowledge

In [3] a revision of an epistemic state (representing generic knowledge), denoted here by $\langle_{initial}$, by an input in the form of an epistemic state, denoted here by \langle_{input} , is defined. The obtained result is a new epistemic state, denoted by \langle_L (L for lexicographic ordering), and defined as follows:

- $\forall \omega, \omega' \in \Omega$, if $\omega \langle_{input} \omega'$ then $\omega \langle_L \omega'$.
- $\forall \omega, \omega' \in \Omega$, if $\omega =_{input} \omega'$ then $\omega \langle_L \omega'$ iff $\omega \langle_{initial} \omega'$.

Namely, \langle_L is obtained by refining \langle_{input} by $\langle_{initial}$. For our purpose, we denote $\{E_0, \dots, E_n\}$ the partition of Ω induced by \langle_{input} . Namely:

- $\forall i, j \in \{0, \dots, n\}$, $E_i \cap E_j = \emptyset$, and $\bigcup_{i=1, \dots, n} E_i = \Omega$
- $\forall i \in \{0, \dots, n\}$, $\forall \omega, \omega' \in E_i$, $\omega =_{input} \omega'$,
- $\forall \omega, \omega' \in \Omega$, $\omega \langle_{input} \omega'$ iff $\omega \in E_i, \omega' \in E_j$ and $i < j$.

Let $\pi_{initial}$ and π_{input} be two positive possibility distributions associated respectively with $\langle_{initial}$ and \langle_{input} . To recover this revision of an epistemic state by an epistemic state, first define $\pi_{\langle_L}(\cdot) = \pi_{input}(\cdot | \mu)$ where $\mu = \{(\phi_{E_i}, \epsilon^i) : i = 0, \dots, n\}$ is the result applying possibilistic revision given by equation (6) with $\oplus =$ product. ϕ_{E_i} is a propositional formula that exactly admits E_i as the set of its models. ϵ_i 's are infinitesimal (and by convention $\epsilon^0 = 1$). Then we can show that π_{\langle_L} indeed encodes \langle_L , namely:

$$\forall \omega, \omega' \in \Omega, \pi_{\langle_L}(\omega) > \pi_{\langle_L}(\omega') \text{ iff } \omega \langle_L \omega'.$$

4.5 Improvement Operator

The last approach that we propose to recover is called a reinforcement or improvement operator, recently proposed in [15]. The idea is that a revision of $\langle_{initial}$ by a propositional formula ϕ only allows a small increase of the plausibility of ϕ , namely the result makes ϕ “one unit” more plausible.

The new epistemic state, denoted by \langle_R , obtained after reinforcing ϕ is defined as follows. Let ω be a model of ϕ and ω' be a counter-model of ϕ . Let

$$G_\omega = \{\omega_1, \omega_1 \langle_{initial} \omega \text{ and } \nexists \omega_2 \text{ such that } \omega_1 \langle_{initial} \omega_2 \langle_{initial} \omega\}.$$

G_ω represents the set of interpretations which are “just”, or “slightly” (but strictly) preferred to ω , in the sense of $\langle_{initial}$. Then we have:

- The relative ordering between models (resp. counter-models) of ϕ is preserved.
- if $\omega \langle_{initial} \omega'$ then $\omega \langle_R \omega'$
- if $\omega' \langle_{initial} \omega$ then either $\omega' \in G_\omega$ and $\omega =_R \omega'$ or $\omega' \langle_R \omega$

To recover the reinforcement operator, we first define π_{input} to be a positive possibility distribution associated with $\langle_{initial}$, as defined above. Let $S = \{a_0 = 1, a_1, \dots, a_n\}$ be a finite scale $a_0 = 1 > a_1 > \dots > a_n > 0$ (n is at least equal to twice the number of different degrees in π_{input}). To ensure the success of our

encoding, we assume that $a_i = 2^{-i}$. Define $pred(a_i) = a_{i-1}$ with by convention $pred(a_0) = 1$, and $succ(a_i) = a_{i+1}$ with by convention $succ(a_n) = a_n$.

