Marzena Kryszkiewicz
James F. Peters
Henryk Rybinski
Andrzej Skowron (Eds.)

LNAI 4585

Rough Sets and Intelligent Systems Paradigms

International Conference, RSEISP 2007
Warsaw, Poland, June 2007
Proceedings

Springer

Lecture Notes in Artificial Intelligence 4585

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Marzena Kryszkiewicz James F. Peters
Henryk Rybinski Andrzej Skowron (Eds.)

Rough Sets and Intelligent Systems Paradigms

International Conference, RSEISP 2007
Warsaw, Poland, June 28-30, 2007
Proceedings

Springer

Volume Editors

Marzena Kryszkiewicz
Henryk Rybinski
Warsaw University of Technology
Institute of Computer Science, Warsaw, Poland
E-mail: {M.Kryszkiewicz, H.Rybinski}@ii.pw.edu.pl

James F. Peters
University of Manitoba
Department of Electrical and Computer Engineering, Winnipeg, Canada
E-mail: jfpeters@ee.umanitoba.ca

Andrzej Skowron
Warsaw University
Institute of Mathematics, Warsaw, Poland
E-mail: skowron@mimuw.edu.pl

Library of Congress Control Number: 2007929743

CR Subject Classification (1998): I.2, F.4.1, F.1, H.2.8, I.5.1, I.4, H.3, H.4

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-73450-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-73450-5 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12086276 06/3180 5 4 3 2 1 0

Preface

The International Conference on Rough Sets and Emerging Intelligent Systems Paradigms (RSEISP 2007) was held under the auspices of the Committee of Computer Science of the Polish Academy of Sciences. The conference was dedicated to the memory of Prof. Zdzisław Pawlak[1]. During his lifetime, the research interests and contributions of Pawlak were rich and varied.[2] His research ranged from his pioneering work on knowledge description systems and rough sets during the 1970s and 1980s as well as his work on the design of computers, information retrieval, modeling conflict analysis and negotiation, genetic grammars and molecular computing. Added to that was Pawlak's lifelong interest in painting, photography and poetry. Pawlak nurtured worldwide interest in approximation, approximate reasoning and rough set theory and its applications. Evidence of the influence of Pawlak's work can be seen in the growth in the rough set literature that now includes over 4,000 publications, as well, as in the growth and maturity of the International Rough Set Society[3], a number of international conferences dedicated to research concerning the foundations and applications of rough set theory, and the publication of seven volumes of the *Transactions on Rough Sets* journal since its inception in 2004[4].

During the past 35 years, since the introduction of knowledge description systems in the 1970s, the theory and applications of rough sets has grown rapidly. In particular, RSEISP 2007 focused on various forms of soft and granular computing such as rough and fuzzy sets, knowledge technology and discovery, data processing and mining, as well as their applications in intelligent information systems. Rough set theory proposed by Zdzisław Pawlak in 1981 provides a model for approximate reasoning. The main idea underlying this approach is to discover to what extent a given set of objects approximates another set containing objects of interest. This approach led to the discovery of affinities between

[1] Prof. Pawlak passed away on April 7, 2006.

[2] See, *e.g.*, E. Orłowska, J.F. Peters, G. Rozenberg, A. Skowron (Eds.): *New Frontiers in Scientific Discovery. Commemorating the Life and Work of Zdzisław Pawlak.* IOS Press, Amsterdam, 2007. ISBN: 978-1-58603-717-8
`http://www.iospress.nl/loadtop/load.php?isbn=9781586037178`
J.F. Peters and A. Skowron: Zdzisław Pawlak: Life and Work 1926-2006. *Transactions on Rough Sets* V, LNCS 4100 (2006) 1-24.
Additional commemorative volumes: *Transactions on Rough Sets* VI and VII, LNCS 4374 (2007) and LNCS 4400 (2007).

[3] IRSS:`http://roughsets.home.pl/www/`

[4] See ISSN: 1861-2059 (print version) and ISSN: 1861-2067 (electronic version) available from Springer at
`http://www.springer.com/west/home/computer/lncs?SGWID=4-164-6-99627-0`

objects that come to light by considering function values associated with object features or attributes. In applications, rough set methodology focuses on approximate representation of knowledge derivable from experimental data and domain knowledge. This led to many significant results in areas such as smart systems, image processing, pattern recognition, signal processing, data mining, machine learning, finance, industry, multimedia, medicine, and recently in bioinformatics and robotics.

The RSEISP 2007 Proceeding continue the tradition begun with other conferences such as Rough Sets and Knowledge Technology (RSKT 2006[5]), Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC 2005[6]), Rough Sets and Current Trends in Computing (RSCTC 2006[7]), and the Joint Rough Set Symposium (JRS 2007[8]). In particular, RSEISP 2007 introduced a number of new advances in the foundations and applications of rough sets as well as other intelligent systems paradigms. These advances have profound implications in a number of research areas such as affine description, approximate reasoning, artificial intelligence, brain informatics, bioinformatics, biology, classification of complex structured objects, computer engineering (rough set processors), data mining, data warehousing, decision systems, Dempster–Shafer theory, feature selection, feature extraction, formal concept analysis, foundations of rough sets, fuzzy logic, fuzzy sets, generalized constraint language, genetics, granulated decision systems, granular computing, granular knowledge representation, grey-rough sets, image recognition, incomplete information (missing values), information fusion, information granularity, interval calculus, knowledge consistency, knowledge discovery, map granules, medical image classification, machine learning, medicine, mereology, mining association rules, mining numerical data, music information retrieval, natural language computation, natural language engineering, neural computing, online dispute resolution, Petri net modeling, quality of service, radial basis function neural models, pattern recognition, Pawlak flow graphs, reasoning with incomplete information, reducts, rough argumentation, rough classifiers, rough inclusion, rough induction, similarity coverage model, software engineering, spam filtering, support vector machine (SVM) classifiers, text processing, universal networks, variable precision rough sets model, voice recognition, Web-based medical support systems, Web information gathering, Web intelligence, and Zadeh's calculus of linguistically quantified propositions.

A total of 161 researchers from 20 countries are represented in this volume, namely, Australia, Canada, India, P.R. China, Egypt, Finland, France, Italy, Japan, Poland, Spain, Sweden, Thailand, The Netherlands, Romania, Russia, Slovakia, Thailand, UK and USA.

We would like to dedicate this volume to the father of fuzzy set theory, Lotfi A. Zadeh, who is continuously inspiring the research of the rough set

[5] LNCS 4062 (2006).

[6] Part 1: LNCS 3641 (2005) and Part 2: LNCS 3642 (2005).

[7] LNCS 4259 (2006).

[8] 14-16 May 2007, Toronto, Canada. See `http://www.infobright.com/jrs07/`

community. Let us express our gratitude to Lotfi A. Zadeh, who kindly accepted our invitation to serve as the Honorary Chair, and to deliver the keynote talk for the conference.

We also wish to express our thanks to Jiming Liu, Sankar K. Pal and Roman Słowiński for accepting to be keynote speakers as well as Jerzy Grzymała-Busse, Victor Marek, Ryszard Michalski, Hung Son Nguyen, Ewa Orłowska, James F. Peters, Lech Polkowski, Zbigniew Raś, Jarosław Stepaniuk, Shusaku Tsumoto, YiYu Yao, Wojciech Ziarko for accepting to be plenary speakers.

Our special thanks go to members of the Organizing Committee and Program Committee of the RSEISP 2007 for their contribution to the scientific program of the conference. The high quality of the proceedings of the RSEISP 2007 Conference was made possible thanks to the reviewers as well as to the laudable efforts of many generous persons and organizations. We would also like to thank all individuals who submitted papers to the conference, and to thank the conference participants.

The organization of the conference benefitted from contributions by Piotr Gawrysiak, Łukasz Skonieczny and Robert Bembenik. We are also grateful to Bożenna Skalska, whose administrative support and cheery manner were invaluable throughout. The editors and authors of this volume also extend an expression of gratitude to Alfred Hofmann, Ursula Barth, Christine Günther and the other *Lecture Notes in Computer Science* staff at Springer for their support in making this volume possible.

June 2007

Marzena Kryszkiewicz
James F. Peters
Henryk Rybinski
Andrzej Skowron

Organization

RSEISP 2007 was organized by the Institute of Computer Science, Warsaw University of Technology.

Conference Committee

Honorary Chair	Lotfi Zadeh (University of California, Berkeley)
Conference Chairs	Marzena Kryszkiewicz (Warsaw University of Technology, Poland) Mieczysław Muraszkiewicz (Warsaw University of Technology, Poland) Henryk Rybinski (Warsaw University of Technology, Poland) Andrzej Skowron (Warsaw University, Poland)

Organizing Committee

Tadeusz Czachórski (Institute for Theoretical and Applied Informatics, Polish Academy of Sciences, Poland)
Andrzej Czyżewski (Gdansk University of Technology, Poland)
Piotr Dembiński (Institute of Computer Science, Polish Academy of Sciences, Poland)
Anna Gomolińska (Bialystok University, Poland)
Jerzy W. Grzymala-Busse (University of Kansas, USA)
Janusz Kacprzyk (Systems Research Institute, Polish Academy of Sciences, Poland)
Mieczysaw A. Kłopotek (Institute of Computer Science, Polish Academy of Sciences, Poland)
Bożena Kostek (Gdansk University of Technology, Poland)
Marzena Kryszkiewicz (Warsaw University of Technology, Poland)
Jiming Liu (University of Windsor, Canada)
Witold Łukaszewicz (University of Computer Science and Economics TWP Olsztyn, Poland)
Mieczysław Muraszkiewicz (Warsaw University of Technology, Poland)
Ewa Orłowska (National Institute of Telecommunications, Poland)
Lech Polkowski (University of Warmia and Mazury in Olsztyn, Poland)
Zbyszek Ras (University of North Carolina, Charlotte, USA)
Henryk Rybinski (Warsaw University of Technology, Poland)
Andrzej Skowron (Warsaw University, Poland)
Roman Słowiński (Poznan University of Technology, Poland)
Krzysztof Słowiński (Poznan University of Medical Science, Poland)

Jerzy Stefanowski (Poznan University of Technology, Poland)
Jaroslaw Stepaniuk (Bialystok University of Technology, Poland)
Zbigniew Suraj (Rzeszow University, Poland)
Tomasz Szmuc (AGH University of Science and Technology, Poland)
Alicja Wakulicz-Deja (University of Silesia, Poland)
Wojciech Ziarko (University of Regina, Canada)

Program Committee

Aijun An (York University, Canada)
Andrzej Bargiela (Nottingham Trent University, UK)
Jan Bazan (Rzeszow Universtity, Poland)
Cory Butz (University of Regina, Canada)
Nicholas Cercone (York University, Canada)
Martine De Cock (Ghent University, The Netherlands)
Chris Cornelis (Ghent University, The Netherlands)
Tadeusz Czachórski (Institute for Theoretical and Applied Informatics, Polish Academy of Sciences, Poland)
Andrzej Czyżewski (Gdansk University of Technology, Poland)
Alexandre Delteil (France Telecom, France)
Piotr Dembiński (Institute of Computer Science, Polish Academy of Sciences, Poland)
Bernhard Ganter (Technische Universität Dresden, Germany)
Piotr Gawrysiak (Warsaw University of Technology, Poland)
Anna Gomolińska (Bialystok University, Poland)
Jarek Gryz (York University, Canada)
Jerzy W. Grzymala-Busse (University of Kansas, USA)
Mirsad Hadzikadic (University of North Carolina at Charlotte, USA)
Aboul E. Hassanien (Cairo University, Egypt)
Gerhard Heyer (Leipzig University, Germany)
Ryszard Janicki (McMaster University, Canada)
Janusz Kacprzyk (Systems Research Institute, Polish Academy of Sciences, Poland)
Mieczysław A. Kłopotek (Institute of Computer Science, Polish Academy of Sciences, Poland)
Boena Kostek (Gdansk University of Technology, Poland)
Marzena Kryszkiewicz (Warsaw University of Technology, Poland)
Masahiro Inuiguchi (Osaka University, Japan)
T.Y. Lin (San Jose State University, USA)
Pawan Lingras (Saint Mary's University, Canada)
Jiming Liu (University of Windsor, Canada)
Tadeusz Łuba (Warsaw University of Technology, Poland)

Witold Łukaszewicz (University of Computer Science and Economics TWP Olsztyn, Poland)
Solomon Marcus (Romanian Academy, Romania)
Victor Marek (University of Kentucky, USA)
Stan Matwin (University of Ottawa, Canada)
Ernestina Menasalvas Ruiz (University of Madrid, Spain)
Wojtek Michalowski (University of Ottawa, Canada)
Ryszard Michalski (George Mason University, USA)
Tadeusz Morzy (Poznan University of Technology, Poland)
Mikhail Moshkov (University of Silesia, Poland)
Mieczysław Muraszkiewicz (Warsaw University of Technology, Poland)
Ewa Orłowska (National Institute of Telecommunications, Poland)
Andrzej Pacut (Warsaw University of Technology, Poland)
Sankar K. Pal (Indian Statistical Institute, India)
Witold Pedrycz (University of Alberta, Canada)
James F. Peters (University of Manitoba, Canada)
Lech Polkowski (University of Warmia and Mazury in Olsztyn, Poland)
Sheela Ramanna (University of Winnipeg, Canada)
Anna Radzikowska (Warsaw University of Technology, Poland)
Zbyszek Ras (University of North Carolina, at Charlotte, USA)
Kenneth Revett (University of Westminster, UK)
Henryk Rybinski (Warsaw University of Technology, Poland)
Wladyslaw Skarbek (Warsaw University of Technology, Poland)
Andrzej Skowron (Warsaw University, Poland)
Dominik Slezak (Infobright Inc., Canada)
Roman Słowiński (Poznan University of Technology, Poland)
Krzysztof Słowiński (Poznan University of Medical Science, Poland)
Nguyen Hung Son (Warsaw University, Poland)
Jerzy Stefanowski (Poznan University of Technology, Poland)
Jaroslaw Stepaniuk (Bialystok University of Technology, Poland)
Zbigniew Suraj (Rzeszow University, Poland)
Piotr Synak (Infobright Inc., Canada)
Andrzej Szalas (Linköping University, Sweden)
Tomasz Szapiro (Warsaw School of Economics, Poland)
Tomasz Szmuc (AGH University of Science and Technology, Poland)
Ryszard Tadeusiewicz (AGH University of Science and Technology, Poland)
Li-Shiang Tsay (Hampton University, USA)
Shusaku Tsumoto (Shimane University, Japan)
Dimiter Vakarelov (Sofia University, Bulgaria)
Alicja Wakulicz-Deja (University of Silesia, Poland)
Krzysztof Walczak (Warsaw University of Technology, Poland)
Guoyin Wang (Institute of Electrical and Electronics Engineers, China)
Anita Wasilewska (Stony Brook State University of NY, USA)
Arkadiusz Wojna (Warsaw University, Poland)
Jakub Wróblewski (Warsaw University, Poland)

Xindong Wu (University of Vermont, USA)
JingTao Yao (University of Regina, Canada)
Yiyu Yao (University of Regina, Canada)
Lotfi Zadeh (University of California, Berkeley, USA)
Wojciech Ziarko (University of Regina, Canada)
Ning Zhong (Maebashi Institute of Technology, Japan)

Sponsoring Institutions

AGH University of Science and Technology, Poland
Bialystok University, Poland
Bialystok University of Technology, Poland
Gdansk University of Technology, Poland
Systems Research Institute, Polish Academy of Sciences, Poland
Institute for Theoretical and Applied Informatics, Polish Academy of Sciences, Poland
Institute of Computer Science, Polish Academy of Sciences, Poland
Knowledge Technology Foundation, Poland
National Institute of Telecomunications, Poland
Polish Japanese Institute of Information Technology, Poland
University of Computer Science and Economics TWP Olsztyn, Poland
University of North Carolina, Charlotte, USA
Poznan University of Medical Sciences, Poland
Poznan University of Technology, Poland
Rzeszow University, Poland
University of Kansas, USA
University of Regina, Canada
University of Silesia, Poland
University of Warmia and Mazury in Olsztyn, Poland
Warsaw University, Poland

Table of Contents

Keynote Presentations

Invited Papers

Foundations of Rough Sets

Foundations and Applications of Fuzzy Sets

Granular Computing

Algorithmic Aspects of Rough Sets

Rough Set Applications (Invited)

Rough - Fuzzy Approach

Information Systems and Rough Sets (Invited)

Data and Text Mining

Machine Learning

Hybrid Methods and Applications

Multiagent Systems

Applications in Bioinformatics and Medicine

Multimedia Applications

Web Reasoning and Human Problem Solving (Invited)

Granular Computing and Rough Set Theory*

Lotfi A. Zadeh

Department of EECS, University of California
Berkeley, CA 94720-1776
Tel.: 510-642-4959; Fax: 510-642-1712
zadeh@eecs.berkeley.edu

To the memory of Professor Zdzisław Pawlak

Extended Abstract

Granulation plays an essential role in human cognition and has a position of centrality in both granular computing and rough set theory. Informally, granulation involves partitioning of an object into granules, with a granule being a clump of elements drawn together by indistinguishability, equivalence, similarity, proximity or functionality. For example, an interval is a granule; so is a fuzzy interval; so is a gaussian distribution; so is a cluster of points; and so is an equivalence class in rough set theory. A granular variable is a variable which takes granules as values. If G is value of X, then G is referred to as a granular value of X. If G is a singleton, then G is a singular value of X. A linguistic variable is a granular variable whose values are labeled with words drawn from a natural language. For example, if X is temperature, then 101.3 is a singular value of temperature, while “high” is a granular (linguistic) value of temperature.

Basically, granular computing is a mode of computation in which the objects of computation are granular variables. A granular value, X, may be interpreted as a representation of the state of imprecise knowledge about the true value of X. In this sense, granular computing may be viewed as a system of concepts and techniques for computing with variables whose values are either not known precisely or need not be known precisely.

A concept which serves to precisiate the concept of a granule is that of a generalized constraint. The concept of a generalized constraint is the centerpiece of granular computing.

A generalized constraint is an expression of the form X is$r$ R, where X is the constrained variable, R is the constraining relation, and r is an indexical variable which serves to identify the modality of the constraint. The principal modalities are: possibilistic ($r = blank$); veristic ($r = v$); probabilistic ($r = p$); usuality ($r = u$); random set ($r = rs$); fuzzy graph ($r = fg$); bimodal ($r = bm$); and group ($r = g$). The primary constraints are possibilistic, veristic and probabilistic. The

* Research supported in part by ONR N00014-02-1-0294, BT Grant CT1080028046, Omron Grant, Tekes Grant, Chevron Texaco Grant and the BISC Program of UC Berkeley.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 1–4, 2007.

standard constraints are bivalent possibilistic, bivalent veristic and probabilistic. Standard constraints have a position of centrality in existing scientific theories.

A generalized constraint, $GC(X)$, is open if X is a free variable, and is closed (grounded) if X is instantiated. A proposition is a closed generalized constraint. For example, "Lily is young," is a closed possibilistic constraint in which $X = Age(\text{Lily})$; $r = blank$; and $R =$ young is a fuzzy set. Unless indicated to the contrary, a generalized constraint is assumed to be closed.

A generalized constraint may be generated by combining, projecting, qualifying, propagating and counterpropagating other generalized constraints. The set of all generalized constraints together with the rules governing combination, projection, qualification, propagation and counterpropagation constitute the Generalized Constraint Language (GCL).

In granular computing, computation or equivalently deduction, is viewed as a sequence of operations involving combination, projection, qualification, propagation and counterpropagation of generalized constraints. An instance of projection is deduction of $GC(X)$ from $GC(X, Y)$; an instance of propagation is deduction of $GC(f(X))$ from $GC(X)$, where f is a function or a functional; an instance of counterpropagation is deduction of $GC(X)$ from $GC(f(X))$; an instance of combination is deduction of $GC(f(X, Y))$ from $GC(X)$ and $GC(Y)$; and an instance of qualification is computation of X is$r$ R when X is a generalized constraint. An example of probability qualification is (X is *small*) is *likely*. An example of veristic (truth) qualification is (X is *small*) is *not very true*.

The principal deduction rule in granular computing is the possibilistic extension principle: $f(X)$ is $A \longrightarrow g(X)$ is B, where A and B are fuzzy sets, and B is given by $\mu_B(v) = sup_u(\mu_A(f(u)))$, subject to $v = g(u)$. μ_A and μ_B are the membership functions of A and B, respectively.

A key idea in granular computing may be expressed as the fundamental thesis: information is expressible as a generalized constraint. The traditional view that information is statistical in nature may be viewed as a special, albeit important, case of the fundamental thesis.

A proposition is a carrier of information. As a consequence of the fundamental thesis, the meaning of a proposition is expressible as a generalized constraint. This meaning postulate serves as a bridge between granular computing and NL-Computation, that is, computation with information described in a natural language.

The point of departure in NL-Computation is (a) an input dataset which consists of a collection of propositions described in a natural language; and (b) a query, q, described in a natural language. To compute an answer to the query, the given propositions are precisiated through translation into the Generalized Constraint Language (GCL). The translates which express the meanings of given propositions are generalized constraints. Once the input dataset is expressed as a system of generalized constraints, granular computing is employed to compute the answer to the query.

As a simple illustration assume that the input dataset consists of the proposition "Most Swedes are tall," and the query is "What is the average height of

Swedes?" Let h be the height density function, meaning that $h(u)du$ is the fraction of Swedes whose height lies in the interval $[u, u+du]$. The given proposition "Most Swedes are tall," translates into a generalized constraint on h, and so does the translate of the query "What is the average height of Swedes?" Employing the extension principle, the generalized constraint on h propagates to a generalized constraint on the answer to q. Computation of the answer to q reduces to solution of a variational problem. A concomitant of the close relationship between granular computing and NL-Computation is a close relationship between granular computing and the computational theory of perceptions. More specifically, a natural language may be viewed as a system for describing perceptions. This observation suggests a way of computing with perceptions by reducing the problem of computation with perceptions to that of computation with their natural language descriptions, that is, to NL-Computation. In turn, NL-Computation is reduced to granular computing through translation/precisiation into the Generalized Constraint Language (GCL).

An interesting application of the relationship between granular computing and the computational theory of perceptions involves what may be called perception-based arithmetic. In this arithmetic, the objects of arithmetic operations are perceptions of numbers rather than numbers themselves. More specifically, a perception of a number, a, is expressed as *usually* $({}^*a)$, where ${}^*a$ denotes "*approximately a.*" For concreteness, ${}^*a$ is defined as a fuzzy interval centering on a, and *usually* is defined as a fuzzy probability. In this setting, a basic question is: What is the sum of *usually* $({}^*a)$ and *usually* $({}^*b)$? Granular computing and, more particularly, granular arithmetic, provide a machinery for dealing with questions of this type.

Granular computing is based on fuzzy logic. Fuzzy logic has endured many years of skepticism and derision largely because fuzziness is a word with pejorative connotations. Today, fuzzy logic is used in a wide variety of products and systems ranging from digital cameras, home appliances and medical instrumentation to automobiles, elevators, subway trains, paper making machinery and traffic control systems. By this measure, fuzzy logic has achieved success.

There are two basic rationales which underlie the success of fuzzy logic. Indirectly, the same rationales apply to granular computing and rough set theory. The second rationale is referred to as "The fuzzy logic gambit." To understand the rationales it is necessary to differentiate between two meanings of precision: precision in value, v-precision; and precision in meaning, m-precision. For example, if X is a real-valued variable, then the proposition X is in the interval $[a, b]$, where a and b are precisely defined numbers, is v-imprecise and m-precise. Additionally, we have to differentiate between mh-precisiation, that is, human-oriented m-precisiation, and mm-precisiation, that is, machine-oriented m-precisiation. For example, a dictionary definition of stability may be viewed as an instance of mh-precisiation, while Lyapunov's definition of stability is an instance of mm-precisiation of stability. Furthermore, v-imprecisiation may be imperative (forced) or intentional (deliberate). For example, if I do not know Lily's age and describe her as young, then v-imprecisiation is imperative (forced). If I know her birthday but choose to describe her age as young, then v-imprecisiation is intentional (deliberate).

Let X be a variable taking values in U. U may be a space of numbers, functions, relations, distributions, etc. Consider two cases.

Case 1: Values of X are not known precisely, i.e., X is v-imprecise, denoted as $^*X$.

Case 2: Values of X are known precisely, i.e., X is v-precise.

In *Case* l, I have some information, $Inf(^*X)$, about values of $^*X$. I mm-precisiate $Inf(^*X)$ by using an information description language, IDL. IDL may be the language of bivalent logic and probability theory, BL + PT; or the language of fuzzy logic, FL; or a natural language, NL. NL may be mm-precisiated through translation into FL. FL is a superlanguage of (BL + PT) in the sense that it has a much higher expressive power than (BL + PT).

In *Case* 1, the use of FL as the information description language serves to enhance the accuracy of description of values of *X, especially when $^*X$ takes values in the space of functions, relations or distributions. This is Rationale 1 for the use of fuzzy logic as an information description language when the values of $^*X$ are not known precisely.

Turning to *Case* 2, we observe that, in general, precision carries a cost. If there is a tolerance for imprecision, we can exploit it by sacrificing precision through v-imprecisiation of X. This is what we do when we perform data compression, summarization and other information-reduction operations. More generally, we v-imprecisiate X to $^*X$ to reduce cost. By so doing, we reduce *Case* 2 to *Case* 1. Then we mm-precisiate $^*X$ through the use of NL as an information description language. This is the essence of Rationale 2 for the use of fuzzy logic when the values of a variable are known precisely. In this context, the fuzzy logic gambit may be stated as:

If there is a tolerance for imprecision, exploit it through v-imprecisiation followed by mm-precisiation.

The fuzzy logic gambit is Rationale 2 for the use of fuzzy logic when the values of a variable are known precisely.

It is of historical interest to note that my 1965 paper “Fuzzy sets” was motivated by Rationale l. My 1973 paper, “Outline of a new approach to the analysis of complex systems and decision processes,” was motivated by Rationale 2. Today, most applications of fuzzy logic employ the concepts of a linguistic variable and fuzzy if-then rule sets – concepts which were introduced in the 1973 paper.

Imprecision, uncertainty and partiality of truth are pervasive characteristics of the real world. As we move further into the age of machine intelligence and automated reasoning, the need for an enhancement of our ability to deal with imprecision, uncertainty and partiality of truth is certain to grow in visibility and importance. It is this need that motivated the genesis of granular computing and rough set theory, and is driving their progress. In coming years, granular computing, rough set theory and NL-Computation are likely to become a part of the mainstream of computation and machine intelligence.

Dominance-Based Rough Set Approach to Reasoning About Ordinal Data

Roman Słowiński[1], Salvatore Greco[2], and Benedetto Matarazzo[2]

[1] Institute of Computing Science, Poznań University of Technology,
60-965 Poznań, and Institute for Systems Research,
Polish Academy of Sciences, 01-447 Warsaw, Poland
roman.slowinski@cs.put.poznan.pl

[2] Faculty of Economics, University of Catania,
Corso Italia, 55, 95129 – Catania, Italy
{salgreco,matarazz}@unict.it

Abstract. Dominance-based Rough Set Approach (DRSA) has been proposed by the authors to handle background knowledge about ordinal evaluations of objects from a universe, and about monotonic relationships between these evaluations, e.g. "the larger the mass and the smaller the distance, the larger the gravity" or "the greater the debt of a firm, the greater its risk of failure". Such a knowledge is typical for data describing various phenomena, and for data concerning multiple criteria decision making or decision under uncertainty. It appears that the Indiscernibility-based Rough Set Approach (IRSA) proposed by Pawlak involves a primitive idea of monotonicity related to a scale with only two values: "presence" and "absence" of a property. This is why IRSA can be considered as a particular case of DRSA. Monotonicity gains importance when the binary scale, including only "presence" and "absence" of a property, becomes finer and permits to express the presence of a property to certain degree. This observation leads to very natural fuzzy generalization of the rough set concept via DRSA. It exploits only ordinal properties of membership degrees and monotonic relationships between them, without using any fuzzy connective. We show, moreover, that this generalization is a natural continuation of the ideas given by Leibniz, Frege, Boole, Łukasiewicz and Pawlak. Finally, the fuzzy rough approximations taking into account monotonic relationships between memberships to different sets can be applied to case-based reasoning. In this perspective, we propose to consider monotonicity of the type: "the more similar is y to x, the more credible is that y belongs to the same set as x".

Keywords: Rough sets, Ordinal data, Dominance-based Rough Set Approach, Decision support, Granular computing, Fuzzy rough sets, Case-based reasoning.

1 Sketch of the Presentation

By this presentation, we wish to pay tribute to late Zdzisław Pawlak who introduced us to his philosophy of reasoning about data, which appeared to have so great potential in decision support.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 5–11, 2007.

According to Pawlak [21], rough set theory refers to some ideas of Gottfried Leibniz (indiscernibility), Gottlob Frege (vague concepts), George Boole (reasoning methods), Jan Łukasiewicz (multi-valued logics), and Thomas Bayes (inductive reasoning).

Referring to these ideas, we represent fundamental concepts of rough set theory in terms of a generalization, called *Dominance-based Rough set Approach* (DRSA), that permits to deal with *ordinal data.* DRSA have been proposed by the authors (see e.g. [10,12,14,15,16,23]) to take into account ordinal properties of data related to preferences. We show that DRSA is also relevant in case where preferences are not considered but a kind of monotonicity relating attribute values is meaningful for the analysis of data at hand. In general, monotonicity concerns relationship between different aspects of a phenomenon described by data, e.g.: "the larger the house, the higher its price" or "the more a tomato is red, the more it is ripe". The qualifiers, like "large house", "high price", "red" and "ripe", may be expressed either in terms of some measurement units, or in terms of degrees of membership to some fuzzy sets. In this perspective, the DRSA gives a very general framework in which the classical Indiscernibility-based Rough Set Approach (IRSA) can be considered as a particular case [19].

Looking at DRSA from *granular computing* perspective, we can say that DRSA permits to deal with *ordered data* by considering a specific type of information granules defined by means of *dominance based constraints* having a syntax of the type: "x is at least R" or "x is at most R", where R is a qualifier from a properly ordered scale. In evaluation space, such granules are *dominance cones.* In this sense, the contribution of DRSA consists in:

- extending the paradigm of granular computing to problems involving ordered data,
- specifying a proper syntax and modality of information granules (the dominance based constraints which should be adjoined to other modalities of information constraints, such as possibilistic, veristic and probabilistic [24]),
- defining a methodology dealing properly with this type of information granules, and resulting in a theory of computing with words and reasoning about data in case of ordered data.

Let us observe that other modalities of information constraints, such as veristic, possibilistic and probabilistic, have also to deal with ordered values (with qualifiers relative to grades of truth, possibility and probability). We believe, therefore, that granular computing with ordered data and DRSA as a proper way of reasoning about ordered data, are very important in the future development of the whole domain of granular computing.

DRSA can be applied straightforward to multiple criteria *classification* (called also *sorting*) problems. The data contain in this case the preference information in form of a finite set of classification examples provided by the decision maker. Note that, while multiple criteria classification is based on absolute evaluation of objects, multiple criteria *choice* and *ranking* refer to pairwise comparisons of objects. These pairwise comparisons are in this case the preference information provided by the decision maker. The decision rules to be discovered from the pairwise comparisons characterize a comprehensive preference relation on the set of objects. In consequence, the preference model of the decision maker is a *set of decision rules*. It may be used to *explain* the decision policy of the decision maker and to *recommend* a good choice or preference ranking with respect to new objects [5].

In [13] we opened a new avenue for applications of the rough set concept to analysis of preference-ordered data. We considered the classical problem of *decision under uncertainty* extending DRSA by using *stochastic dominance*. We considered the case of traditional additive probability distribution over the set of future states of the world; however, the model is rich enough to handle non-additive probability distributions and even qualitative ordinal distributions. The rough set approach gives a representation of DM's preferences under uncertainty in terms of "*if..., then...*" decision rules induced from rough approximations of sets of exemplary decisions (preference-ordered classification of acts described in terms of outcomes in uncertain states of the world). This extension is interesting with respect to multicriteria decision analysis from two different points of view:

- each decision under uncertainty can be viewed as a multicriteria decision, where criteria are outcomes in different states of the world;
- DRSA adapted to decision under uncertainty can be applied to deal with multicriteria decision under uncertainty, i.e. a decision problem where in each future state of the world the outcomes are expressed in terms of a set of criteria.

Even if DRSA has been proposed to deal with ordinal properties of data related to preferences in decision problems, the concept of dominance-based rough approximation can be used in a much more general context [17]. This is because the *monotonicity*, which is crucial for DRSA, is also meaningful for problems where preferences are not considered. Monotonicity is a property translating in a formal language a primitive intuition of relationship between different concepts of our knowledge.

In IRSA, the idea of monotonicity is not evident, although it is also present there. Because of very coarse representation of considered concepts, monotonicity is taken into account in the sense of "presence" or "absence" of particular aspects characterizing the concepts. This is why IRSA can be considered as a particular case of DRSA.

Monotonicity gains importance when the binary scale, including only "presence" and "absence" of an aspect, becomes finer and permits to consider the presence of a property to a certain degree. Due to *graduality*, the idea of monotonicity can be exploited in the whole range of its potential. Graduality is typical for fuzzy set philosophy and thus, a joint consideration of rough sets and fuzzy sets is worthwhile. In fact, rough sets and fuzzy sets capture the two basic complementary aspects of monotonicity: rough sets deal with relationships between different concepts, and fuzzy sets deal with expression of different dimensions in which the concepts are considered. For this reason, many approaches have been proposed to combine fuzzy sets with rough sets (see e.g. [1,2,4,22]).

The main preoccupation in almost all the studies combining rough sets with fuzzy sets was related to a fuzzy extension of Pawlak's definition of lower and upper approximations using *fuzzy connectives* (t-norm, t-conorm, fuzzy implication). DRSA can also be combined with fuzzy sets along this line, obtaining a rough set model permitting to deal with fuzziness in preference representation [10,11,7]. Let us remark, however, that in fact there is no rule for the choice of the "right" fuzzy connective, so this choice is always arbitrary to some extent. Moreover, there is another drawback for fuzzy extensions of rough sets involving fuzzy connectives: they are based on cardinal properties of membership degrees. In consequence, the result of these extensions is sensitive to order preserving transformation of membership degrees.

The DRSA approach proposed in [8,9] for a fuzzy extension of rough sets avoids arbitrary choice of fuzzy connectives and not meaningful operations on membership degrees. It exploits only ordinal character of the membership degrees and proposes a methodology of fuzzy rough approximation that infers the most cautious conclusion from available imprecise information. In particular, any approximation of knowledge about Y using knowledge about X is based on positive or negative relationships between premises and conclusions, i.e.:

i) "the more x is X, the more it is Y" (positive relationship),
ii) "the more x is X, the less it is Y" (negative relationship).

These relationships have the form of *gradual decision rules*. Examples of these decision rules are:

"if a car is speedy with credibility at least 0.8 and it has high fuel consumption with credibility at most 0.7, then it is a good car with a credibility at least 0.9",

and

"if a car is speedy with credibility at most 0.5 and it has high fuel consumption with credibility at least 0.8, then it is a good car with a credibility at most 0.6".

Remark that the syntax of gradual decision rules is based on monotonic relationship between degrees of credibility that can also be found in dominance-based decision rules induced from preference-ordered data. This explains why one can build a fuzzy rough approximation using DRSA.

Finally, the fuzzy rough approximation taking into account monotonic relationships can be applied to *case-based reasoning* [18]. Case-based reasoning regards the inference of some proper conclusions related to a new situation by the analysis of similar cases from a memory of previous cases. It is based on two principles :

i) similar problems have similar solutions,
ii) types of encountered problems tend to recur.

Gilboa and Schmeidler [6] observed that the basic idea of case-based reasoning can be found in the following sentence of Hume [20]: "From causes which appear *similar* we expect similar effects. This is the sum of all our experimental conclusions." Rephrasing Hume, one can say that "the more similar are the causes, the more similar one expects the effects."

In this perspective, we propose to consider monotonicity of the type "the more similar is y to x, the more credible is that y belongs to the same set as x". Application of DRSA in this context leads to decision rules similar to the gradual decision rules:

"the more object z is similar to a referent object x w.r.t. condition attribute s, the more z is similar to a referent object x w.r.t. decision attribute t",

or, equivalently, but more technically,

$$s(z,x) \geq \alpha \Rightarrow t(z,x) \geq \alpha$$

where functions s and t measure the credibility of similarity with respect to condition attribute and decision attribute, respectively. When there are multiple condition and decision attributes, functions s and t aggregate similarity with respect to these attributes.

Measuring similarity is the essential point of all case-based reasoning and, particularly, of fuzzy set approach to case-based reasoning [3]. This explains the many problems that measuring similarity generates within case-based reasoning. Problems of modelling similarity are relative to two levels:

- at the level of similarity with respect to single features: how to define a meaningful similarity measure with respect to a single feature?
- at the level of similarity with respect to all features: how to properly aggregate the similarity measure with respect to single features in order to obtain a comprehensive similarity measure?

Our DRSA approach to case-based reasoning tries to be possibly "neutral" and "objective" with respect to similarity relation. At the level of similarity concerning single features, we consider only ordinal properties of similarity, and at the level of aggregation, we do not impose any particular functional aggregation (involving operators, like weighted L_p norms, min, etc.) based on some very specific axioms (see, for example, [6]), but we consider a set of decision rules based on the general monotonicity property of comprehensive similarity with respect to similarity of single features. Moreover, the decision rules we propose permit to consider different thresholds for degrees of credibility in the premise and in the conclusion.

Therefore, our approach to case-based reasoning is very little "invasive", comparing to the many other existing approaches.

References

1. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. Internat. J. General Systems 17(2-3), 191–209 (1990)
2. Dubois, D., Prade, H.: Putting rough sets and fuzzy sets together. In: Słowińsk, R. (ed.) Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, pp. 203–232. Kluwer, Dordrecht (1992)

3. Dubois, D., Prade, H., Esteva, F., Garcia, P., Godo, L., Lopez de Mantara, R.: Fuzzy Set Modelling in Case-based Reasoning. International Journal of Intelligent Systems 13, 345–373 (1998)
4. Dubois, D., Grzymala-Busse, J., Inuiguchi, M., Polkowski, L.: Transactions on Rough Sets II. LNCS, vol. 3135. Springer, Berlin (2004)
5. Fortemps, Ph., Greco, S., Słowiński, R.: Multicriteria decision support using rules that represent rough-graded preference relations, European Journal of Operational Research (to appear 2007)
6. Gilboa, I., Schmeidler, D.: A Theory of Case-Based Decisions. Cambridge University Press, Cambridge (2001)
7. Greco, S., Inuiguchi, M., Słowiński, R.: Dominance-based rough set approach using possibility and necessity measures. In: Alpigini, J.J., Peters, J.F., Skowron, A., Zhong, N. (eds.) RSCTC 2002. LNCS (LNAI), vol. 2475, pp. 85–92. Springer, Berlin (2002)
8. Greco, S., Inuiguchi, M., Słowiński, R.: A new proposal for rough fuzzy approximations and decision rule representation. In: Dubois, D., Grzymala-Busse, J., Inuiguchi, M., Polkowski, L. (eds.) Transactions on Rough Sets II. LNCS, vol. 3135, pp. 156–164. Springer, Berlin (2004)
9. Greco, S., Inuiguchi, M., Słowiński, R.: Fuzzy rough sets and multiple-premise gradual decision rules. International Journal of Approximate Reasoning 41, 179–211 (2006)
10. Greco, S., Matarazzo, B., Słowiński, R.: The use of rough sets and fuzzy sets in MCDM, chapter 14. In: Gal, T., Stewart, T., Hanne, T. (eds.): Advances in Multiple Criteria Decision Making, pp. 14.1–14.59, Kluwer Academic Publishers, Boston (1999)
11. Greco, S., Matarazzo, B., Słowiński, R.: A fuzzy extension of the rough set approach to multicriteria and multiattribute sorting. In: Fodor, J., De Baets, B., Perny, P. (eds.) Preferences and Decisions under Incomplete Information, pp. 131–154. Physica-Verlag, Heidelberg (2000)
12. Greco, S., Matarazzo, B., Słowiński, R.: Rough sets theory for multicriteria decision analysis. European Journal of Operational Research 129, 1–47 (2001)
13. Greco, S., Matarazzo, B., Słowiński, R.: Rough set approach to decisions under risk. In: Ziarko, W., Yao, Y. (eds.) RSCTC 2000. LNCS (LNAI), vol. 2005, pp. 160–169. Springer, Berlin (2001)
14. Greco, S., Matarazzo, B., Słowiński, R.: Dominance-Based Rough Set Approach to Knowledge Discovery (I) - General Perspective, chapter 20. In: Zhong, N., Liu, J. (eds.) Intelligent Technologies for Information Analysis, pp. 513–552. Springer, Berlin (2004)
15. Greco, S., Matarazzo, B., Słowiński, R.: Dominance-Based Rough Set Approach to Knowledge Discovery (II) - Extensions and Applications, chapter 21. In: Zhong, N., Liu, J. (eds.) Intelligent Technologies for Information Analysis, pp. 553–612. Springer, Berlin (2004)
16. Greco, S., Matarazzo, B., Słowiński, R.: Decision rule approach, chapter 13. In: Figueira, J., Greco, S., Ehrgott, M. (eds.) Multiple Criteria Decision Analysis: State of the Art Surveys, pp. 507–563. Springer, Berlin (2005)
17. Greco, S., Matarazzo, B., Słowiński, R.: Generalizing rough set theory through Dominance-based Rough Set Approach. In: Slezak, D., Yao, J., Peters, J., Ziarko, W., Hu, X. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3642, pp. 1–11. Springer, Berlin (2005)

18. Greco, S., Matarazzo, B., Słowiński, R.: Dominance-based Rough Set Approach to Case-Based Reasoning. In: Torra, V., Narukawa, Y., Valls, A., Domingo-Ferrer, J. (eds.) MDAI 2006. LNCS (LNAI), vol. 3885, pp. 7–18. Springer, Berlin (2006)
19. Greco, S., Matarazzo, B., Słowiński, R.: Dominance-based Rough Set Approach as a proper way of handling graduality in rough set theory. In: Transactions on Rough Sets VII. LNCS, vol. 4400, pp. 36–52. Springer, Berlin (2007)
20. Hume, D.: An Enquiry Concerning Human Understanding. Clarendon Press, Oxford, 1748
21. Pawlak, Z.: Rough Set Theory. Kunstliche Intelligenz 3, 38–39 (2001)
22. Radzikowska, A.M., Kerre, E.E.: A comparative study of fuzzy rough sets. Fuzzy Sets and Systems 126, 137–155 (2002)
23. Słowiński, R., Greco, S., Matarazzo, B.: Rough set based decision support, chapter 16. In: Burke, E.K., Kendall, G. (eds.) Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, pp. 475–527. Springer, New York (2005)
24. Zadeh, L.: From computing with numbers to computing with words – from manipulation of measurements to manipulation of perception. IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications 45, 105–119 (1999)

Mining Numerical Data—A Rough Set Approach

Jerzy W. Grzymala-Busse

[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
http://lightning.eecs.ku.edu/index.html

Abstract. We present an approach to mining numerical data based on rough set theory using calculus of attribute-value blocks. An algorithm implementing these ideas, called MLEM2, induces high quality rules in terms of both simplicity (number of rules and total number of conditions) and accuracy. Additionally, MLEM2 induces rules not only from complete data sets but also from data with missing attribute values, with or without numerical attributes.