The way the reinforcement of ϕ is implemented depends on the position of the best models of ϕ with respect to the best models of $\neg\phi$. If $\Pi(\neg\phi) > \Pi(\phi)$ (namely best models of $\neg\phi$ are strictly preferred to best models of ϕ) then we simply shift up “by one unit” possibility degrees of models of ϕ (thanks to $pred$ function). Conversely, if $\Pi(\neg\phi) \leq \Pi(\phi)$ then we simply shift down “by one unit” possibility degrees of countermodels of ϕ (with the help of $succ$ function). This means that the possibility degree of new formula ϕ (after its reinforcement) will be $pred(\Pi(\phi))$ (recall that $pred(1) = 1$). While the possibility degree of $\neg\phi$ (after the reinforcement of ϕ) is $succ(\Pi(\neg\phi)) \ominus \Pi(\phi)$, where $a \ominus b$ is equal to 1 if $b < 1$ and is equal to a otherwise. Therefore if one defines $\pi_{<R}(\cdot) = \pi_{initial}(\cdot | \mu)(\cdot)$ where $\mu = \{(\phi, pred(\Pi(\phi))), (\neg\phi, succ(\Pi(\neg\phi)) \ominus \Pi(\phi))\}$, then we can show that $\pi_{<R}$ indeed encodes $<_R$. Namely: $\forall \omega, \omega' \in \Omega, \pi_{<R}(\omega) > \pi_{<R}(\omega')$ iff $\omega <_R \omega'$.

5 Conclusion

Due to the fundamental nature of the need to maintain an epistemic state that faithfully reflects an agent’s understanding there has been considerable scientific efforts invested in developing effective belief revision mechanisms and strategies such as [12], [6], [22], [18], and [2]. In this paper we show how Jeffrey’s rule can be used to justify several key existing approaches to belief revision, then having established this sound relationship we show that reinforcement operators can be specified using our framework. Moreover, we propose a new form of belief revision where the input is only a partial representation of epistemic states using Jeffrey’s rule. All these methods can be used to enhance the belief management capabilities of intelligent agents.

Acknowledgments. The authors would like to thank Sebastien Konieczny for his useful comments. Research of the first author was supported by grants from ANR projects MICRAC and PLACID.

References

1. Ben Amor, N., Benferhat, S., Dubois, D., Geffner, H., Prade, H.: Independence in qualitative uncertainty frameworks. In: Seventh International Conference on Principles of Knowledge Representation and Reasoning KR2000, Breckenridge, Colorado, pp. 235–246. Morgan Kaufmann, San Francisco (2000)
2. Benferhat, S., Dubois, D., Prade, H., Williams, M.-A.: A practical approach to revising prioritized knowledge bases. *Studia Logica Journal* 70, 105–130 (2002)
3. Benferhat, S., Konieczny, S., Papini, O., Pino Pérez, R.: Iterated revision by epistemic states: axioms, semantics and syntax. In: Proc. of the 14th European Conf. on Artificial Intelligence (ECAI 2000), Berlin, Allemagne, August 2000, pp. 13–17. IOS Press, Amsterdam (2000)
4. Boutilier, C.: Revision Sequences and Nested Conditionals. In: Proc. of the 13th Inter. Joint Conf. on Artificial Intelligence (IJCAI 1993), pp. 519–531 (1993)