1 Introduction

For knowledge acquisition (or data mining) from data with numerical attributes special techniques are applied [13]. Most frequently, an additional step, taken before the main step of rule induction or decision tree generation and called *discretization* is used. In this preliminary step numerical data are converted into symbolic or, more precisely, a domain of the numerical attribute is partitioned into intervals. Many discretization techniques, using principles such as equal interval frequency, equal interval width, minimal class entropy, minimum description length, clustering, etc., were explored, e.g., in [1,2,3,5,6,8,9,10,19,21,22,23,24,27]. Note that discretization used as preprocessing and based on clustering is superior to other preprocessing techniques of this type [8].

Discretization algorithms which operate on the set of all attributes and which do not use information about decision (concept membership) are called *unsupervised*, as opposed to *supervised*, where the decision is taken into account [9]. Methods processing the entire attribute set are called *global*, while methods working on one attribute at a time are called *local* [8]. In all of these methods discretization is a preprocessing step and is undertaken before the main process of knowledge acquisition.

Another possibility is to discretize numerical attributes during the process of knowledge acquisition. Examples of such methods are MLEM2 [14] and MODLEM [20,29,30] for rule induction and C4.5 [28] and CART [4] for decision tree generation. These algorithms deal with original, numerical data and the process of knowledge acquisition and discretization are conducted at the same time. The

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 12–21, 2007.

MLEM2 algorithm produces better rule sets, in terms of both simplicity and accuracy, than clustering methods [15]. However, discretization is an art rather than a science, and for a specific data set it is advantageous to use as many discretization algorithms as possible and then select the best approach.

In this paper we will present the MLEM2 algorithm, one of the most successful approaches to mining numerical data. This algorithm uses rough set theory and calculus of attribute-value pair blocks. A similar approach is represented by MODLEM. Both MLEM2 and MODLEM algorithms are outgrowths of the LEM2 algorithm. However, in MODLEM the most essential part of selecting the best attribute-value pair is conducted using entropy or Laplacian conditions, while in MLEM2 this selection uses the most relevance condition, just like in the original LEM2.

2 MLEM2

The MLEM2 algorithm is a part of the LERS (Learning from Examples based on Rough Sets) data mining system. Rough set theory was initiated by Z. Pawlak [25,26]. LERS uses two different approaches to rule induction: one is used in machine learning, the other in knowledge acquisition. In machine learning, or more specifically, in learning from examples (cases), the usual task is to learn the smallest set of minimal rules, describing the concept. To accomplish this goal, LERS uses two algorithms: LEM1 and LEM2 (LEM1 and LEM2 stand for Learning from Examples Module, version 1 and 2, respectively) [7,11,12].

The LEM2 algorithm is based on an idea of an attribute-value pair block. For an attribute-value pair $(a,\ v) = t$, a *block* of t, denoted by $[t]$, is a set of all cases from U such that for attribute a have value v. For a set T of attribute-value pairs, the intersection of blocks for all t from T will be denoted by $[T]$. Let B be a nonempty lower or upper approximation of a concept represented by a decision-value pair (d, w). Set B *depends* on a set T of attribute-value pairs $t = (a, v)$ if and only if

$$\emptyset \neq [T] = \bigcap_{t \in T} [t] \subseteq B.$$

Set T is a *minimal complex* of B if and only if B depends on T and no proper subset T' of T exists such that B depends on T'. Let $\mathcal{T}$ be a nonempty collection of nonempty sets of attribute-value pairs. Then $\mathcal{T}$ is a *local covering* of B if and only if the following conditions are satisfied:

- each member T of $\mathcal{T}$ is a minimal complex of B,
- $\bigcap_{t \in \mathcal{T}} [T] = B$, and
- $\mathcal{T}$ is minimal, i.e., $\mathcal{T}$ has the smallest possible number of members.

The user may select an option of LEM2 with or without taking into account attribute priorities. The procedure LEM2 with attribute priorities is presented below. The option without taking into account priorities differs from the one presented below in the selection of a pair $t \in T(G)$ in the inner loop WHILE. When LEM2 is not to take attribute priorities into account, the first criterion is ignored. In our experiments all attribute priorities were equal to each other.

```
Procedure LEM2
(input: a set $B$,
output: a single local covering $\mathcal{T}$ of set $B$);
begin
    $G := B$;
    $\mathcal{T} := \emptyset$;
    while $G \neq \emptyset$
        begin
        $T := \emptyset$;
        $T(G) := \{t | [t] \cap G \neq \emptyset\}$ ;
        while $T = \emptyset$ or $[T] \not\subseteq B$
            begin
                select a pair $t \in T(G)$ with the highest
                attribute priority, if a tie occurs, select a pair
                $t \in T(G)$ such that $|[t] \cap G|$ is maximum;
                if another tie occurs, select a pair $t \in T(G)$
                with the smallest cardinality of $[t]$;
                if a further tie occurs, select first pair;
                $T := T \cup \{t\}$ ;
                $G := [t] \cap G$ ;
                $T(G) := \{t | [t] \cap G \neq \emptyset\}$;
                $T(G) := T(G) - T$ ;
                end {while}
            for each $t \in T$ do
                if $[T - \{t\}] \subseteq$ B then $T := T - \{t\}$;
            $\mathcal{T} := \mathcal{T} \cup \{T\}$;
            $G := B - \bigcup_{T \in \mathcal{T}} [T]$;
    end {while};
    for each $T \in \mathcal{T}$ do
        if $\bigcup_{S \in \mathcal{T} - \{T\}} [S] = B$ then $\mathcal{T} := \mathcal{T} - \{T\}$;
end {procedure}.
```

For a set X, $|X|$ denotes the cardinality of X.

Rules induced from raw, training data are used for classification of unseen, testing data. The classification system of LERS is a modification of the *bucket brigade algorithm*. The decision to which concept a case belongs is made on the basis of three factors: strength, specificity, and support. They are defined as follows: *Strength* is the total number of cases correctly classified by the rule during training. *Specificity* is the total number of attribute-value pairs on the left-hand side of the rule. The matching rules with a larger number of attribute-value pairs are considered more specific. The third factor, *support*, is defined as the sum of scores of all matching rules from the concept. The concept C for which the support (i.e., the sum of all products of strength and specificity, for all rules matching the case, is the largest is a winner and the case is classified as being a member of C).

MLEM2, a modified version of LEM2, categorizes all attributes into two categories: numerical attributes and symbolic attributes. For numerical attributes MLEM2 computes blocks in a different way than for symbolic attributes. First, it sorts all values of a numerical attribute. Then it computes cutpoints as averages for any two consecutive values of the sorted list. For each cutpoint x MLEM2 creates two blocks, the first block contains all cases for which values of the numerical attribute are smaller than x, the second block contains remaining cases, i.e., all cases for which values of the numerical attribute are larger than x. The search space of MLEM2 is the set of all blocks computed this way, together with blocks defined by symbolic attributes. Starting from that point, rule induction in MLEM2 is conducted the same way as in LEM2.

Let us illustrate the MLEM2 algorithm using the following example from Table 1.

Table 1. An example of the decision table

	Attributes		Decision
Case	Gender	Cholesterol	Stroke
1	man	180	no
2	man	240	yes
3	man	280	yes
4	woman	240	no
5	woman	280	no
6	woman	320	yes

Rows of the decision table represent *cases*, while columns are labeled by *variables*. The set of all cases will be denoted by U. In Table 1, $U = \{1, 2, ..., 6\}$. Independent variables are called *attributes* and a dependent variable is called a *decision* and is denoted by d. The set of all attributes will be denoted by A. In Table 1, $A = \{Gender, Cholesterol\}$. Any decision table defines a function ρ that maps the direct product of U and A into the set of all values. For example, in Table 1, $\rho(1, Gender) = man$. The decision table from Table 1 is *consistent*, i.e., there are no conflicting cases in which all attribute values are identical yet the decision values are different. Subsets of U with the same decision value are called *concepts*. In Table 1 there are two concepts: $\{1, 4, 5\}$ and $\{2, 3, 6\}$.

Table 1 contains one numerical attribute (*Cholesterol*). The sorted list of values of *Cholesterol* is 180, 240, 280, 320. The corresponding cutpoints are: 210, 260, 300.

Since our decision table is consistent, input sets to be applied to MLEM2 are concepts. The search space for MLEM2 is the set of all blocks for all possible attribute-value pairs $(a, v) = t$. For Table 1, the set of all attribute-value pair blocks are

$[(Gender, man)] = \{1, 2, 3\}$,
$[(Gender, woman)] = \{4, 5, 6\}$,
$[(Cholesterol, 180..210)] = \{1\}$,
$[(Cholesterol, 210..320)] = \{2, 3, 4, 5, 6\}$,
$[(Cholesterol, 180..260)] = \{1, 2, 4\}$,
$[(Cholesterol, 260..320)] = \{3, 5, 6\}$,
$[(Cholesterol, 180..300)] = \{1, 2, 3, 4, 5\}$,
$[(Cholesterol, 300..320)] = \{6\}$.

Let us start running MLEM2 for the concept {1, 4, 5}. Thus, initially this concept is equal to B (and to G). The set $T(G)$ is equal to {(Gender, man), (Gender, woman), (Cholesterol, 180..210), (Cholesterol, 210..320), (Cholesterol, 180..260), (Cholesterol, 260..320), (Cholesterol, 180..300)}.

For the attribute-value pair (Cholesterol, 180..300) from $T(G)$ the following value $|[(attribute, value)] \cap G|$ is maximum. Thus we select our first attribute-value pair $t =$ (Cholesterol, 180..300). Since $[(Cholesterol, 180..300)] \not\subseteq B$, we have to perform the next iteration of the inner WHILE loop. This time $T(G) =$ {(Gender, man), (Gender, woman), (Cholesterol, 180..210), (Cholesterol, 210..320), (Cholesterol, 180..260), (Cholesterol, 260..320)}. For three attribute-value pairs from $T(G)$: (Gender, woman), (Cholesterol, 210..320) and (Cholesterol, 180..260) the value of $|[(attribute, value)] \cap G|$ is maximum (and equal to two). The second criterion, the smallest cardinality of [(attribute, value)], indicates (Gender, woman) and (Cholesterol, 180..260) (in both cases that cardinality is equal to three). The last criterion, "first pair", selects (Gender, woman). Moreover, the new $T =$ {(Cholesterol, 180..300), (Gender, woman)} and new G is equal to {4, 5}. Since $[T] = [(Cholesterol, 180..260] \cap [(Gender, woman)] = \{4, 5\} \subseteq B$, the first minimal complex is computed.

Furthermore, we cannot drop any of these two attribute-value pairs, so $\mathcal{T} = \{T\}$, and the new G is equal to $B - \{4, 5\} = \{1\}$.

During the second iteration of the outer WHILE loop, the next minimal complex T is identified as {(Cholesterol, 180..210)}, so $\mathcal{T} =$ {{(Cholesterol, 180..300), (Gender, woman)}, {(Cholesterol, 180..210)}} and $G = \emptyset$.

The remaining rule set, for the concept {2, 3, 6} is induced in a similar manner. Eventually, rules in the LERS format (every rule is equipped with three numbers, the total number of attribute-value pairs on the left-hand side of the rule, the total number of examples correctly classified by the rule during training, and the total number of training cases matching the left-hand side of the rule) are:

```
2, 2, 2
(Gender, woman) & (Cholesterol, 180..300) -> (Stroke, no)
1, 1, 1
(Cholesterol, 180..210) -> (Stroke, no)
```

2, 2, 2
(Gender, man) & (Cholesterol, 210..320) -> (Stroke, yes)
1, 1, 1
(Cholesterol, 300..320) -> (Stroke, yes)

3 Numerical and Incomplete Data

Input data for data mining are frequently affected by missing attribute values. In other words, the corresponding function ρ is incompletely specified (partial). A decision table with an incompletely specified function ρ will be called *incompletely specified*, or *incomplete*.

Though four different interpretations of missing attribute values were studied [18]; in this paper, for simplicity, we will consider only two: lost values (the values that were recorded but currently are unavailable) and ”do not care” conditions (the original values were irrelevant).

For the rest of the paper we will assume that all decision values are specified, i.e., they are not missing. Also, we will assume that all missing attribute values are denoted either by ”?” or by ”$*$”, lost values will be denoted by ”?”, ”do not care” conditions will be denoted by ”$*$”. Additionally, we will assume that for each case at least one attribute value is specified.

Incomplete decision tables are described by characteristic relations instead of indiscernibility relations. Also, elementary blocks are replaced by characteristic sets, see, e.g., [16,17,18]. An example of an incomplete table is presented in Table 2.

Table 2. An example of the incomplete decision table

	Attributes		Decision
Case	Gender	Cholesterol	Stroke
1	?	180	no
2	man	$*$	yes
3	man	280	yes
4	woman	240	no
5	woman	?	no
6	woman	320	yes

For incomplete decision tables the definition of a block of an attribute-value pair must be modified. If for an attribute a there exists a case x such that $\rho(x, a) =?$, i.e., the corresponding value is lost, then the case x is not included in the block $[(a, v)]$ for any value v of attribute a. If for an attribute a there exists a case x such that the corresponding value is a ”do not care” condition, i.e., $\rho(x, a) = *$, then the corresponding case x should be included in blocks

$[(a, v)]$ for all values v of attribute a. This modification of the definition of the block of attribute-value pair is consistent with the interpretation of missing attribute values, lost and "do not care" condition. Numerical attributes should be treated in a little bit different way as symbolic attributes. First, for computing characteristic sets, numerical attributes should be considered as symbolic. For example, for Table 2 the blocks of attribute-value pairs are:

$[(Gender, man)] = \{2, 3\}$,
$[(Gender, woman)] = \{4, 5, 6\}$,
$[(Cholesterol, 180)] = \{1, 2\}$,
$[(Cholesterol, 240)] = \{2, 4\}$,
$[(Cholesterol, 280)] = \{2, 3\}$,
$[(Cholesterol, 320)] = \{2, 6\}$.

The *characteristic set* $K_B(x)$ is the intersection of blocks of attribute-value pairs (a, v) for all attributes a from B for which $\rho(x, a)$ is specified and $\rho(x, a) = v$. The characteristic sets $K_B(x)$ for Table 2 and $B = A$ are:

$K_A(1) = U \cap \{1, 2\} = \{1, 2\}$,
$K_A(2) = \{2, 3\} \cap U = \{2, 3\}$,
$K_A(3) = \{2, 3\} \cap \{2, 3\} = \{2, 3\}$,
$K_A(4) = \{4, 5, 6\} \cap \{2, 4\} = \{4\}$,
$K_A(5) = \{4, 5, 6\} \cap U = \{4, 5, 6\}$,
$K_A(6) = \{4, 5, 6\} \cap \{2, 6\} = \{6\}$.

For incompletely specified decision tables lower and upper approximations may be defined in a few different ways [16,17,18]. We will quote only one type of approximations for incomplete decision tables, called concept approximations. A *concept* B-lower approximation of the concept X is defined as follows:

$$\underline{B}X = \cup\{K_B(x) | x \in X, K_B(x) \subseteq X\}.$$

A concept B-upper approximation of the concept X is defined as follows:

$$\overline{B}X = \cup\{K_B(x) | x \in X, K_B(x) \cap X \neq \emptyset\} = \cup\{K_B(x) | x \in X\}.$$

For Table 2, concept lower and upper approximations are:

$$\underline{A}\{1, 4, 5\} = \{4\},$$

$$\underline{A}\{2, 3, 6\} = \{2, 3, 6\},$$

$$\overline{A}\{1, 4, 5\} = \{1, 2, 4, 5, 6\},$$

$$\overline{A}\{2, 3, 6\} = \{2, 3, 6\}.$$

For inducing rules from data with numerical attributes, blocks of attribute-value pairs are defined differently than in computing characteristic sets. Blocks of attribute-value pairs for numerical attributes are computed in a similar way as for complete data, but for every cutpoint the corresponding blocks are computed taking into account interpretation of missing attribute values. Thus,

$[(Gender, man)] = \{1, 2\}$,
$[(Gender, woman)] = \{4, 5, 6\}$,
$[(Cholesterol, 180..210)] = \{1, 2\}$,
$[(Cholesterol, 210..320)] = \{2, 3, 4, 6\}$,
$[(Cholesterol, 180..260)] = \{1, 2, 4\}$,
$[(Cholesterol, 260..320)] = \{2, 3, 6\}$,
$[(Cholesterol, 180..300)] = \{1, 2, 3, 4\}$,
$[(Cholesterol, 300..320)] = \{2, 6\}$.

Using the MLEM2 algorithm, the following rules are induced:

certain rule set (induced from the concept lower approximations):

```
2, 1, 1
(Gender, woman) & (Cholesterol, 180..260) -> (Stroke, no)
1, 3, 3
(Cholesterol, 260..320) -> (Stroke, yes)
```

possible rule set (induced from the concept upper approximations):

```
1, 2, 3
(Gender, woman) -> (Stroke, no)
1, 1, 3
(Cholesterol, 180..260) -> (Stroke, no)
1, 3, 3
(Cholesterol, 260..320) -> (Stroke, yes)
```

4 Conclusions

We demonstrated that both rough set theory and calculus of attribute-value pair blocks are useful tools for data mining from numerical data. The same idea of an attribute-value pair block may be used in the process of data mining not only for computing elementary sets (for complete data sets) but also for rule induction. The MLEM2 algorithm induces rules from raw data with numerical attributes, without any prior discretization, and MLEM2 provides the same results as LEM2 for data with all symbolic attributes. Additionally, experimental results show that rule induction based on MLEM2 is one of the best approaches to data mining from numerical data [15].

References

1. Bajcar, S., Grzymala-Busse, J.W., Hippe, Z.S.: A comparison of six discretization algorithms used for prediction of melanoma. In: Proc. of the Eleventh International Symposium on Intelligent Information Systems, IIS'2002, Sopot, Poland, 2002, pp. 3–12. Physica-Verlag, Heidelberg (2003)

2. Bay, S.D.: Multivariate discretization of continous variables for set mining. In: Proc. of the 6-th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Boston, MA, 2000, pp. 315–319 (2000)
3. Biba, M., Esposito, F., Ferilli, S., Mauro, N.D., Basile, T.M.A.: Unsupervised discretization using kernel density estimation. In: Proc. of the 20-th Int. Conf. on AI, Hyderabad, India, 2007, pp. 696–701 (2007)
4. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth & Brooks, Monterey CA (1984)
5. Catlett, J.: On changing continuous attributes into ordered discrete attributes. In: EWSL-91, Proc. of the European Working Session on Learning, Porto, Portugal, March 1991. LNCS (LNAI), pp. 164–178. Springer, Berlin (1991)
6. Chan, C.C., Batur, C., Srinivasan, A.: Determination of quantization intervals in rule based model for dynamic systems. In: Proc. of the IEEE Conference on Systems, Man, and Cybernetics, Charlottesville, VA, 1991, pp. 1719–1723 (1991)
7. 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, December 1991, p. 20 (1991)
8. 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)
9. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: Proc of the 12-th Int. Conf. on Machine Learning, Tahoe City, CA, July 9–12, 1995, pp. 194–202 (1995)
10. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proc. of the 13th Int. Joint Conference on AI, Chambery, France, 1993, pp. 1022–1027 (1993)
11. Grzymala-Busse, J.W.: LERS—A system for learning from examples based on rough sets. In: Slowinski, R. (ed.) Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory, pp. 3–18. Kluwer Academic Publishers, Dordrecht (1992)
12. Grzymala-Busse, J.W.: A new version of the rule induction system LERS. Fundamenta Informaticae 31, 27–39 (1997)
13. 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)
14. Grzymala-Busse, J.W.: MLEM2: A new algorithm for rule induction from imperfect data. In: Proc. of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2002, Annecy, France, 2002, pp. 243–250 (2002)
15. Grzymala-Busse, J.W.: A comparison of three strategies to rule induction from data with numerical attributes. In: Proc. of the Int. Workshop on Rough Sets in Knowledge Discovery (RSKD 2003), in conjunction with the European Joint Conferences on Theory and Practice of Software, Warsaw, 2003, pp. 132–140 (2003)
16. Grzymala-Busse, J.W.: Rough set strategies to data with missing attribute values. Workshop Notes, Foundations and New Directions of Data Mining. In: conjunction with the 3-rd International Conference on Data Mining, Melbourne, FL, 2003, pp. 56–63 (2003)
17. Grzymala-Busse, J.W.: Data with missing attribute values: Generalization of idiscernibility relation and rule induction. In: Transactions on Rough Sets. Lecture Notes in Computer Science Journal Subline, vol. 1, pp. 78–95. Springer, Heidelberg (2004)

18. Grzymala-Busse, J.W.: Incomplete data and generalization of indiscernibility relation, definability, and approximations. In: Proc. of the RSFDGrC'2005, the Tenth International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, pp. 244–253. Springer, Regina, Canada (2005)
19. Grzymala-Busse, J.W., Stefanowski, J.: Discretization of numerical attributes by direct use of the rule induction algorithm LEM2 with interval extension. In: Proc. of the Sixth Symposium on Intelligent Information Systems (IIS'97), Zakopane, Poland, 1997, pp. 149–158 (1997)
20. Grzymala-Busse, J.W., Stefanowski, J.: Three discretization methods for rule induction. Int. Journal of Intelligent Systems 16, 29–38 (2001)
21. Kerber, R.: ChiMerge: Discretization of numeric attributes. In: Proc. of the 10th National Conf. on AI, San Jose, CA, 1992, pp. 123–128 (1992)
22. Kohavi, R., Sahami, M.: Error-based and entropy-based discretization of continous features. In: Proc of the 2-nd Int. Conf. on Knowledge Discovery and Data Mining, Portland, OR, 1996, pp. 114–119 (1996)
23. Liu, H., Hussain, F., Tan, C.L., Dash, M.: Discretization: An enabling technique. Data. Mining and Knowledge Discovery 6, 393–423 (2002)
24. 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)
25. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
26. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
27. 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 4-th ACM SIGKDD Workshop on Data Mining in Bioinformatics, 2004, pp. 24–30 (2004)
28. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA (1993)
29. Stefanowski, J.: Handling continuous attributes in discovery of strong decision rules. In: Proc. of the 1-st Int. Conference on Rough Sets and Current Trends in Computing, Warsaw, pp. 394–401. Springer, Berlin (1998)
30. Stefanowski, J.: Algorithms of Decision Rule Induction in Data Mining. Poznan University of Technology Press, Poznan, Poland (2001)

Rough Sets and Approximation Schemes

Victor W. Marek and Mirosław Truszczynski

Department of Computer Science
University of Kentucky
Lexington, KY 40506-0046, USA

Abstract. Approximate reasoning is used in a variety of reasoning tasks in Logic-based Artificial Intelligence. In this abstract we compare a number of such reasoning schemes and show how they relate and differ from the approach of Pawlak's Rough Sets.

1 Introduction

Humans reason more often than not with incomplete information. The effect is that the conclusions must often be revised, and treated as approximate. Frequently we face the following situation: some features of objects of interest are firmly established (based on observations and on domain properties), some other are known to be false. But there remains a "grey area" of features of objects of interest that are not determined by the current knowledge. In this note we discuss several schemes that have been proposed in the literature for handling approximate reasoning when available knowledge may be incomplete. They include rough sets [Paw82], approximation for propositional satisfiability [KS96], approximation semantics for logic programs including brave and skeptical answer-set semantics, Kripke-Kleene semantics and well-founded semantics [Kun87, Fit85], the semantics of repairs in databases [ABC03], knowledge compilation of propositional theories [KS96], and least- and largest- pair of fixpoints for the operator associated with a Horn program [Ll87]. For some of these, we will be able to show that they fit into the rough set paradigm.

2 Approximations and Three-Valued Reasoning Schemes

We discuss here a variety of approximating schemes. They all have a common feature – they use a three-valued approach to sets of objects.

2.1 Approximations and the Ordering $\preceq_{kn}$

Given a set (universe) U, an approximation over U is any pair of subsets of U, X_1, X_2 such that $X_1 \subseteq X_2$. An approximation $\langle X_1, X_2 \rangle$ provides bounds on every set X such that $X_1 \subseteq X \subseteq X_2$. The Kleene (or *knowledge*) ordering of approximations [Kl67] is defined as follows:

$$\langle X_1, X_2 \rangle \preceq_{kn} \langle Y_1, Y_2 \rangle \text{ if } X_1 \subseteq Y_1 \text{ and } Y_2 \subseteq X_2.$$

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 22–28, 2007.

Let $\mathcal{A}_U$ be the set of all approximations in U. The structure $\langle \mathcal{A}_U, \preceq_{kn} \rangle$ is a chain-complete poset. Unless $|X| \leq 1$, this poset is not a lattice. It is a complete lower-semilattice, and the least upper bound exists for any pair of approximations that have an upper bound. The maximal elements of $\langle \mathcal{A}_U \preceq_{kn} \rangle$ are of the form $\langle X, X \rangle$ for $X \subseteq U$. They are called *exact approximations*.

2.2 Rough Sets

Rough sets are special class of approximations. Let O be a finite set of objects (universe). Every equivalence relation r in U determines its concept of rough set as follows. For every $X \subseteq O$, *Pawlak's approximation* (or the *rough set* associated with X) is defined as an approximation $\langle \underline{X}, \overline{X} \rangle$ where: $\underline{X}$ is the union of all cosets of r contained in X, and $\overline{X}$ is the union of all cosets of r that have a nonempty intersection with X. The pair $\langle \underline{X}, \overline{X} \rangle$ is an approximation in O. It is characterized [MT99] as the $\preceq_{kn}$-largest approximation $\langle L, U \rangle$ so that:

1. $\langle L, U \rangle$ approximates X
2. The sets L and U are unions of cosets of r.

As each equivalence relation in O determines its own class of rough sets, the question arises how these classes are related. The collection of equivalence relations on a set O (not necessarily finite) determines a complete, but non-distributive lattice, with the refinement ordering $\sqsubseteq$. Specifically, $r_1 \sqsubseteq r_2$ if every coset of r_1 is the union of cosets of r_2. Let $r_1 \sqsubseteq r_2$ be two equivalence relations in O. One can show that for every subset X of O the Pawlak rough sets determined by r_1 and r_2, say $\langle \underline{X}_1, \overline{X} \rangle_1$ and $\langle \underline{X}_1, \overline{X} \rangle_1$, respectively, are related as follows:

$$\langle \underline{X}_1, \overline{X}_1 \rangle \preceq_{kn} \langle \underline{X}_2, \overline{X}_2 \rangle.$$

In other words, the ordering $\sqsubseteq$ in the lattice of equivalence relations on O induces the ordering $\preceq_{kn}$ in the corresponding Pawlak approximations.

2.3 Propositional Satisfiability

We consider a fixed set of propositional variables At. A valuation of At is any mapping of At into $\{0, 1\}$. We can identify valuations with the subsets of At as follows. We identify a valuation v with the set $M \subseteq At$ so that $v = \chi_M$, that is, $M = \{p : v(p) = 1\}$. We write v_M for the valuation v that corresponds to M.

Now, let T be a consistent set of formulas of the propositional language $\mathcal{L}_{At}$. Then T determines an approximation $\langle X_1, X_2 \rangle$ in set At as follows: $X_1 = \{p : T \vdash p\}$, and $X_2 = \{p : T \not\vdash \neg p\}$. Then $X_1 \subseteq M \subseteq X_2$ for every M such that $v_M \models T$. Let us denote this "canonical" approximation of models of T by $\langle \underline{T}, \overline{T} \rangle$. Then, we have the following property of theories $T_1 \subseteq T_2$ that are consistent and closed under consequence:

$$\langle \underline{T}_1, \overline{T}_1 \rangle \preceq_{kn} \langle \underline{T}_2, \overline{T}_2 \rangle.$$

In other words, the canonical approximation of the theory T_2 is $\preceq_{kn}$ bigger than that of T_1. The maximal approximations (i.e. Pawlak's rough sets in this case) are the complete consistent theories.

2.4 Knowledge Compilation

Many tasks in knowledge representation and reasoning reduce to the problem of deciding, given a propositional CNF theory T and a propositional clause φ, whether $T \models \varphi$. This task is coNP-complete. As a way to address this computational difficulty [KS96] proposed an approach in which T is compiled off-line, possibly in exponential time, into some other representation, under which the query answering would be efficient. While there is an initial expense of the compilation, if the query answering task is frequent that cost will eventually be recuperated.

An approximation to a theory T is a pair of theories (T', T') such that

$$T' \models T \models T''.$$

If (T', T'') is an approximation to T, then $T \models \varphi$ if $T'' \models \varphi$, and $T \not\models \varphi$ if $T' \not\models \varphi$. In other words,

$$\{\varphi \colon T'' \models \varphi\} \subseteq \{\varphi \colon T \models \varphi\} \subseteq \{\varphi \colon T' \models \varphi\}.$$

Desirable approximations are "tight", that is, $\{\varphi \colon T' \models \varphi\} \setminus \{\varphi \colon T'' \models \varphi\}$ is small, and support efficient reasoning. Concerning the latter point, if U is a Horn theory and φ is a clause, then $U \models \varphi$ can be decided in polynomial time. Therefore, we define *approximations* to be pairs (T', T''), where T' and T'' are Horn theories such that $T' \models T''$.

A key problem is: given a CNF theory T, find the most precise Horn approximation to T. This problem has been studied in [KS96]. It turns out that there is a unique (up to logical equivalence) Horn least upper bound. However, there is no greatest Horn upper bound. The set of Horn lower approximations has, however, maximal elements.

2.5 Approximating Semantics for Logic Programs

Logic Programming studies semantics of *logic programs*, i.e. sets of *program clauses*. In the simplest case those are expressions of the form $p \leftarrow q_1, \ldots, q_m, \neg r_1, \ldots, \neg r_n$. The meaning of such clause is, informally, this: "if $q_1, \ldots, q_m$ have been derived, and none of $r_1, \ldots, r_n$ has, or ever will be, then derive p" (various different meanings are also associated with program clauses). It is currently commonly assumed that the correct semantics of a logic program (i.e. set of program clauses as above) is provided by means of fixpoints of the Gelfond-Lifschitz operator GL_P. Those fixpoints are called *stable models* of P [GL88], and more recently also *answer sets* for P. The operator GL_P is antimonotone, thus existence of fixpoints of GL_P is not guaranteed. However the operator GL_P^2 is monotone, and thus possesses a least and largest fixpoints.

A number of approximation schemes for stable semantics of logic programs has been proposed. The earliest proposal is the so-called Kripke-Kleene approximation ([Kun87, Fit85]). In this approach, one defines a *three-valued* van-Emden-Kowalski operator $\mathcal{T}_P$. That operator is monotone in the ordering $\preceq_{kn}$, and thus possesses a least $\preceq_{kn}$ fixpoint. That fixpoint (which can be treated as an approximation) approximates all stable models of the logic program P. A stronger approximation scheme has been proposed in [VRS91], and is called a *well-founded model* of the program. Essentially, that model is defined by means of the least and largest fixpoint of GL_P^2. Like the Kripke-Kleene fixpoint, the well-founded approximations provides an approximation to all stable models of the program. Yet another approximation scheme which turns out to be stricter than the well-founded semantics is the *ultimate approximation* of [DMT04].

Of course, one can assign to a logic program P the $\preceq_{kn}$-largest approximation for the family of all stable models of P. Let us denote by KK_P the Kripke-Kleene approximation, WF_P the well-founded approximation, U_P, the ultimate approximation and A_P the most precise approximation of all stable models of P. Then, assuming P possesses a stable model, we have

$$KK_P \preceq_{kn} WF_P \preceq_{kn} U_P \preceq_{kn} A_P$$

and examples can be given where all the relationships are strict. The complexity of computing each of these approximations is also different, in general. Nevertheless, these constructions assign, on analogy to rough sets, approximations to programs. Thus, in case of Logic Programming approximations there exist a classification of approximations to the family of all stable models of the program.

We note the the Kripke-Kleene approximation KK_P approximates not only all stable models of P but also all supported models of P. In the case when P is a Horn program the fixpoint KK_P is given by the pair (S_l, S_u), where S_l is the least and S_u is the greatest supported model of P (which are guaranteed to exist).

2.6 Approximating Possible-World Structures

The language of modal logic with the semantics of autoepistemic expansions and extensions [DMT03] provides a way to describe approximations to possible-world structures. Let us consider a theory T in a language of propositional modal logic. The theory T is meant to describe a possible world structure providing the account of what is known and what is not known given T.

Since T may be incomplete, there may be several possible-world structures one could associate with T (autoepistemic logic provides a specific characterization of such structures; other nonmonotonic modal logics could be used, too [MT93]). To reason about the epistemic content of T one has two choices: to compute all possible-world structures for T according to the semantics of the autoepistemic logic, or compute an approximation to the epistemic content of T common to all these structures. The former is computationally complex, being a Σ_P^2-task. Hence, the latter is often the method of choice.

At least three different approximations can be associated with T, Kripke-Kleene approximation, the well-founded approximation and the ultimate approximation, listed here according to the precision, with which they approximate possible-world structures of T [DMT03, DMT04]. It is worth noting that the computational complexity of each of these approximations is lower that the complexity of computing even a (single) possible-world structure for T.

2.7 Minimal Models Reasoning and Repairs in Databases

Approximations play an important role in the theory and practice of databases. In this paper, we regard a database as a finite structure of some language $\mathcal{L}$ of first-order logic that does not contain function symbols. Typically, legal databases are subject to *integrity constraints*, properties that at any time the database is supposed to have. In general, integrity constraints can be represented as arbitrary formulas of $\mathcal{L}$.

Databases are frequently modified over their lifetime. Updates create the possibility of entering erroneous data, especially that in most cases databases are modified by different users at different locations. Consequently, it does happen that databases do not satisfy the integrity constraints. Once such a situation occurs, the database needs to be *repaired* [ABC03].

Let D be a database and let IC be a set of integrity constraints. A pair $R = (R^+, R^-)$ is a *repair* of D with respect to IC if $(D \cup R^+) \setminus R^- \models IC$ (the repair condition), and for every (Q^+, Q^-) such that $Q^+ \subseteq R^+$, $Q^- \subseteq R^-$, and $(D \cup Q^+) \setminus Q^- \models IC$, we have $Q^+ = R^+$ and $Q^- = R^-$ (the minimality condition). We write $R(D)$ for the database $(D \cup R^+) \setminus R^-$ resulting from D by applying a repair R. We write $Rep(D, IC)$ to denote all repairs of D with respect to IC. The minimality condition implies that if (R^+, R^-) is a repair, then $R^+ \cap D = \emptyset$ and $R^- \subseteq D$.

Repairing a database D that violates its integrity constraints IC consists of computing a repair $R \in Rep(D, IC)$ and applying it to D, that is computing $R(D)$. There are two problems, though. First, computing repairs is computationally complex (even in some simple settings deciding whether repairs exist is Σ^2_P-complete). Second, it often is the case that multiple repairs exist, which results in the need for some principled selection strategy.

These problems can be circumvented to some degree by modifying the semantics of the database. Namely, a database D with integrity constraints IC could be viewed as an *approximation* to an actual database D', not available explicitly but obtainable from D by means of a repair with respect to IC. The approximation to D' represented by (D, IC) is the pair of sets (D_l, D_u), where

$$D_l = \bigcap\{R(D)\colon\ R \in Rep(D, IC)\} \ \text{ and } \ D_l = \bigcup\{R(D)\colon\ R \in Rep(D, IC)\}.$$

In other words, expressions (D, IC) define approximations, and query answering algorithms have to be adjusted to provide best possible answers to queries to D' based on the knowledge of D_l and D_u only.

3 Further Work, and Conclusions

We discussed a number of approximation schemes as they appear in logic, logic programming, artificial intelligence, and databases. Doubtless there are other approaches to approximate reasoning that can be cast as approximations, and in particular rough sets. One wonders if there is a classification of approximations that allows to capture a common structure laying behind these, formally different, approaches. In other words, are there general classification principles for approximations? Are there categories of approximations that allow to classify approximations qualitatively?

Another fundamental issue is the use of languages that describe approximations. Pawlak [Paw91] noticed that, in its most abstract form, rough sets are associated with equivalence relations; each equivalence relation induces its own rough set notion. Such abstract approach leads to Universal Algebra considerations that have roots in [JT51] and have been actively pursued by Orłowska and collaborators [DO01, OS01, SI98]. One can find even more abstract versions within the Category Theory. But usually, the applications of rough sets and other approximation schemes cannot choose its own language. For instance, more often than not (and this was the original motivation of Pawlak) the underlying equivalence relation is given to the application (for instance as the equivalence induced by an information system [MP76]). Then, and the literature of rough sets is full of such considerations, one searches for the coarser equivalence relations that are generated by various attribute reduction techniques. To make the point, these equivalence relations are not arbitrary, but determined by the choice of the language used for data description. This linguistic aspect of rough sets and approximations in general, needs more attention of rough set community.

Acknowledgments

This research was supported by the National Science Foundation under Grant IIS-0325063 and by the Kentucky Science and Engineering Foundation under Grant 1036-RDE-008.

References

[ABC03] Arenas, M., Bertossi, L.E., Chomicki, J.: Answer sets for consistent query answering in inconsistent databases. Theory and Practice of Logic Programming 3(4-5), 393–424 (2003)

[DP92] Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order. Cambridge University Press, Cambridge (1992)

[DMT03] Denecker, M., Marek, V., Truszczyński, M.: Uniform semantic treatment of default and autoepistemic logics. Artificial Intelligence Journal 143, 79–122 (2003)

[DMT04] Denecker, M., Marek, V., Truszczyński, M.: Ultimate approximation and its application in nonmonotonic knowledge representation systems. Information and Computation 192, 84–121 (2004)

[DO01] Düntsch, I., Orłowska, E.: Beyond Modalities: Sufficiency and Mixed Algebras. Chapter 16 of [OS01] (2001)

[Fit85] Fitting, M.C.: A Kripke-Kleene semantics for logic programs. Journal of Logic Programming 2(4), 295–312 (1985)

[GL88] Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proceedings. of the International Joint Conference and Symposium on Logic Programming, pp. 1070–1080. MIT Press, Cambridge (1988)

[Jo91] Jonsson, B.: A Survey of Boolean Algebras with Oprators. In: Algebras and Order, pp. 239–284. Kluwer, Dordrecht (1991)

[JT51] Jonsson, B., Tarski, A.: Boolean Algebras with Operators. American Journal of Mathematics 73, 891–939 (1951)

[Kl67] Kleene, S.C. Kleene, S.C.: Introduction to Metamathematics. North-Holland, Fifth reprint (1967)

[Kun87] Kunen, K.: Negation in logic programming. Journal of Logic Programming 4(4), 289–308 (1987)

[Ll87] Lloyd, J.W.: Foundations of Logic Programming. Springer, Heidelberg (1987)

[MP76] Marek, W., Pawlak, Z.: Information storage and retrieval systems, mathematical foundations. Theoretical Computer Science 1(4), 331–354 (1976)

[MP84] Marek, W., Pawlak, Z.: Rough sets and information systems. Fundamenta Informaticae 7(1), 105–115 (1984)

[MT93] Marek, V.W., Truszczyński, M.: Nonmonotonic Logic; Context-Dependent Reasoning. Springer, Berlin (1993)

[MT99] Marek, V.W., Truszczynski, M.: Contributions to the Theory of Rough Sets. Fundamenta Informaticae 39(4), 389–409 (1999)

[OS01] Orłowska, E., Szałas, A.: Relational Methods for Computer Science Applications. In: Selected Papers from 4th International Seminar on Relational Methods in Logic, Algebra and Computer Science (RelMiCS'98), Studies in Fuzziness and Soft Computing, vol. 65, Physica-Verlag/Springer, Heidelberg (2001)

[Paw82] Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)

[Paw91] Pawlak, Z.: Rough Sets – theoretical aspects of reasoning about data. Kluwer, Dordrecht (1991)

[SI98] SanJuan, E., Iturrioz, L.: Duality and informational representability of some information algebras. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery, Methodology and Applications, pp. 233–247. Physica-Verlag, Heidelberg (1998)

[KS96] Selman, B., Kautz, H.: Knowledge Compilation and Theory Approximation. Journal of the ACM 43(2), 193–224 (1996)

[VRS91] Van Gelder, A., Ross, K.A., Schlipf, J.S.: The well-founded semantics for general logic programs. Journal of the ACM 38(3), 620–650 (1991)

Generalizing Data in Natural Language

Ryszard S. Michalski[1,2] and Janusz Wojtusiak[1]

[1] Machine Learning and Inference Laboratory,
George Mason University, Fairfax, VA 22030, USA
[2] Institute of Computer Science,
Polish Academy of Science, Warsaw, Poland

Abstract. This paper concerns the development of a new direction in machine learning, called *natural induction*, which requires from computer-generated knowledge not only to have high predictive accuracy, but also to be in human-oriented forms, such as natural language descriptions and/or graphical representations. Such forms facilitate understanding and acceptance of the learned knowledge, and making mental models that are useful for decision making. An initial version of the AQ21-NI program for natural induction and its several novel features are briefly described. The performance of the program is illustrated by an example of deriving medical diagnostic rules from micro-array data.