5. Chan, H., Darwiche, A.: On the revision of probabilistic beliefs using uncertain evidence. *Artificial Intelligence* 163, 67–90 (2005)
6. Darwiche, A., Pearl, J.: On the logic of iterated revision. *Artificial Intelligence* 89, 1–29 (1997)
7. Dubois, D., Prade, H.: Updating with belief functions, ordinal conditional functions and possibility measures. In: Bonissone, P.P., Henrion, M., Kanal, L.N., Lemmer, J.F. (eds.) *Uncertainty in Artificial Intelligence* 6, pp. 311–329. Elsevier Science Publ. B.V., Amsterdam (1991)
8. Dubois, D., Prade, H.: A synthetic view of belief revision with uncertain inputs in the framework of possibility theory. *Int. J. Approx. Reasoning* 17, 295–324 (1997)
9. Dubois, D., Prade, H.: Possibility theory: qualitative and quantitative aspects. In: Gabbay, D., Smets, P. (eds.) *Handbook of Defeasible Reasoning and Uncertainty Management Systems. Quantified Representation of Uncertainty and Imprecision*, vol. 1, pp. 169–226 (1998)
10. Dubois, D.: Three scenarios for the revision of epistemic states. *J. Log. Comput.* 18(5), 721–738 (2008)
11. Dubois, D., Fargier, H., Prade, H.: Ordinal and probabilistic representations of acceptance. *J. Artif. Intell. Res. (JAIR)* 22, 23–56 (2004)
12. Gärdenfors, P.: *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. Bradford Books, MIT Press, Cambridge (1988)
13. Jeffrey, R.C.: *The logic of decision*. Mc. Graw Hill, New York (1965)
14. Konieczny, S., Pino Pérez, R.: A framework for iterated revision. *Journ. of Applied Non-Classical Logics* 10(3-4) (2000)
15. Konieczny, S., Pino Perez, R.: Improvement operators. In: 11th International Conference on Principles of Knowledge Representation and Reasoning (KR 2008), pp. 177–186 (2008)
16. Makinson, D.: General patterns in nonmonotonic inference. In: Gabbay, D.M., Hogger, C.J., Robinson, J.A., Nute, D. (eds.) *Handbook of Logic in Artificial Intelligence and Logic Programming*, vol. 3, pp. 35–110. Oxford University Press, Oxford (1994)
17. Nayak, A.: Iterated belief change based on epistemic entrenchment. *Erkenntnis* 41, 353–390 (1994)
18. Nebel, B.: Base revision operations and schemes: semantics, representation, and complexity. In: *Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI 1994)*, pp. 341–345 (1994)
19. Papini, O.: Iterated revision operations stemming from the history of an agent’s observations. *Frontiers of Belief Revision* (to appear, 2000)
20. Spohn, W.: Ordinal conditiona functions: a dynamic theory of epistemic states. *Causation in Decision, Belief Change, and Statistics* 2, 105–134 (1988)
21. Thielscher, M.: Handling implicational and universal quantification constraints in flux. In: van Beek, P. (ed.) *CP 2005. LNCS*, vol. 3709, pp. 667–681. Springer, Heidelberg (2005)
22. Williams, M.A.: Transmutations of Knowledge Systems. In: Doyle, J., et al. (eds.) *Inter. Conf. on principles of Knowledge Representation and reasoning (KR 1994)*, pp. 619–629. Morgan Kaufmann, San Francisco (1994)

Author Index

- Abe, Hidenao 251
Abraham, Ajith 533
Appice, Annalisa 563
Arias-Aranda, Daniel 463
Atzmueller, Martin 35
- Babič, František 321
Bakker, Jorn 493
Bariatier, Patrick 171
Barták, Roman 582
Basile, Pierpaolo 241
Béchet, Nicolas 431
Benferhat, Salem 612
Berzal, Fernando 15, 271
Biba, Marenglen 402
Bieliková, Mária 331
Błaszczycyński, Jerzy 382
Bobbillo, Fernando 151
Bombini, Grazia 361
Bonnaire, Xavier 483
Bosc, Patrick 311
Boulcaut, Jean-François 513
Budzynska, Katarzyna 201
Buffett, Scott 442
- Cabanes, Guenael 341
Caputo, Annalina 241
Caruso, Costantina 563
Castro, Juan L. 463
Ceci, Michelangelo 119, 563
Cerf, Loïc 513
Cestnik, Bojan 129
Chomatek, Lukasz 351
Chudá, Daniela 331
Colucci, Simona 473
Crémilleux, Bruno 45
Cubero, Juan-Carlos 15
Cuissart, Bertrand 45
- d'Amato, Claudia 161
Dapoigny, Richard 171
Dardzińska, Agnieszka 66
Di Mauro, Nicola 361
Di Noia, Tommaso 473
Di Sciascio, Eugenio 473
- Djouadi, Yassine 592
Dubois, Didier 612
- El-Béze, Marc 431
Esposito, Floriana 161, 361, 402
- Fajardo, Waldo 271
Fanizzi, Nicola 161
Ferilli, Stefano 361, 402
Fraňová, Marta 573
Fresnau, Dominique 341
Fumarola, Fabio 563
- Galassi, Ugo 341
Gao, Chao 503
Geng, Liqiang 442, 453
Giordana, Attilio 341
Grekow, Jacek 261
Grzymala-Busse, Jerzy W. 25
Gu, Wei 453
- Habib, Sami 542
Hadzikadic, Mirsad 301
Hamilton, Bruce 442
Horák, Zdeněk 533
- Ilijašić, Lovro 523
- Jaudoin, Hélène 139
Jiménez, Aída 15, 271
Jirkovský, Vojtěch 88
- Kacprzak, Magdalena 201
Karagianni, Georgia 109
Kelemen, Jozef 5
Kessler, Rémy 431
Khan, Latifur 552
Kliegr, Tomáš 88
Kłopotek, Mieczysław 371
Kočířbová, Jana 533
Kodratoff, Yves 573
Korba, Larry 442
Kubera, Elżbieta 281
Kubik-Komar, Agnieszka 281, 291
Kursa, Miron 281
Kyriazopoulos, G. 109