1 Introduction

Most of machine learning research has been striving for achieving high predictive accuracy of knowledge learned from data, but has not paid much attention to the understandability and interpretability of that knowledge. This is evidenced by the fact that research papers on different learning methods, including learning decision trees, random forests, decision rules, ensemblies, neural nets, support vector machines, etc., typically list only predictive accuracies obtained by the reported and compared methods (e.g., [1]), but very rarely present actual knowledge learned.

While predictive accuracy of inductively acquired knowledge is obviously important, for many applications it is imperative that computer-generated knowledge is in the forms facilitating its understanding and making mental models of it by an expert. Such fields include, for example, medicine, bioinformatics, agriculture, social sciences, economy, business, archeology, defense, and others. Although the need for understandability of computer-generated knowledge has been indicated for a long time (e.g., [7], [13]), research on this topic has been inadequate. The main reason for this situation may be that understandability and interpretability of knowledge is subjective and difficult to measure.

This paper concerns the development of a new direction in machine learning, called *natural induction*, which strives to achieve high understandability and interpretability of computer-generated knowledge by learning it and presenting it in the forms resembling those in which people represent knowledge. Such forms include natural language descriptions and simple graphical representations. To serve this objective, we employed *attributional calculus* [8] as a

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 29–39, 2007.

logic and knowledge representation for learning. Attributional calculus combines selected features of propositional, predicate and multi-valued logics, and introduces several new constructs formalizing relevant features of natural language. We developed algorithms for learning attributional rules with these constructs, and also for transforming these rules into simple natural language descriptions. These algorithms have been implemented in the AQ21-NI program, briefly, NI, whose selected features are described in this paper.

2 Brief Overview of Natural Induction

The natural induction methodology for learning natural language descriptions from data involves three stages of processing. The first stage induces formal rules in attributional calculus. Such rules are more expressive than standard decision rules in which conditions are limited to ¡attribute relation value¿ forms and are also closer to equivalent natural language descriptions. The second stage transforms learned attributional rules into logically equivalent and grammatically correct natural language descriptions. The third stage employs cognitive constraints and relevant background knowledge to improve the descriptions' interpretability and to derive additional implications from them that are useful for decision making. This paper concerns the first two stages. The third stage is under development.

Let us start by briefly characterizing the general task addressed by the first stage. The goal of this stage is to take a set of data points (training examples) that exemplify decision classes C_1,...,C_k, and relevant background knowledge, and induce hypotheses, H_1,..., H_k that generally describe these classes and optimize a multi-criterion measure of of description quality. In the method implemented in the AQ21-NI program, the generated hypotheses are different forms of attributional rules. Adopting formalism presented in [8], the basic form of an attributional rule is:

$$CONSEQUENT <= PREMISE \tag{1}$$

where CONSEQUENT and PREMISE are conjunctions of attributional conditions, that are formal equivalents of simple natural language statements. Here is an example of a basic attributional rule:

```
[Task_to_do = run_experiments]
    <= [Day = weekday] & [#tissue-samples-to-analyze  = 2..7] &
       [Tests-to-perform: PAP & estradiol_level] &
       [available-lab =lab1 v lab3]
```

The second stage transforms the learned attributional rules into equivalent and grammatically correct natural language descriptions. For example, the above rule is translated to the following natural language description:

”The task is to run experiments, if the day is weekday, the number of tissue samples to analyze is between 2 and 7, the tests to perform are PAP and estradiol level, and the available lab is lab1 or lab3.”

As one can notice, the above natural language description closely corresponds to the attributional rule from which it was derived. The "weekday" is a program-abstracted value of the structured attribute "day" (the domain of a structured attribute is a hierarchy). The attribute "tests-to-perform" is a *compound attribute* whose legal values are *internal conjunctions* of values of constituent attributes (an internal conjunction binds atoms rather than statements).

As this example shows, attributional rules significantly extend standard decision rules whose conditions are limited to the form [ATTR REL VAL], where ATTR is a single attribute, REL is =, ≤ or ≥ and VAL is an attribute value. It addition to constructs presented in this example, attributional rules may involve also conditions with *count attribute* that counts the number of statements that are true in a given set, or counts the number properties satisfying a given condition. Expressions with count attributes can be viewed as a special case of statements in the second order predicate calculus (Section 2.4). Attributional rules may also include *exception clause* (Section 2.2), and several other forms that resemble those used by people in natural language descriptions (e.g., *provided-that clause*).

2.1 The Q(w) Criterion of Description Optimality

The first stage integrates the well-known separate and conquer algorithm AQ for learning consistent and complete rules (e.g., [6] or [7]), with an algorithm for discovering patterns from data, and with procedures for learning new constructs briefly mentioned above. These constructs are described in more detail in Sections 2.2 to 2.5. The implementation of this stage is based on the AQ21 learning and pattern discovery system [20] that enhances the basic AQ-type learning method by a number of new features.

Among the new features is the ability of the program to work in either Theory Formation (TF) or Pattern Discovery (PD) mode, which is controlled by the "mode" parameter. The TF mode can generate several different types of optimized complete and consistent descriptions of training examples, such as attributional rules without or with exception clauses (Section 2.2). The rules optimize a user-defined multi-criterion measure of rule optimality, LEF (e.g., [6] or [7]).

The PD mode searches for patterns or approximate descriptions that maximize a description optimality criterion defined as:

$$Q(R, w) = cov^{w} * config^{1-w} \tag{2}$$

where cov=p/P and config=((p / (p + n)) - (P /(P + N))) * (P +N) /N are measures of coverage and confidence gain, respectively, of the rule R, and w is a user-controlled parameter. Here, p and n are the numbers of positive and negative examples covered by R, and P and N are the numbers of positive and negative examples in training dataset, respectively. The confidence gain captures the increase of confidence in the rule in relation to confidence in decisions made according to their prior probabilities. As one can see, the criterion Q allows trading inconsistency (n ≠ 0) for an increase in rule coverage.

2.2 Learning Descriptions with Exception Clauses

Exceptions are commonly used by humans when describing rarely occurring anomalies that are inconsistent with a given rule or a theory. It is not unusual that an approximate description of observations can be very simple, but a perfect description, fully consistent with all observations, would be significantly more complex. In such cases, it may be useful to learn rules with exception clauses, also called censored rules (e.g., [10], [8], [17]). AQ21-NI can be set to learn censored rules in the form:

$$CONSEQUENT <= PREMISE \mid\_ \; EXCEPTION \tag{3}$$

where |_ is an exception operator, and EXCEPTION is either a conjunctive attributional description (an exception clause) or a list of examples constituting exceptions to the basic rule. The rule is read: If PREMISE is true then assert CONSEQUENT, except when EXCEPTION is true.

In PD mode, where inconsistency is allowed, the program learns basic rules that maximize Q(R,w), and then adds to them exception clauses. The latter are generated by re-applying the AQ algorithm to the examples covered by the rule to describe negative examples covered by this rule. In TF mode, where consistency has to be guaranteed, learning censored rules first involves creating basic rules and an exception list for each of them. Such a list contains examples that would introduce a significant complexity, if the rule was transferred into an expression consisting of fully consistent rules, and only if the number of examples on the exception list is significantly smaller than the number of positive examples the rule covers. If all of the exceptions on a list can be characterized by one conjunctive statement, then it is used as the EXCEPTION clause; otherwise, EXCEPTION is an explicit list of exceptions.

To illustrate differences between basic and censored rules, let us consider a simple problem of learning descriptions for "friendly" robots from their examples and counter examples. When asked to produce basic rules, AQ21-NI created two rules (the premise of each rule is preceded by <=).

```
[Robot=friendly]
   <= [Holding=book: 4,4] &
      [Size=small..medium: 6,6]: p=4,n=0
   <= [Holding=book v flag: 6,8] &
      [~Antennas: 3,9] &
      [Size=small..medium: 6,6]: p=3,n=0
```

The numbers inside conditions, after a ":", represent their positive and negative coverage of the condition, respectively; parameters p and n after each rule represent the number of positive and negative examples covered by the rule, respectively.

When asked to produce censored rules, the program generated a single rule that also covers completely and consistently the training examples:

```
[Robot=friendly]
    <= [Holding=book v flag: 6,8] & [Size=small..medium: 6,6]
    |_ [Holding=flag: 2,3] & [Antennas:1,5]: p=6,n=0
```

When transformed into an equivalent natural language expression, the above censored rule becomes:

"A robot is friendly, if it is holding a book or a balloon, and its size is between small and medium, inclusively, except when it is holding a flag and has antennas."

Note that the numbers of positive examples covered by conditions in the EXCEPTION clause are significantly smaller than those covered by the PREMISE, and the numbers of negative examples covered exceed the numbers of negative examples covered by rule conditions.

2.3 Learning Descriptions with Compound Attributes

One of the novel features of AQ21-NI is that it implements compound attributes that facilitate learning natural language descriptions of objects, or their components that require different attributes to describe them. Consider, for example, a description of weather in the style of standard propositional logic:

```
[Windy=yes] & [Cloudy=yes] & [Humid=not]
```

Using a compound attribute, such a description would be expressed as:

```
[Weather: windy & cloudy & not humid]
```

that resembles the equivalent natural language statement: "Weather is windy, cloudy and not humid." In this example, "weather" is a compound attribute, and windy, cloudy, and humid are values of its constituent attributes [8]. Learning expressions with compound attributes is done by learning rules using constituent attributes, and then transforming appropriate groups of attributes into compound forms.

2.4 Learning Descriptions with Counting Attributes

In some applications, in particular, in medicine, it is not unusual that a medical decision (e.g., diagnosis) is made on the basis of counting of number features, (e.g., symptoms), observed in the patient, and comparing it with a threshold. If the number of symptoms exceeds the threshold, the disease is implicated. To illustrate such a case by a real world example, consider a problem of classifying the severity of prostate cancers in terms of three known risk factors:

Factor 1: PSA $\geq$ 10 ng/ml ("PSA" measures the amount of prostate specific antigen)
Factor 2: Gleason's score ≥ 7 ("Gleason's score" measures the cancer cells' abnormality)
Factor 3: Stage $\geq$ T2b ("Stage" measures the level of disease development).

Based on these factors, prostate cancer patients are classified into four categories, representing an increasing severity of their disease:

Category is 1, if no factors are present; Category is 2, if one factor is present; Category is 3, if two factors are present; Category is 4, if all three factors are present.

This classification was obtained from Dr. P. Koutrovalis, Director of URO-Radiology Prostate Institute in Washington, D.C. Using attributional rules, the above classification schema can be represented by four attributional rules:

[Category = 1] <= [count(PSA $\geq$ 10 ng/ml, Gleason's $\geq$ 7, Stage $\geq$ T2b) = 0]
[Category = 2] <= [count(PSA $\geq$ 10 ng/ml, Gleason's $\geq$ 7, Stage $\geq$ T2b) = 1]
[Category = 3] <= [count(PSA $\geq$ 10 ng/ml, Gleason's $\geq$ 7, Stage $\geq$ T2b) = 2]
[Category = 4] <= [count(PSA $\geq$ 10 ng/ml, Gleason's $\geq$ 7, Stage $\geq$ T2b) = 3]

where count(S1, S2, .., Sn) is a derived attribute that counts the number of sentences between the parentheses that are true. To express the above classification schema by a decision tree or standard decision rules would require a more complex and more difficult to interpret structure, for example, a decision tree with eight leaves and seven internal nodes, or eight standard rules.

Expressions with a count attribute are generalizations of the so-called n-of-m relations (stating that n of m binary attributes are true in a description) [15]. AQ21-NI can learn, however, not only n-of-m special cases, but more general expressions that involve both count attributes and other conditions, for example:
[DiseaseState=severe] <= [count(C1, C5, C8)$\geq$ 2]&[Abnormality-type=A v C]

As one can see, the attributional rules can express quite elaborate conditions and are closely related to the equivalent natural language descriptions. The latter feature makes them easy to translate into such descriptions.

2.5 Learning Optimized Sets of Alternative Classifiers

From any non-trivial set of concept examples, it is usually possible to generate many alternative inductive generalizations of these examples. Such alternative hypotheses can be useful in a variety of practical applications. For example, in medicine, it may be desirable to generate alternative explanations of the symptoms to protect a doctor from overlooking a rare disease. AQ21-NI seeks a collection of alternative classifiers that optimizes a user-defined multi-criterion measure. Here, a classifier is a collection of attributional rulesets, where each ruleset is associated with one value, e.g., one disease, in the domain of the output attribute, e.g., diagnosis (for more explanation, consult [8]). The purpose of optimizing the collection is to include in it, for example, alternative classifiers that are maximally different from each other.

For example, for the robots problem presented above, one execution of AQ21-NI generates two alternative rulesets for the class "friendly" at the first stage of processing:

```
Classifier 1:
[robot=friendly]
  <= [holding=book] & [size=small..medium] : p=4,n=0
  <= [holding=book v flag]&[antenas=no]& [size=small..medium]
       : p=3,n=0
```

```
Classifier 2:
[class=friendly]
   <= [holding=book] & [size=small..medium]       :  p=4,n=0
   <= [holding=book v flag] & [size=medium]       :  p=4,n=0
   <= [holding=flag] & [size=small] & [hands=yes] :  p=1,n=0.
```

The algorithm for creating optimized collections of alternative classifiers is described in [9].

3 Generating Natural Language Descriptions

The second stage involves transforming the learned attributional rulesets into grammatically correct natural language descriptions. This task is done according to the following hard-coded rules:

1. Attribute names used in the rules are translated onto their natural language equivalences provided by the user.
2. Symbols ”=” and ”:” in rule conditions used with regular and compound attributes are translated into the word ”is.” Symbols ”>”, ”<” are translated into ”greater than”, ”smaller than,” and symbols ”≥” and ”≤” are translated into ”at least,” and ”at most”.
3. Attribute values connected in a condition by internal disjunction or internal conjunction are separated by a comma, except for the last value that is separated by an ”or,” or ”and”, respectively. Range expressions ”val1..val2” are translated into a statement ”between val1 and val2, inclusively”.
4. Conditions with a count attribute are transformed according to a template. If the count refers to statements, then the condition is transformed into ”The number of true statements on the list”, followed by the list of conditions transformed to natural language, ”is”, and followed by the value indicated in the count condition. If the count refers to attributes, then the condition is transformed to a statement: ”The number of attributes on the list L whose values are Rel is Val”, where L, Rel and Val are indicated in the condition. For example, [count(x1, x3, x5, x8 > 3) = 2] is transformed into ”The number of attributes on the list (x1, x3, x5, x8) whose values are greater than 3 is 2.”
5. If there is more than one, but at most three implications ”<=” after the rule head, that is, the rule consequent (this number is a modifiable parameter), they are replaced by an ”or.” To reflect the spirit of cognitive aspects of attributional calculus, if there are more ”<=” than allowed by the parameter, they are transformed into a sentence: ”The strongest rule implying <head condition> is <natural language version of the strongest rule>. Other rules also implying the <head condition> are <natural language representation of the remaining rules>”. The rules’ strength is determined according to a user-defined criterion, such as Q(w) (default), coverage, confidence, etc.

6. Numbers p and n listed at the end of a rule are translated to natural language by filling a template: "The rule is satisfied by p positive and n negative training examples." This statement may be followed by lists of these examples. Similarly, a template is used to translate the weights associated with each condition in a rule.

To obtain a satisfactory natural language representation of attributional rulesets, it is important to appropriately name attributes and their legal values in preparing the input to the program.

4 An Example of Applications in Medicine

This example describes an application of AQ21-NI to the problem of diagnosing medulloblastoma from patients' gene micro-arrays (representing degrees of expressions of patients' genes). Medulloblastoma is a highly invasive primitive neuroectodermal tumor of the cerebellum and the most common malignant brain tumor of childhood. The data for this application were obtained from Gene Expression Omnibus, available from www.ncbi.nlm.nih.gov/geo. The original gene micro-array data consists of 46 records, split into two groups: 20 and 26 records, describing patients with metastatic and non-metastatic tumors, respectively. Each record registers values of 2059 real-valued attributes (representing gene expressions). In the experiments we obtained 16 and 12 unique examples of metastatic and non-metastatic tumors, respectively [11].

From these examples, AQ21-NI at the first stage generated two simple rules for diagnosing metastatic tumor that require measuring only four gene expressions:

```
[Cancer = metastatic]
<= [Gene-1611-expression <= 100.9: 18, 8] &
   [Gene-1036-expression=-41.76..160.8: 18, 20] &
   [Gene-914-expression<=21.5: 20, 15]              : p=16,n=0
<= [Gene-1783-expression >= 96.6: 6,0]              : p= 6,n=0
```

An equivalent natural language description is: *"Cancer is metastatic, if gene 1611 expression is at most 100.9, gene 1036 expression is between -41.76 and 160.8, and gene 914 expression is at most 21.5, or gene 1783 expression is at least 96.6."*

When the experiment was performed using 5-fold cross-validation, the rules obtained by AQ21-NI had predictive accuracy about 95%. It is noteworthy that in the experiments that inspired our work in this domain [5], the authors developed a neural net that requires measuring 80 genes, and its reported predictive accuracy was about 72%. Thus, their result is not only less accurate than that obtained by AQ21-NI, but is also a significantly more complex, as it requires measuring expression of many more genes. Moreover, it is a black box solution that is very difficult to interpret.

5 Relation to Other Work

The first phase of AQ21-NI is related to programs learning standard decision rules. Among such programs are CN2 [2], C4.5 [14], RIPPER [3], programs using rough-set theory approach, e.g., [12] and [4], programs learning fuzzy rules, e.g., [19], and those applying evolutionary computation, e.g. [16]. Because standard decision rules have a relatively low expression power, these programs cannot learn more expressive attributional rules that are learned by AQ21-NI, and have fewer capabilities. To the best of author's knowledge, AQ21-NI is the only program currently in existence that has such a large number of different capabilities integrated in one program.

Also, the authors are not aware of any existing rule learning program that performs the second stage of learning, that is, generates natural language concept descriptions. Work on this stage concerns generating natural language descriptions from logical-style rules. The task of generating natural language descriptions is usually addressed from two different perspectives, the template-based, which maps non-linguistic input directly to natural language (without intermediate representations), and the standard method, which builds sentences through a semantic analysis of the text being generated [18].

As was mentioned earlier, attributional calculus facilitates learning of richer and frequently also simpler generalizations of examples than representations based standard decision rules. The cost for this advantage is, however, a significantly higher complexity of the learning algorithm, and, consequently, a longer time of its execution. Due to the great progress in increasing speed of modern computers, the second issue is of decreasing importance. In our experiments, AQ21-NI has proven to be quite efficient. An earlier version of AQ learning program was effectively applied to problems with millions of training examples.

6 Summary

Natural induction aims at creating knowledge from data that is not only accurate but also easy to understand and interpret. The latter objective is achieved by first learning expressions in attributional calculus that adds to standard logic several new constructs, and then transforming the learned descriptions into equivalent natural language descriptions. A methodology for natural induction has been implemented in the AQ21-NI rule learning program. The program seamlessly integrates several new features that include learning in two modes-theory formation and pattern discovery, learning with compound attributes, learning censored rules and learning optimized collections of alternative classifiers. Due to space limitations, the paper includes only very brief descriptions of these features. More detailed descriptions are in publications downloadable from http://www.mli.gmu.edu. An application of AQ21-NI to a problem in bioinformatics produced a hypothesis that a medical expert evaluated as having an important medical value, because it suggests adjusting thresholds in the currently used diagnostic procedure.

The ability of AQ21-NI to produce natural language descriptions makes it attractive for application domains in which understandability of computer-generated knowledge is highly important, such as medicine, bioinformatics, sociology, psychology, economy, business, archeology, civil engineering, and others.

References

1. Caruana, R., Niculescu-Mizil, A.: An Empirical Comparison of Supervised Learning Algorithms. In: Proceedings of the 23rd Intl Conference on Machine Learning, 2006 (2006)
2. Clark, P., Niblett, T.: The CN2 Induction Algorithm. Machine Learning 3, 261–289 (1989)
3. Cohen, W.: Fast Effective Rule Induction. In: Proc. of the 12th Intl Conference on Machine Learning 1995 (1995)
4. Grzymala-Busse J.W.: Rough Set Strategies to Data with Missing Attribute Values. In: Proc. of the Workshop on Found. and New Directions in Data Mining, 2003 (2003)
5. MacDonald, T.J., Brown, K., LaFleur, B., Paterson, K., Lawlor, C., Chen, Y., Packer, R., Cogen, P., Stephan, D.: Expression Profiling of Medulloblastoma: PDGFRA and the RAS/MAPK Pathway as Therapeutic Targets for Metastatic Disease. Nature Genetics 29, 143–152 (2001)
6. Michalski, R. S.: AQVAL/1–Computer Implementation of a Variable-Valued Logic System VL1 and Examples of its Application to Pattern Recognition. In: Proceedings of the First International Joint Conference on Pattern Recognition, 1973, pp. 3–17 (1973)
7. Michalski, R.S.: A Theory and Methodology of Inductive Learning. Artificial Intelligence, pp. 111–161 (1983)
8. Michalski, R.S.: Attributional Calculas: A Logic and Representation Language for Natural Induction. Reports of the Machine Learning and Inference Laboratory MLI 04-2. George Mason University (2004)
9. Michalski, R. S.: Generating Alternative Hypotheses in AQ Learning. Reports of the Machine Learning and Inference Laboratory MLI 04-6. George Mason Univ. (2004)
10. Michalski, R.S., Winston, P.H.: Variable Precision Logic. Artificial Intelligence Journal 29, 121–146 (1986)
11. Michalski, R. S., Kaufman, K., Pietrzykowski, J., Wojtusiak, J., Mitchell, S., Seeman, W.D.: Natural Induction and Conceptual Clustering: A Review of Applications. In; Reports of the Machine Learning and Inference Laboratory MLI 06-3. George Mason University (2006)
12. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
13. Pazzani, M. J.: Knowledge discovery from data? IEEE Intelligent Systems (10-13, March/April 2000)
14. Quinlan, J.R.: C4.5 Systems for Machine Learning. Morgan Kaufmann Publ, San Francisco (1993)
15. Sebag, M.: Constructive Induction: A Version Space-based Approach. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence, IJCAI 99 (1999)

16. Setzkorn, C., Paton, R.C.: On the use of multi-objective evolutionary algorithms for the induction of fuzzy classification rule systems. Biosystems 81(2) (2005)
17. Suzuki, E., Zytkow, J.M.: Unified algorithm for undirected discovery of exception rules. International Journal of Intelligent Systems 20(6), 673–691 (2005)
18. Van Deemter, K., Theune, M., Krahmer, E.: Real vs. Template-Based Natural Language Generation: A false Opposition? Computational Linguistics 31(1) (2005)
19. Van Zyl, J., Cloete, I.: Simultaneous Concept Learning of Fuzzy Rules. In: Proceedings of the Fifteenth European Conference on Machine Learning, 2004 (2004)
20. Wojtusiak, J., Michalski, R. S., Kaufman, K., Pietrzykowski, J.: The AQ21 Natural Induction Program for Pattern Discovery: Initial Version and its Novel Features. In: Proceedings of The 18th IEEE International Conference on Tools with Artificial Intelligence, Washington, D.C., 2006 (2006)

Hierarchical Rough Classifiers

Sinh Hoa Nguyen[1] and Hung Son Nguyen[2]

[1] Polish-Japanese Institute of Information Technology
Koszykowa 86, 02008, Warszawa, Poland
[2] Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
hoa@mimuw.edu.pl, son@mimuw.edu.pl

Abstract. The major applications of rough set theory in data mining are related to the modeling of concepts using rough classifiers, i.e., the algorithms classifying unseen objects into lower or upper approximations of concepts. This paper investigates a class of compound classifiers called multi-level (or hierarchical) rough classifiers (MLRC). We present the most recent issues on the construction of such classifiers from data using concept ontology as an additional domain knowledge. The idea is based on the bottom-up manner to gradually synthesize the multi-layer rough classifier for the complex target concept from the simpler classifiers. We illustrate the proposed method by experiments on real-life data.

Keywords: Rough sets, concept approximation, knowledge discovery.

1 Introduction

A great effort of researchers in machine learning and data mining has been made to develop efficient methods for approximation of concepts from data [1]. In a typical process of concept approximation we assume that there is given information consisting of values of conditional and decision attributes on objects from a finite subset (training set, sample) of the object universe and on the basis of this information one should induce approximations of the concept over the whole universe. Nevertheless, there exist many problems that are still unsolvable for existing state of the art methods, because of the high complexity of learning algorithms or even unlearnability of hypothesis spaces.

Rough set theory [2] [3], has been introduced as a tool for concept approximation from incomplete information or imperfect data. The essential idea of rough set approach is to search for two descriptive sets called the *lower and upper approximations* containing those objects that certainly, or possibly belong to the concept, respectively.

Most concept approximation methods realize the inductive learning approach, which assumes that a partial information about the concept is given by a finite sample, so called *the training sample or training set*, consisting of positive and negative cases. The information from training tables makes the search for patterns describing the given concept possible. Utilization of domain knowledge into

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 40–50, 2007.

learning process becomes a big challenge for improving and developing more efficient concept approximation methods. In previous papers [4] [5] we investigated the domain knowledge given in form of a concept hierarchy which is the simplest form of concept ontology. We have proposed some algorithms for induction of “multi-layer rough classifier” (MLRC) from data [4] based on the layered learning [6] and rough set approaches. The main idea is to gradually synthesize a target concept from simpler ones. The learning process can be imagined as a treelike structure. At the lowest layer, primitive concepts are approximated using feature values available from a data set. At the next layers the complex concepts are synthesized from the primitive ones. This process is repeated for successive layers until the target concept which is located at the highest layer is reached, and as the result the multi-layer rough classifier is returned.

This paper summarizes the most recent applications of rough sets in construction of hierarchical classifiers. We present the general framework of rough set based hierarchical learning algorithm, we will also discuss some related issues and illustrate our ideas in the corresponding case study problems. In particular, we investigate several strategies of choosing the appropriate learning algorithm for first level concepts. We also present the method of learning the intermediate concepts and some methods of embedding the domain knowledge into the learning process in order to improve the quality of hierarchical classifiers. We illustrate the proposed method for the sunspot recognition problem.

2 Preliminaries

Concepts can be understood as definable sets of objects. Formally, any subset X of a given universe $\mathfrak{U}$ which can be described by a formula of $\mathcal{L}$ is called the concept in $\mathcal{L}$. The *concept approximation problem* can be understood as searching for approximate description – using formulas of a predefined language $\mathcal{L}$ – of concepts that are definable in other language $\mathcal{L}^*$. Inductive learning is the concept approximation method that searches for description of unknown concept using finite set $U \subset \mathfrak{U}$ of training examples.

Rough set theory has been introduced by Professor Z. Pawlak [2] as a tool for approximation of concepts under uncertainty. The theory is featured by operating on two definable subsets, i.e., a lower approximation and upper approximation. The first definition, so called the “standard rough sets”, was introduced by Pawlak in his pioneering book on rough set theory [2].

Given an information system $\mathbb{S} = (U, A)$, where U is the set of training objects, A is the set of attributes and a concept $X \subset U$. Assuming at the moment that only some attributes from $B \subset A$ are accessible, then this problem can be also described by appropriate decision table $\mathbb{S} = (U, B \cup \{dec_X\})$, where $dec_X(u) = 1$ for $u \in X$, and $dec_X(u) = 0$ for $u \notin X$.

First one can define called the *B-indiscernibility relation* $IND(B) \subset U \times U$ in such a way that $x\ IND(B)\ y$ if and only if x, y are indiscernible by attributes from B, i.e., $inf_B(x) = inf_B(y)$. Let $[x]_{IND(B)} = \{u \in U : (x,u) \in IND\,(B)\}$ denote the equivalence class of $IND\,(B)$ defined by x. The lower and upper approximations of X (using attributes from B) are defined by: $\mathbf{L}_B(X) = \big\{x : [x]_{IND(B)} \subseteq X\big\}$ and $\mathbf{U}_B(X) = \big\{x : [x]_{IND(B)} \cap X \neq \varnothing\big\}$. Let us point out that there are many extensions of the standard definition of rough sets, e.g., variable rough set model [7] or tolerance approximation space [8]. In these methods, rough approximations of concepts can be also defined by *rough membership function*, i.e., a mapping $\mu_X : U \to [0,1]$ such that $\mathbf{L}_{\mu_X} = \{x \in U : \mu_C(x) = 1\}$ and $\mathbf{U}_{\mu_X} = \{x \in U : \mu_X(x) > 0\}$ are lower and upper approximation of a given concept X. In case of the classical rough set theory, the rough membership function is defined by $\mu_X^B(x) = \frac{\left|X \cap [x]_{IND(B)}\right|}{\left|[x]_{IND(B)}\right|}$.

The inductive learning approach to rough approximations of concepts we assume that U is a finite sample of objects from a universe $\mathfrak{U}$ and $X = \mathcal{C} \cap U$ is the representation of a unknown concept $\mathcal{C} \subset \mathfrak{U}$ in U. The problem can be understood as searching for an extended rough membership function $\mu_{\mathcal{C}} : \mathfrak{U} \to [0,1]$ for $\mathcal{C} \subset \mathfrak{U}$ such that the corresponding rough approximations defined by $\mu_{\mathcal{C}}$ are the good approximations of $\mathcal{C}$.

$$\begin{tikzcd}
U \arrow[r, dashed] & \mu_X : U \to [0,1] \\
\cap & \Downarrow \\
\mathfrak{U} \arrow[r, dashed] & \mu_{\mathcal{C}} : \mathfrak{U} \to [0,1]
\end{tikzcd}$$

The algorithm that calculates the value $\mu_{\mathcal{C}}(x)$ of extended rough membership function for each new unseen object $x \in \mathfrak{U}$ is called *the rough classifier*.

In fact, rough classifiers can be constructed from any other classifiers [4] [9]. This process is called *the roughyfication*, and the general idea is to change the binary output of the original classifier into a multi-value rough membership function. By this way, rough classifier can be constructed from any classifier including decision tree, neural network, SVM classifier, etc. All these methods will be used as building blocks for construction of compound classifiers.

A classifier that is created by composition of some other classifiers is called the *hierarchical classifier*. Formally, let $\mathbb{CA}, \mathbb{CA}_1, ..., \mathbb{CA}_k$ be classifiers realizing the computation of concepts $F, f_1, ..., f_k$, respectively, then the hierarchical classifier $\mathbb{CA}^* = \mathbb{CA}(\mathbb{CA}_1, ..., \mathbb{CA}_k)$ realizes the function $F^* = F(f_1, ..., f_k)$. Let us note that by this way one can extend the learnability of a the concept approximation problem, because even if all simple classifiers $\mathbb{CA}, \mathbb{CA}_1, ..., \mathbb{CA}_k$ realize functions from a hypothesis space H, then the hierarchical classifiers can compute functions outside of H.

3 Induction of Hierarchical Rough Classifiers

In this section we present general multi layered learning scheme for synthesis of hierarchical rough classifier. Layered learning is designed for domains that are too complex for learning a mapping directly from the input to the output representation. The main principles of the layered learning paradigm [6].

1. **Breaking down the problem into several task layers:** At each layer, a concept needs to be acquired and a learning algorithm solves the local concept-learning task.

2. **Using a bottom-up incremental approach to hierarchical concept decomposition:** Starting with low-level concepts, the process of creating new sub-concepts continues until the high-level concepts, that deal with the full domain complexity, are reached.
3. **Exploiting data and learning methods to train and/or adapt. Learning occurs separately at each level:** Learning algorithms may be different for different sub-concepts in the decomposition hierarchy.
4. **The output of learning in one layer feeds into the next layer:** This is the key characteristic of hierarchical learning, because each learned layer directly affects the learning at the next layer.

When using the layered learning paradigm, we assume that the target concept can be decomposed into simpler ones called sub-concepts. A hierarchy of concepts has a treelike structure. Basic concepts are located at the lowest level and the target concept at the highest level. Basic concepts are learned directly from input data when any higher level concept is composed by the concepts located at lower levels. We assume that a concept decomposition hierarchy is given by domain knowledge [10]. However, one should observe that concepts and dependencies among them represented in domain knowledge are expressed often in natural language. Hence, there is a need to approximate such concepts and such dependencies as well as the whole reasoning process. This issue is directly related to the computing with words paradigm [11], [12] and to rough-neural approach [13], in particular to rough mereological calculi on information granules (see, e.g., [14], [15], [16], [17], [10]).

Formally, the concept hierarchy is a tuple $\mathcal{H} = (\mathbb{C}, \mathbb{R})$, where $\mathbb{C} = A \cup \{C_1, ..., C_n\} \cup \{dec\}$ is a finite set of concepts including basic concepts (attributes from A), intermediated concepts $(C_1, ..., C_n)$ and the target concept dec; and $\mathbb{R} \subset \{C_1, ..., C_n, dec\} \times \mathbb{R}$ is a directed acyclic graph (DAG) describing the relationship between concepts and attributes.

Each concept C in the given hierarchy together with all its descendants forms a subtree $\mathcal{H}|_C$. We says that the concept C is in the layer h if and only if $h = height(\mathcal{H}|_C)$. In this way, elements of the hierarchy are divided into layers according to the heights of corresponding subtrees. Thus all input attributes should be placed in the lowest layer (layer 0), while the decision attribute is on the highest level. In this paper we assume that a concept hierarchy $\mathcal{H}$ is given. The training set is represented by a decision table $\mathbb{S}_S = (U, A, D)$, where D is a set of decision attributes related to concepts. Decision values indicate whether an objects belong to to a concept in the hierarchy.

3.1 Learning Framework for Hierarchical Rough Classifiers

The main issue in layered learning algorithms is designing a general schema for learning local concept composition. Our method operates from the lowest level to the highest one. Assume that each concept C_k in the hierarchy is associated with a tuple $T_{C_k} = (U_k, A_k, O_k, ALG_k, h_k)$ where U_k is the set of training objects used for learning the concept C_k; A_k is the set of attributes relevant for learning the concept C_k; O_k is the set of outputs used to define the concept C_k; ALG_k is the algorithm used for learning the function mapping vector values over A_k into O_k; h_k is the hypothesis, which is a result of running the algorithm ALG_k on the training set U_k. Any hypothesis h_k of the concept C_k affects on creation of the tuple T_C for its direct ancestor C in the next level of the decomposition hierarchy in two ways, i.e., h_k is used to construct (1) the set of training examples U and (2) the set of features A for the concept C.

In rough sets, the hypothesis h_k is represented by a pair $(\mu_{C_k}(.), \mu_{\overline{C_k}}(.))$ of two membership functions. Let us describe in detail some important issues that should be settled when applying the layered learning idea in the synthesis of compound concepts.

Usually, primitive concepts are approximated using input features available from the data set. The choice of the proper algorithm is the most important in this step. In case of supervised learning, using information available from a concept hierarchy for each primitive concept C_b, one can create a training decision system $\mathbb{S}_{C_b} = (U, A_{C_b}, dec_{C_b})$, where $A_{C_b} \subseteq A$, and $dec_{C_b} \in D$. To approximate the concept C_b one can apply any classical method (e.g., k-NN, decision tree, or rule-based approach [18], [19]) to the table $\mathbb{S}_{C_b}$ (see Section 2).

Let us point out a special case when a concept is a generalization of another concept. This problem is very intensively investigated in data mining and KDD [20]. Many methods have been proposed to create a whole concept hierarchy for one attribute. In case of real value attributes, this process is called the discretization. The usual discretization methods define the more general concept by cuts. The rough set approach to discretization utilizes the idea of “soft cuts” instead of traditional “crisp cuts” [21] [22].

For compound concepts in the hierarchy, we can use the rough classifiers as a building blocks to develop a multi-layered classifier. Precisely, let $prev(C) = \{C_1, ..., C_m\}$ be the set of concepts, which are connected with C in the hierarchy. Assume that we are given rough membership functions $\mu_{C_1}(x)$, ..., $\mu_{C_m}(x)$. The rough approximation of the concept C can be determined by performing two steps: (1) construct a decision table $\mathbb{S}_C = (U, A_C, dec_C)$ relevant for the concept C; and (2) induce a rough classifier for C using decision table $\mathbb{S}_C$. In [4], the training table $\mathbb{S}_C = (U, A_C, dec_C)$ was constructed as follows:

- The set of objects U is common for all concepts in the hierarchy.
- $A_C = h_{C_1} \cup h_{C_2} \cup ... \cup h_{C_m}$, where h_{C_i} is the output of the hypothetical classifier for the concept $C_i \in prev(C)$. If C_i is an input attribute $a \in A$ then $h_{C_i}(x) = \{a(x)\}$, otherwise $h_{C_i}(x) = \{\mu_{C_i}(x), \mu_{\overline{C_i}}(x)\}$.

Repeating those steps for each concept through the bottom to the top layer we obtain a “hybrid classifier” for the target concept, which is a combination of classifiers of various types. In the second step, the learning algorithm should use the decision table $\mathbb{S}_C = (U, A_C, dec_C)$ to “resolve conflicts” between classifiers of its children. We have proposed in [4] two methods for learning approximation of compound concept from decision table $\mathbb{S}_C$:

- **Naive (simple) method:** This method treat $\mathbb{S}_C$ as a normal decision table $\mathbb{S}'$ with more attributes. By extracting rules from $\mathbb{S}'$ the rule-based approximations of the concept C are created.
- **Stratification method:** Instead of using just a value of membership function or weight we are using linguistic statements such as *"the likeliness of the occurrence of* C_1 *is low"*. That yields fuzzy-like layout, or linguistic variables, of attribute values. One may (and in some cases should) consider also the case when these subsets overlap.

The presented above supervised method is applicable only for data sets in which the decision attribute (i.e., O_k) for each concept in the hierarchy is given. In situation, when our knowledge about the concept C_k is limited to the fact that it is depended on a set A_k of other concepts in the lower level, we propose to modify the previous algorithm as follows:

- Granulate the the sample of objects U_k using a clustering algorithm according to the available information from A_k;
- Define the decision attribute O_k as the membership function of objects to clusters.

The proposed ideas are gathered in Algorithm 1.

Algorithm 1. **M**ulti-layered **R**ough **C**lassifier (**MlRC**)

```
Input: Decision system $\mathbb{S} = (U, A, d)$, concept hierarchy $H$;
Output: Schema for concept composition
for $l := 0$ to $max\_level$ do
    for (any concept $C_k$ at the level $l$ in $H$) do
        if $l = 0$ then
            $U_k := U$;
            $A_k := B$, where $B \subseteq A$ is a set relevant to define $C_k$
        else
            $U_k := U$;
            $A_k = \bigcup O_i$ - a collection of outputs generated by all sub-concepts $C_i$ of $C_k$;

            Generate a rule set determining of the concept $C_k$ approximation;
            Generate the output vector $[\mu_{C_k}(x), \mu_{\overline{C_k}}(x)]$ for any object $x \in U_k$
```

3.2 Extended Layered Learning Algorithm

Recall that the concept hierarchy represents a set of concepts and a binary relation which connects a "child" concept with its "parent". The most important relation types are the subsumption relations (written as "is-a" or "is-part-of") defining which objects (or concepts) are members (or parts) of another concepts in the hierarchy. Besides the "child-parent" relations, we consider new kinds of relations associating with concepts in the hierarchy. We call them *domain-specific constraints*. We consider two types of constraints: (1) constraints describing relationships between a “parent” concept and its ”child” concepts; and (2) constraints connecting the “sibling” concepts (having the same parent).

Formally, the extended concept hierarchy is a triple $\mathcal{H} = (\mathbb{C}, \mathbb{R}, Constr)$, where $\mathbb{C} = \{C_1, ..., C_n\}$ is a finite set of concepts including primitive, intermediate and the target concept; $\mathbb{R} \subseteq \mathbb{C} \times \mathbb{C}$ is child-parent relation in the hierarchy; and $Constr$ is a set of constraints. In this paper, we consider constraints expressed by association rules of the form $\mathbf{P} \rightarrow_{\alpha} \mathbf{Q}$, where $\mathbf{P}, \mathbf{Q}$ are boolean formulas over the set of boolean variables corresponding to concepts from $\mathbb{C}$ and their complements, and $\alpha \in [0, 1]$ is the confidence of this rule.