- Lemmerich, Florian 35
 Lewandowski, Jacek 211
 Lin, Zuoquan 181
 Liu, Hongyu 442
 Liu, Jiming 503
 Loglisci, Corrado 119, 563

 Malerba, Donato 119, 563
 Mannila, Heikki 1
 Marimuthu, Paulvanna Nayaki 542
 Masud, Mohammad M. 552
 Molina-Solana, Miguel 271
 Murray, Neil V. 191

 Navarro, Maria 463
 Návrat, Pavol 331
 Nemrava, Jan 88
 Nguyen, Tran Bao Nhan 513
 Nováková, Lenka 56

 Otto, Eridan 483

 Papageorgiou, Elpiniki I. 99, 109
 Papandrianos, Nikolaos 109
 Paralič, Ján 321
 Pechenizkiy, Mykola 493
 Petrič, Ingrid 129
 Pisetta, Vincent 422
 Pivert, Olivier 311
 Poezevara, Guillaume 45
 Polášek, Ivan 5
 Poniszewska-Marañda, Aneta 351
 Ponti, Giovanni 231
 Prade, Henri 392, 592, 612

 Ralbovský, Martin 88
 Raś, Zbigniew W. 66, 261
 Rauch, Jan 76
 Rembelski, Pawel 201
 Ren, Shuang 181
 Rico, Agnes 392
 Riff, María-Cristina 483
 Roche, Mathieu 431
 Rosenthal, Erik 191
 Rozinajová, Viera 331
 Rudnicki, Witold 281
 Rybinski, Henryk 211

 Saitta, Lorenza 523
 Saric, Amar 301

 Schneider, Michel 139
 Semeraro, Giovanni 241
 Serrurier, Mathieu 392
 Sfyras, D. 109
 Simonenko, Ekaterina 321
 Šimůnek, Milan 76, 88
 Snášel, Václav 533
 Soufflet, Olivier 311
 Spyrtatos, Nicolas 321
 Starosta, Bartłomiej 602
 Stefanowski, Jerzy 382
 Štěpánková, Olga 56
 Straccia, Umberto 151, 473
 Sugibuchi, Tsuyoshi 321
 Svátek, Vojtěch 88

 Tagarelli, Andrea 231
 Taha, Mohammad 542
 Tinelli, Eufemia 473
 Toropila, Daniel 582
 Torres-Moreno, Juan-Manuel 431
 Toumani, Karima 139
 Tsumoto, Shusaku 251

 Urbančić, Tanja 129

 Wagner, Jozef 321
 Wang, Deqing 221
 Wang, Xin 442, 453
 Wang, Yunli 442
 Widmer, Gerhard 2
 Wiczorkowska, Alicja 281, 291
 Williams, Mary-Anne 612
 Wilson, David 301
 Wolski, Michał 371
 Woolam, Clay 552

 Zając, Magdalena 382
 Zemánek, Jan 88
 Zemankova, Maria 3
 Zhang, Hui 221
 Zhang, Zhihu 181
 Zhong, Ning 503
 Zhou, Gang 221
 Zighed, Djamel A. 422
 Žliobaitė, Indrė 412
 Zurita, José M. 463