Let us assume that an extended concept hierarchy $\mathcal{H} = (\mathbb{C}, \mathbb{R}, Constr)$ is given. In the layered learning algorithm for hierarchical rough classifier presented in Section 3.1, one can observe that, if sibling concepts $C_1, ..., C_m$ are independent, the membership function values of these concepts are “sent” to the “parent” C, without any correction. Thus the membership value of weak classifiers may disturb the training table for the parent concept and cause the misclassification when testing new unseen objects. We have present two techniques that enable the expert to improve the quality of hybrid classifiers by embedding their domain knowledge into learning process. They were called the *constraint-based refining of weak classifiers* and the *constraint-based selection of learning algorithm*. The detail description of these methods are presented in [9].

4 Case Study: Sunspot Classification Problem

Sunspots are the subject of interest to many astronomers and solar physicists. Sunspot observation, analysis and classification form an important part of furthering the knowledge about the Sun. Sunspot classification is a manual and very labor intensive process that could be automated if successfully learned by a machine. The main goal of the first attempt to sunspot classification problem is to classify sunspots into one of the seven classes $\{A, B, C, D, E, F, H\}$, which are defined according to the McIntosh/Zurich Sunspot Classification Scheme. More detailed description of this problem can be found in [5].

The data was obtained by processing NASA SOHO/MDI satellite images to extract individual sunspots and their attributes characterizing their visual properties like size, shape, positions. The data set consists of 2589 observations from the period of September 2001 to November 2001. The main difficulty in correctly determining sunspot groups concerns the interpretation of the classification scheme itself. There is a wide allowable margin for each class (see Figure 1). Therefore, classification results may differ between different astronomers doing the classification.

In [5], we have presented a method for automatic modeling the domain knowledge about sunspots concept hierarchy. The main part of this ontology is presented in Figure 2. We have shown that rough membership function can be induced using different classifiers, e.g., k-NN, decision tree or decision rule set. The problem is to chose the proper type of classifiers for every node of the hierarchy. In experiments with sunspot data, we applied the rule based approach for

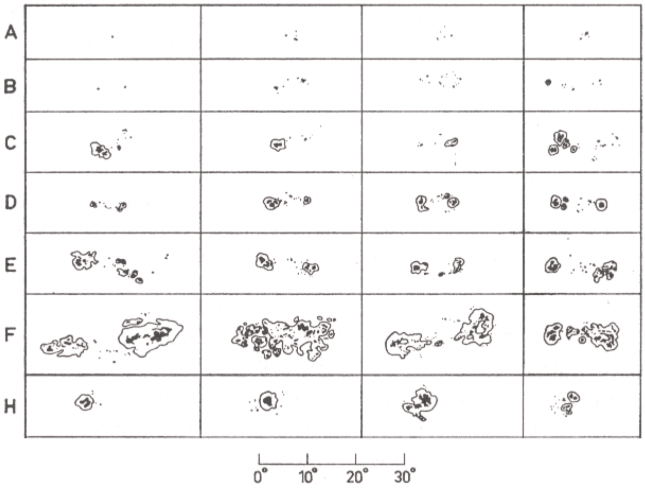


Fig. 1. Possible visual appearances for each class. There is a wide allowable margin in the interpretation of the classification rules making automatic classification difficult.

concepts in the lowest level, decision tree based approach for the concepts in the intermediate levels and the nearest neighbor based approach the target concept.

Figure 3 (left) presents the classification accuracy of "hybrid classifier" obtained by composition of different types of classifiers and "homogenous classifier" obtained by composition of one type of classifiers. The first three bars show qualities of homogenous classifiers obtained by composition of k-NN classifiers, decision tree classifiers and rule based classifiers, respectively. The fourth bar (the gray one) of the histogram displays the accuracy of the hybrid classifier.

The use of constraints also give a profit. In our experiment, these constraints are defined for concepts at the second layer to define the training table for the

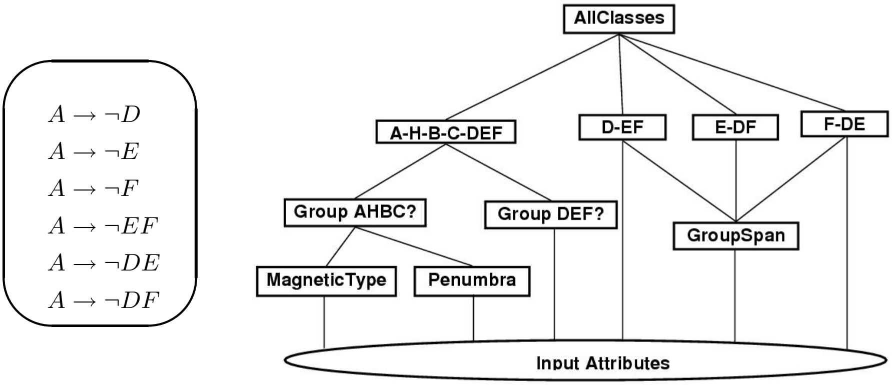


Fig. 2. The concept hierarchy for sunspot recognition problem

target concept *AllClasses*. It is because the noticeable breakdown of accuracy have been observed during experiments. We use the strategy proposed in Section 3.1 to settle the final rough membership values obtained from its children *A-H-B-C-DEF*, *D-EF*, *E-DF*, *F-DE* (see the concept hierarchy in Figure 2). One can observe that using constraints we can promote good classifiers in a composition step. A better classifier has higher priority in a conflict situation. The experiment results are shown in Figure 3. The gray bar of the histogram displays the quality of the classifier induced without concept constraints and the black bar shows the quality of the classifier generated using additional constraints.

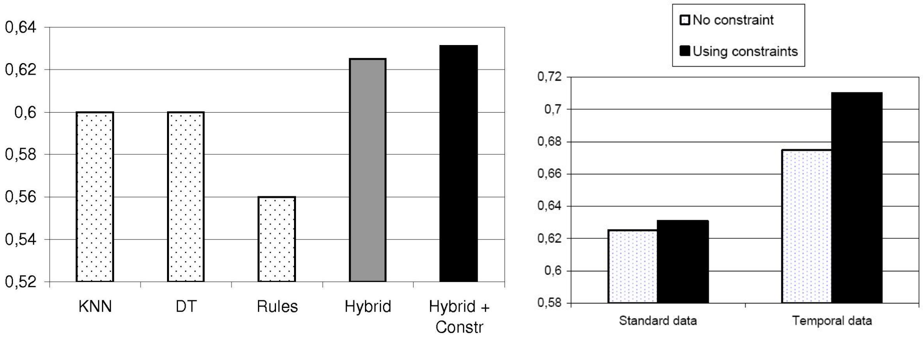


Fig. 3. Accuracy comparison of different hierarchical learning methods

Another approach to manage with sunspot recognition problem is related to temporal features. Comparative results are showed in Figure 3 (right). The first two bars in the graph describe the accuracy of classifiers induced *without* temporal features and the last two bars display the accuracy of classifiers induced *with* temporal features. One can observe a clear advantage of the last classifiers over the first ones. The experimental results also show that the approach for dealing with temporal features and concept constraints considerably improves approximation quality of the complex groups such as B, D, E and F.

5 Conclusions

We presented a new method for concept synthesis. It is based on the hierarchical learning approach. Unlike traditional approach, layered learning methods induce the approximation of concepts not only from accessible data but also from the domain knowledge given by experts. The hierarchical learning approach showed to be promising for the complex concept synthesis. Experimental results with road traffic simulation are showing advantages of this new approach in comparison to the standard learning approach. The main advantages of the hierarchical learning approach include: high precision of concept approximation, high generality of concept approximation, simplicity of concept description, high computational speed, and the possibility of localization sub-concepts difficult to approximate.

References

1. Kloesgen, W., Żytkow, J. (eds.): Handbook of Knowledge Discovery and Data Mining. Oxford University Press, Oxford (2002)
2. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
3. Pawlak, Z.: Some Issues on Rough Sets. Transactions on Rough Sets 1, 1–58 (2004)
4. Nguyen, S.H., Bazan, J., Skowron, A., Nguyen, H.S.: Layered Learning for Concept synthesis. In: Peters, J.F., Skowron, A., Grzymala-Busse, J.W., Kostek, B., Swiniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 187–208. Springer, Heidelberg (2004)
5. Nguyen, T.T., Willis, C.P., Paddon, D.J., Nguyen, S.H., Nguyen, H.S.: Learning sunspot classification. Fundamenta Informaticea 72(1-3), 295–309 (2006)
6. Stone, P.: Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer. The MIT Press, Cambridge, MA (2000)
7. Ziarko, W.: Variable Precision Rough Set Model. Journal of Computer and System Sciences 46, 39–59 (1993)
8. Skowron, A.: Approximation spaces in rough neurocomputing. In: Inuiguchi, M., Tsumoto, S., Hirano, S. (eds.) Rough Set Theory and Granular Computing. Studies in Fuzziness and Soft Computing, vol. 125, pp. 13–22. Springer, Heidelberg (2003)
9. Nguyen, S.H., Nguyen, T.T., Nguyen, H.S.: Ontology driven concept approximation. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Slowinski, R. (eds.) RSCTC 2006. LNCS, vol. 4259, pp. 547–556. Springer, Heidelberg (2006)
10. Skowron, A., Stepaniuk, J.: Information Granules and Rough-Neural Computing [13] pp. 43–84
11. Zadeh, L.A.: Fuzzy logic = computing with words. IEEE Transactions on Fuzzy Systems 4, 103–111 (1996)
12. Zadeh, L.A.: A new direction in AI: Toward a computational theory of perceptions. AI Magazine 22(1), 73–84 (2001)
13. Pal, S.K., Polkowski, L., Skowron, A. (eds.): Rough-Neural Computing: Techniques for Computing with Words. Cognitive Technologies. Springer, Heidelberg (2003)
14. Polkowski, L., Skowron, A.: Rough Mereology: A New Paradigm for Approximate Reasoning. International Journal of Approximate Reasoning 15(4), 333–365 (1996)
15. Polkowski, L., Skowron, A.: Towards Adaptive Calculus of Granules. In: Zadeh, L.A., Kacprzyk, J. (eds.) Computing with Words in Information/Intelligent Systems, pp. 201–227. Springer, Heidelberg (1999)
16. Polkowski, L., Skowron, A.: Rough mereological calculi of granules: A rough set approach to computation. Computational Intelligence 17(3), 472–492 (2001)
17. Skowron, A., Stepaniuk, J.: Information granules: Towards foundations of granular computing. International Journal of Intelligent Systems 16(1), 57–86 (2001)
18. Friedman, J.H., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Heidelberg (2001)
19. Mitchell, T.: Machine Learning. Mc Graw Hill, New York (1998)

20. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc, San Francisco (2000)
21. Nguyen, H.S., Nguyen, S.H.: Fast split selection method and its application in decision tree construction from large databases. International Journal of Hybrid Intelligent Systems 2(2), 149–160 (2005)
22. Nguyen, H.S.: Approximate Boolean Reasoning: Foundations and Applications in Data Mining. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. Lecture Notes on Computer Science, vol. 4100, pp. 334–506. Springer, Heidelberg (2006)

Discrete Duality and Its Applications to Reasoning with Incomplete Information

Ewa Orłowska[1] and Ingrid Rewitzky[2]

[1] National Institute of Telecommunications, Warsaw
[2] University of Stellenbosch, South Africa

Abstract. We present general principles of establishing a duality between a class of algebras and a class of relational systems such that topology is not involved. We show how such a discrete duality contributes to proving completeness of logical systems and to correspondence theory. Next, we outline applications of discrete dualities to analysis of data in information systems with incomplete information in the rough set-style, and in contexts of formal concept analysis.

1 General Principles of Discrete Duality and Duality Via Truth

Duality theory emerged from the work by Marshall Stone [Sto36] on Boolean algebras and distributive lattices in the 1930s. Jónsson and Tarski [JT51] extended Stone's results to Boolean algebras with operators. These operators are now known to be modal possibility operators. Later in the early 1970s Larisa Maksimova [Mak72, Mak75] and Hilary Priestley [Pri70, Pri72] developed analogous results for Heyting algebras, topological Boolean algebras, and distributive lattices. The latter has been extended to distributive lattices with operators by [Gol89, CLP91]. Since then establishing a duality has become an important methodological problem both in algebra and in logic. All the abovementioned classical duality results are developed using topological spaces as dual spaces of algebras.

Discrete duality is a duality where a class of abstract relational systems is a dual counterpart to a class of algebras. These relational systems are referred to as frames following the terminology of non-classical logics. A topology is not involved in the construction of these frames and hence they may be thought of as having a discrete topology. Establishing discrete duality involves the following steps. Given a class Alg of algebras (resp. a class Frm of frames) we define a class Frm of frames (resp. a class Alg of algebras). Next, for an algebra $W \in \mathsf{Alg}$ we define its canonical frame $X(W)$ and for each frame $X \in \mathsf{Frm}$ we define its complex algebra $C(X)$. Then we prove that $\mathcal{X}(W) \in \mathsf{Frm}$ and $\mathcal{C}(X) \in \mathsf{Alg}$. A duality between Alg and Frm holds provided that the following representation theorems are proved:

(D1) Every algebra $W \in \mathsf{Alg}$ is embeddable into the complex algebra of its canonical frame i.e., $\mathcal{C}(\mathcal{X}(W))$.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 51–56, 2007.

(D2) Every frame $X \in \mathsf{Frm}$ is isomorphic with a substructure of the canonical frame of its complex algebra, i.e., $\mathcal{X}(\mathcal{C}(X))$.

A distinguishing feature of this framework for establishing a discrete duality is that the algebraic and the logical notions involved in the proofs are defined in an autonomous way, we do not mix the algebraic and logical methodologies.

The separation of logical and algebraic constructs enables us to view dual classes of algebras and frames as two types of semantic structures of a formal language. As a consequence we easily obtain what we call duality via truth. Given a formal language Lan, a class of frames Frm which determines a relational semantics for Lan and a class Alg of algebras which determines its algebraic semantics, a duality via truth theorem says that these two kinds of semantics are equivalent in the following sense:

(DvT) A formula $\phi \in \mathsf{Lan}$ is true in every algebra of Alg iff it is true in every frame of Frm.

In order to prove such a theorem we need to prove the following lemma referred to as a complex algebra theorem.

(CA) For every frame $X \in \mathsf{Frm}$, a formula $\phi \in \mathsf{Lan}$ is true in X iff ϕ is true in the complex algebra $\mathcal{C}(X)$.

With the theorem (CA) and the representation theorem (D1) we can prove (DvT) theorem. The right-to-left implication of (DvT) follows from the left-to-right implication of (CA) and the left-to-right implication of (DvT) follows from right-to-left implication of (CA) and (D1). In this way the discrete duality contributes to a development of a relational semantics (resp. an algebraic semantics) once an algebraic semantics (resp. a relational semantics) of a language is known.

2 Application to Completeness and Correspondence Theorems

Discrete duality contributes also to a completeness result once a deductive system for the language Lan is given. Assume that an algebraic semantics of Lan is given in terms of a class Alg of algebras and a relational semantics in terms of a class Frm of frames such that a discrete duality holds between these two classes. We assume that the algebras from Alg are based on bounded lattices. To prove completeness we define a binary relation $\approx$ in the set of formulas of Lan in terms of provability of double implication, if it is among the propositional operations of Lan, or otherwise in terms of provability of a sequent built with a pair of formulas. Next we show that this relation is an equivalence relation and a congruence with respect to all the propositional operations admitted in Lan. Then we form the Lindenbaum algebra $\mathcal{A}_{\approx}$ of Lan. Its universe consists of equivalence classes $|\phi|$ (with respect to relation $\approx$) of formulas. Then we show that the algebra $\mathcal{A}_{\approx}$ belongs to the class Alg of algebras. Now, depending on whether we are interested in completeness with respect to the algebraic or relational semantics we proceed as follows.

To prove completeness of the deduction system with respect to the relational semantics we consider the canonical frame $\mathcal{X}(\mathcal{A}_{\approx})$ of the Lindenbaum algebra. Its universe consists of prime filters of $\mathcal{A}_{\approx}$. Then we form a model $M_{\approx}$ based on this frame. Preservation of operations by the mapping that provides an embedding of $\mathcal{A}_{\approx}$ into $\mathcal{C}(\mathcal{X}(\mathcal{A}_{\approx}))$ guaranteed by theorem (D1) enables us to prove the truth lemma saying that satisfaction of a formula ϕ in model $M_{\approx}$ by a filter F is equivalent to $|\phi| \in F$. From this lemma the completeness follows in the usual way.

To prove completeness of the deduction system with respect to the algebraic semantics we define a valuation of atomic formulas of Lan in $\mathcal{A}_{\approx}$ as $v(p) = |p|$ and we prove that it extends to all the formulas so that $v(\phi) = |\phi|$. Then we show that provability of a formula ϕ is equivalent to $v(\phi) = 1_{\approx}$, where $1_{\approx}$ is the unit element of the lattice reduct of $\mathcal{A}_{\approx}$. Then the completeness follows.

Discrete duality is also relevant for the correspondence theory which aims at finding relationships between truth of formulas in a frame and properties of relations in the frame. Typically, a correspondence has the following form:

(Cps) A formula $\phi \in \mathsf{Lan}$ is true in a frame $\mathcal{X}$ iff the relations of the frame have a certain property.

Given the classes Alg and Frm for which a discrete duality and duality via truth theorem with respect to a language Lan hold, we may consider the following correspondences:

(Cps1) The relations of a frame $\mathcal{X} \in \mathsf{Frm}$ have a certain property iff a formula $\phi \in \mathsf{Lan}$ is true in the complex algebra $\mathcal{C}(X)$.

(Cps2) A formula $\phi \in \mathsf{Lan}$ is true in an algebra $W \in \mathsf{Alg}$ iff the relations of the canonical frame $\mathcal{X}(W)$ have a certain property.

It is known that these corespondences are related to the classical correspondence (Cps). The left-to-right implication of (Cps1) and the right-to-left implication of (CA) imply the right-to-left implication of (Cps). The right-to-left implication of (Cps1) and the left-to-right implication of (CA) imply left-to-right implication of (Cps). Examples of the correspondences of these types can be found in [JO06]

3 Applications to Reasoning with Incomplete Information and Data Analysis

The general framework of discrete duality and duality via truth outlined above may be applied to various classes of lattices with operators. In [ORD06] a duality via truth framework is presented and illustrated with four case studies. The classical dualities for Boolean algebras with a possibility operator and for

Boolean algebras with a sufficiency operator are formulated in the form of a discrete duality and duality via truth. Then these results are extended to discrete dualities and dualities via truth for two classes of information algebras arising from information systems. The class of weak similarity algebras is an axiomatic extension of the class of Boolean algebras with a family of possibility operators and the class of strong right orthogonality (or in other words strong disjointness) algebras is an axiomatic extension of the class of Boolean algebras with a family of sufficiency operators. These two classes of information algebras provide a formal means for reasoning about and computing generalized approximation operations in the rough set style [Paw91] determined by similarity relations or their complements.

The framework of discrete duality is also relevant for formal concept analysis [Wil82, GaW99]. The theory of formal concept analysis provides a means of data analysis and discovery of concepts from data structures which are referred to as contexts. Contexts may be identified with information systems whose attributes are binary, in the sense of being features an object may or may not have. In [ORe07b] a class of sufficiency algebras derived from contexts is introduced and referred to as context algebras. A discrete duality and duality via truth for the class of context algebras is developed. These results provide the tools for solving various problems that can be specified within the framework of formal concept analysis e.g., finding extensions (resp. intensions) of concepts once their intensions (resp. extensions) are given; proving implications of sets of attributes; proving entailment of implications.

Many other discrete duality and duality via truth results can be found in the literature. Most of them concern not necessarily distributive lattices with operators. Various types of modal operators (possibility, necessity, sufficiency, dual sufficiency) are dealt with in [OV05]. These operators may be seen as generalizations of rough set style approximation operators. Several kinds of negations on lattices are treated in [DOvA06a, DOvA06b]. Relation algebra operators on lattices are studied in [DOR06]. Residuated lattices and their axiomatic extensions corresponding to substructural logics and some fuzzy logics are studied within the framework of discrete duality in [OR06, OR07]. In the field of distributive lattices, discrete dualities for Heyting algebras with operators (various types of modal operators and negations) are presented in [ORe07a].

References

[CLP91] Cignoli, R., Laflace, S., Petrovich, A.: Remarks on Priestley duality for distributive lattices. Order 8, 299–315 (1991)

[DOR06] Düntsch, I., Orłowska, E., Radzikowska, A.: Lattice-based relation algebras II. LNCS (LNAI), vol. 4342, pp. 267–289. Springer, Heidelberg (2006)

[DORV05] Düntsch, I., Orłowska, E., Radzikowska, A., Vakarelov, D.: Relational representation theorems for some lattice-based structures. Journal of Relational Methods in Computer Science 1, 132–160 (2005)

[DOvA06a] Dzik, W., Orłowska, E., van Alten, C.: Relational representation theorems for general lattices with negations. LNCS, vol. 4136, pp. 162–176. Springer, Heidelberg (2006)

[DOvA06b] Dzik, W., Orłowska, E., van Alten, C.: Relational representation theorems for lattices with negations: A survey. LNCS (LNAI), vol. 4342, pp. 245–266. Springer, Heidelberg (2006)

[GaW99] Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin (1999)

[Gol88] Goldblatt, R.: On closure under canonical embedding algebras, Colloquia Mathematica Societatis Janos Bolyai, Algebraic Logic, Budapest, Hungary, vol. 54, pp. 217–229 (1988)

[Gol89] Goldblatt, R.: Varieties of complex algebras. Annals of Pure. and Applied Logic 44, 173–242 (1989)

[JO06] Järvinen, J., Orłowska, E.: Relational correspondences for lattices with operators. LNCS, vol. 3929, pp. 134–146. Springer, Heidelberg (2006)

[JT51] Jónson, B., Tarski, A.: Boolean algebras with operators, Part I American Journal of Mathematics 73, 891–939 (1951) Part II ibidem 74, 127–162 (1952)

[Mak72] Maksimova, L.L.: Pretabular superintuitionistic logics. Algebra and Logic 11(5), 558–570 (1972)

[Mak75] Maksimova, L.L.: Pretabular extensions of the Lewis' logic S4. Algebra and Logic 14(1), 28–55 (1975)

[OR06] Orłowska, E., Radzikowska, A.: Relational representability for algebras of substructural logics. LNCS, vol. 3929, pp. 212–224. Springer, Heidelberg (2006)

[OR07] Orłowska, E., Radzikowska, A.: Representation theorems for some fuzzy logics based on residuated non-distributive lattices (submitted)

[ORe07a] Orłowska, E., Rewitzky, I.: Discrete duality for Heyting algebras with operators preprint) (2007)

[ORe07b] Orłowska, E., Rewitzky, I.: Discrete duality for context and concepts from formal concept analysis (submitted 2007)

[ORD06] Orłowska, E., Rewitzky, I., Düntsch, I.: Relational semantics through duality. LNCS, vol. 3929, pp. 17–32. Springer, Heidelberg (2006)

[ORe05] Orłowska, E., Rewitzky, I.: Duality via Truth: semantic frameworks for lattice-based logics. Logic Journal of the IGPL 13, 467–490 (2005)

[OV05] rłowska, E. and Vakarelov, D.: Lattice-based modal algebras and modal logics. In: Hajek, P., Valdés-Villanueva, L.M., Westerstahl, D. (eds.): Logic, Methodology and Philosophy of Science. Proceedings of the 12th International Congress, 2005, King's College London Publications, pp. 147–170 (2005)

[Paw91] Pawlak, Z.: Rough Sets - Theoretical Aspects of Reasoning about Knowledge. Kluwer, Dordrecht (1991)

[Pri70] Priestley, H.A.: Representation of distributive lattices by means of ordered Stone spaces. Bulletin of the London Mathematical Society 2, 186–190 (1970)

[Pri72] Priestley, H.A.: Ordered topological spaces and the representation of distributive lattices. Proceedings of the London Mathematical Society 24, 507–530 (1972)

[Sto36] Stone, M.: The theory of representations of Boolean algebras. Transactions of the American Mathematical Society 40, 37–111 (1936)

[Urq78] Urquhart, A.: A topological representation theory for lattices. Algebra Universalis 8, 45–58 (1978)

[Wil82] Wille, R.: Restructuring lattice theory: An approach based on hierarchies of concepts. In: Rival, I. (ed.): Ordered sets, NATO Advanced Studies Institute, Reidel, Dordrecht, vol. 83, pp. 445–470 (1982)

Toward Approximate Adaptive Learning

James F. Peters

Department of Electrical and Computer Engineering,
University of Manitoba
Winnipeg, Manitoba R3T 5V6 Canada
jfpeters@ee.umanitoba.ca

Abstract. The problem considered in this paper is how the classification of observed behaviour of organisms can be used to influence adaptive learning, beneficially. The solution to this problem hearkens back to the pioneering work during the 1980s by Zdzisław Pawlak and others on classification of objects and approximation spaces, where elementary sets of equivalent objects a framework for perceptions concerning observed behaviours. The seminal work by Oliver Selfridge and Chris J.C.H. Watkins on delay rewards and adaptive learning, also during the 1980s, combined with more recent work on reinforcement learning provide a basis for the forms of adaptive learning introduced in this article. In addition, recent work on approximation spaces has led to what is known as approximate adaptive learning. This article presents two forms of run-and-twiddle (RT) adaptive learning, each using the Watkins' stopping time strategy to mark the end of an episode. Twiddling amounts to adjusting what one does to achieve a better result. This becomes more apparent in approximate RT adaptive learning introduced in this article, where a record of observed behaviour patterns during each episode recorded in an ethogram makes it possible to define a pattern-based learning rate in the context of approximation spaces. Both forms of adaptive learning are actor-critic methods. The contribution of this article is the introduction of two forms of adaptive learning with Watkins' stopping time strategy with differential discount on returns in both cases and differential learning rate for adaptive learning in the context of approximation spaces.

Keywords: Actor-critic, adaptive learning, approximation space, behaviour pattern, perception, stopping time.

> *An approximation space ... serves as a formal counterpart of perception ability or observation.*
> – Ewa Orłowska, March, 1982.

1 Introduction

The problem considered in this paper is how the classification of observed behaviour of organisms can be used to influence adaptive learning, beneficially. The term *organism*, in general, is understood in Whitehead's sense as something that

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 57–68, 2007.

emerges from (belongs to) the world [43]. The solution to this problem hearkens back to the pioneering work by Zdzisław Pawlak and others on classification of objects and approximation spaces (see, *e.g.*, [4,13,9,15,16,22,33,34,39]), work on delayed rewards and adaptive learning by Oliver Selfridge and C.J.C.H. Watkins also during the 1980s (see, *e.g.*, [31,40]), extensive work on reinforcement learning (see, *e.g.*, [2,5,38,29,42]), and recent work on reinforcement learning and intelligent systems in the context of approximation spaces (see,*e.g.*, [24,25,23, 17,18,20,21,33,34]). This article presents two forms of run-and-twiddle (RT) adaptive learning, each using the Watkins' stopping time strategy to mark the end of an episode. Twiddling amounts to adjusting what one does to achieve a better result. This becomes more apparent in approximate RT adaptive learning introduced in this article, where a record of observed behaviour patterns tabulated in an ethogram [24,26] during each episode makes it possible to consider a pattern-based learning rate defined in the context of an approximation space. Both forms of adaptive learning introduced in this article are variant actor-critic methods, where action discounting as well as learning rate are defined relative to temporal differences. The contribution of this article is the introduction of two forms of adaptive learning that construct a semi-martingale with Watkins' stopping time strategy with differential discount on returns in both cases and differential learning rate for adaptive learning in the context of approximation spaces.

This article is organized as follows. An approach to RT adaptive learning is presented in Sect. 2. A refinement of the generalized approximation space model is given in Sect. 3. Approximate RT adaptive learning is introduced in Sect. 4.

2 Adaptive Learning

Watson [40] suggests using the value of a state $V(s)$ as the basis for an adaptive control strategy used by an organism to determine what to do next. This strategy can be summarized intuitively as follows.

1. **Estimate.** *If things are expected to improve or stay the same, then carry on with the same action.*
2. **Twiddle.** *If things are expected to get worse, then search for a more promising action.*
3. **End of Episode.** *If things are expected to get worse, regardless which possible action we choose, then that marks the end of an episode.* This is analogous to a situation faced by a gambler who either withdraws from the game because the expected return is not favorable or bets based on luck and stands a chance of losing [3]. The form of adaptive learning in this paper implicitly constructs a semi-martingale [3], where an episode continues as long as $V(s) \leq V(s')$, *i.e.*, $E[R_a] \leq E[R_{a'}]$ based on Monte Carlo estimates [7,30] of $V(s), V(s')$ for actions a, a' in states s, s', respectively[1].

[1] $V(s)$ (value of the current state s) is defined in terms of $E[R_a]$, the expected value of return R_a for an action a. $V(s')$ denotes the value of next state s' following s.

This control strategy was originally suggested by Oliver Selfridge in 1984 [31] and elaborated in the context of the value of a state and Monte Carlo methods by Chris J.C.H. Watkins in 1989 [40]. Selfridge called this a *run-and-twiddle* (RT) strategy, which he based on observations of the behavior of E. coli bacteria, male silk moths, and ants.

The notion of a stochastic process and what known as semi-martingales are important in RT adaptive learning introduced in this article.

Definition 1. Stochastic Process
A stochastic process is any family of random variables $\{X_t, t \in T\}$ *[3]. In practice,* X_t *is an observation at time* t*. A random variable (r.v.)* X_t *is a real-valued function* $X : \Omega \to \Re$ *defined on* $(\Omega, \mathcal{F})$*, where* $\Omega, \mathcal{F}$ *is sample space and family of events, respectively [8,44].*

It can be shown that during each episode of RT adaptive learning, what is known as a semi-martingale is constructed. Semi-martingales were introduced by Doob during the early 1950s [3] and elaborated by many others (see, *e.g.*, [8,44]).

Definition 2. Semi-Martingale
A semi-martingale is a stochastic process $\{X_t, t \in T\}$ *such that*

$$E[X_t] \leq E[X_{t+1}],$$

where $E[|X_t|] < \infty$*.*

The form of semi-martingale we have in mind is $\{R_t, t \in T\},\ E[R_t] \leq E[R_{t+1}]$, where R_t is the return on a sequence of actions at time t during an episode.

2.1 Toward RT Adaptive Learning

The basic framework for an approach to a run-and-twiddle (RT) form adaptive learning is shown in Fig. 1, where the conventional framework for actor-critic learning has been changed. Instead of the usual temporal difference (TD) δ term [38,41], a TD γ is source of input to a critic in evaluating observed action-rewards[2]. The policy structure enforced during adaptive learning is an actor, since the selection of an action a in each state s is determined by a policy $\pi(s, a)$. The estimated value function $V(s)$ serves a critic during adaptive learning. Twiddling begins at the end of each episode[3], where the actions within an episode are discontinued as a result of some halting condition being satisfied.

An elaborate form of twiddling is possible by recording observed behaviours during an episode and constructing what is known as a rough ethogram. A *rough ethogram* is a decision table that records acceptable as well as unacceptable behaviour patterns of organisms [26]. It will become apparent that an ethogram represents a decision system, where each possible behaviour leading from the current state to a new state is evaluated relative to an action-selection policy.

[2] TD γ denotes the rate of change of action rewards.

[3] *i.e.*, an episode is constituted by a sequence of actions that ends in a terminal state.

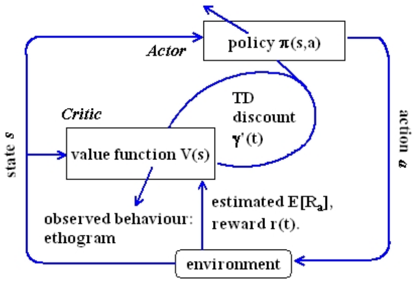


Fig. 1. Basic Framework for Adaptive Learning

That is, among all of the possible actions in a state, an action a that has been selected represents a perceptual judgement *accept* a based on a perception that the performance of a conforms to a standard more than the other possible actions, which is explained in the sequel. By the same token, an ethogram provides a record of each action b deemed unacceptable and a corresponding perceptual judgement *reject* b.

1. **Reward signal:** Define action a in terms of a reward $r(t)$ as a function representing a signal observed at time t_i, which results from interaction with the environment as a result of performing some action a at time t_{i-1}[4]. Then associate with each action $a(t)$ a discounted reward $r(t)$ at time t, namely, $\gamma'(t)r(t)$, where $\gamma(t)$ is a discount function and $\gamma'(t)$ denotes the differential of $r(t)$. It is important to define a reward function $r(t)$ that reflects the form of the signal produced by each action.
2. **Discount γ:** Either choose fixed $\gamma(t)$, *e.g.*, $\gamma(t) = 1$, or put $\gamma(t) = r(t)$ and obtain the differential

$$\gamma'(t) = \frac{d(r(t))}{dt}\Bigg|_{t \leftarrow t_i} \approx \frac{|r(t_i) - r(t_{i-1})|}{|t_i - t_{i-1}|}.$$

   In other words, let the value of γ vary over time instead of using a fixed value of γ that diminishes (*i.e.*, monotonically decreases) over time[5]. The critic in Fig. 1 is influenced by a Temporal Difference (TD) discount γ, which replaces the usual TD δ term (see, *e.g.*, [38,42]). The discount factor reflects the rate of change of the signal $r(t)$ coming from the environment at time t.
3. **Return:** Let $E[R_t], r_a, t_i$ denote expected return at time t, reward for action a, elapsed time at step i during an episode, respectively. Define $V(s_t) = E[R_t] \approx \frac{1}{n}\sum_{i=1}^{n} \gamma'(t_i) \cdot r_a(t_i)$, where value of state $V(s)$ is estimated over n time steps for each action a in state s. The assumption made here is that a reward r_t is a r.v. and, as a consequence, return R_t is a r.v. and $\{R_t, t \in T\}$

[4] t can be viewed as the elapsed time since the start of an episode.

[5] The form of discount factor introduced in this paper differs from what was originally suggested by Watkins [40] in estimating return R_t at time t, where $R_t = r_1 + \gamma r_2 + \cdots + \gamma^{t-1} r_t, \gamma \in [0,1]$.

is a stochastic process, where R_t is the return computed at time t for each state during an episode and T is a set of episode times. It is also the case that the $Pr(R_t = \omega)$ is unknown for $\omega \in \Omega$. For this reason, $E[R_t]$ is estimated using a Monte Carlo method [7,30] (for a detailed explanation, see [25]).

Theorem 1. Adaptive Learning Semi-martingale.
The RT form of adaptive learning constructs a semi-martingale.

Proof. An episode continues as long as $V(s) \leq V(s')$. Let t_i denote elapsed time t at the start of i^{th} state during an episode and let s' denote the state immediately following state s. Each time an episode continues after finding that the condition $V(s') > V(s)$ is satisfied at time t_n, another term is added to a sequence of estimates of $V(s)$ at time t_n, namely, $V(s') \approx E[X_{t_{n+1}}]$, namely,

$$E[X_{t_1}] \leq E[X_{t_2}], \ldots, \leq E[X_{t_n}] \leq E[X_{t_{n+1}}]. \qquad \square$$

An important problem to consider in constructing semi-martingales is a stopping time, *i.e.*, a time $\mathcal{T}$ when a semi-martingale ends. The notion of a stopping time can be explained in general.

Definition 1. Stopping Time. *A stopping time results from a strategy for determining when to stop a sequence based only on the outcomes seen so far [8].*

Axiom 1. Discount Rate. *During each episode, $\gamma'(t) < \varepsilon$ for any given threshold $\varepsilon > 0$ and for sufficiently large t. This means that $|r(t_{i+1}) - r(t_i)| < \varepsilon|t_{i+1} - t_i|$ for sufficiently large i, e.g., $i > n_{large}$.*

Theorem 2. Adaptive Learning Semi-martingale with Stopping Time.
In RT adaptive learning, (1) a semi-martingale constructed during each episode has a stopping time, and (2) $E[R_{t_n}] > E[R_{t_{n+1}}]$ occurs at some time t_n, (3) each adaptive learning episode has finite duration and each semi-martingale has a finite number of terms.

Proof. During adaptive learning, construction of a semi-martingale ends whenever Watkins' condition $V(s') > V(s)$ is not satisfied. Hence, (1) holds, *i.e.*, from Def. 1, Watkins' condition provides a stopping time strategy. (2) From Ax. 1, $\gamma'(t) \rightarrow 0$ during each episode. Hence, $E[R_{t_n}] > E[R_{t_{n+1}}]$ occurs at some time t_n, since the estimated value of $E[R_{t_{n+1}}]$ gets smaller than $E[R_{t_n}]$ for $\lim\limits_{i \rightarrow n_{large}} \gamma'(t_i) < \varepsilon$ for sufficiently large i. (3) *sunset* $\rightarrow 0$ during each episode in Alg 1 and Alg. 2, which guarantees that each episode has finite duration. Hence, each semi-martingale constructed during an adaptive learning episode has finite length. $\square$

2.2 Adaptive Control Algorithm

The run-and-twiddle control strategy is given a more formal representation by Watkins [40], p. 67. Let $a(x_t), s, s'$ denote action of object x at time t in state s, current state and next state, respectively. A representation of the adaptive

Algorithm 1. RT Adaptive Learning

```
Input  : States $s \in S$, Actions $a \in A$, Objects $x \in U$, $V(s)$.
Output: Semi-martingale, i.e., $\{R_t, t \in T\}$.
while True do
    Begin episode;
    Initialize policy $\pi(s,a), s, V(s), sunset \leftarrow maxTime, episode \leftarrow true$;
    Estimate $V(s') = E[R_t]$ for every $a$ leading from $s$ to $s'$;
    while $V(s) \leq V(s')$ do
        $V(s) \leftarrow V(s')$ ;
        Perform action $a$, observe $r(t)$ signal, compute $\gamma'(t)$;
        Update $a(x_t) \leftarrow \gamma'(t) \cdot r(t)$;
        Choose new $a$ from new $s$ according to policy $\pi(s,a)$ ;
        Estimate new $V(s') = E[R_t]$;
        $sunset \leftarrow sunset - 1$;
        if $sunset > 1$ then
            if $V(s') > V(s)$ then
                episode continues ;
            end
        else
            $episode \leftarrow false$ {publish $\{R_t, t \in T\}$} ;
        end
    end
end
```

learning method suggested by Selfridge is represented by Alg. 1. This algorithm reflects Selfridge's run-and-twiddle (RT) adaptive control strategy. In its simplest form, RT is a greedy method that works by steepest ascent hill-climbing, where an attempt is made to maximize return R over time by choosing the most promising action in each state. The *most promising action* a means that action a has the highest estimated expected return R_t at time t. Alg. 1 looks one step ahead in each state during an episode and takes the best pick among all possible actions for the next step.

3 Approximation Spaces

The original generalized approximation space (GAS) model [32] has recently been extended as a result of recent work on nearness of objects (see, *e.g.*, [6,17, 18,20,21,33,34]). A nearness approximation space (NAS) is a tuple

$$NAS = (U, A, N_r, \nu_B),$$

where U is a universe of objects, A, a set of probe functions, N_r, a family of neighbourhoods and ν_B is an overlap function defined by

$$\nu_B : \mathcal{P}(U) \times \mathcal{P}(U) \longrightarrow [0, 1],$$

where $\mathcal{P}(U)$ is the powerset of U. The overlap function ν_B maps a pair of sets to a number in $[0, 1]$ representing the degree of overlap between the sets of objects with features defined by $B \subseteq A$ and $\mathcal{P}(U)$ is the powerset of U [35]. For each subset $B \subseteq A$ of probe functions, define the binary relation $\sim_B = \{(x, x') \in U \times U : \forall f \in B, f(x) = f(x')\}$. Since each $\sim_B$ is, in fact, the usual Ind_B (indiscernibility) relation, for $B \subset F$ and $x \in U$, let $[x]_B$ denote the equivalence class containing x, *i.e.*,

$$[x]_B = \{x' \in U : \forall f \in B, f(x') = f(x)\} \subseteq U.$$

If $(x, x') \in \sim_B$ (also written $x \sim_B x'$), then x and x' are said to be *indiscernible* with respect to all feature probe functions in B, or simply, B*-indiscernible.* Then define a family of neighborhoods $N_r(A)$, where

$$N_r(A) = \bigcup_{B_r \subseteq P_r(A)} [x]_{B_r},$$

where $P_r(A) = \{B \subseteq A \mid |B| = r\}$ for any r such that $1 \leq r \leq |A|$. That is, r denotes the number of features used to construct families of neighborhoods. For the sake of clarity, we sometimes write $[x]_{B_r}$ to specify that the equivalence class represents a neighborhood formed using r features from B. Families of neighborhoods are constructed for each combination of probe functions in B using $\binom{|B|}{r}$, *i.e.*, $|B|$ probe functions taken r at a time. Information about a sample $X \subseteq U$ can be approximated from information contained in B by constructing a $N_r(B)$-lower approximation

$$N_r(B)_*X = \bigcup_{x:[x]_{B_r} \subseteq X} [x]_{B_r},$$

and a $N_r(B)$-upper approximation

$$N_r(B)^*X = \bigcup_{x:[x]_{B_r} \cap X \neq \emptyset} [x]_{B_r}.$$

Then $N_r(B)_*X \subseteq N_r(B)^*X$ and the boundary region $BND_{N_r(B)}(X)$ between upper and lower approximations of a set X is defined to be the complement of $N_r(B)_*X$, *i.e.*

$$BND_{N_r(B)}(X) = N_r(B)^*X \backslash N_r(B)_*X = \{x \in N_r(B)^*X \mid x \notin N_r(B)_*X\}.$$

A set X is termed a "near set" relative to a chosen family of neighborhoods $N_r(B)$ *iff* $|BND_{N_r(B)}(X)| \geq 0$. This means that, relative to B, every rough set is a near set but not every near set is a rough set. Object recognition and the problem of the nearness of objects have motivated the introduction of near sets (see, *e.g.*, [17,20]).

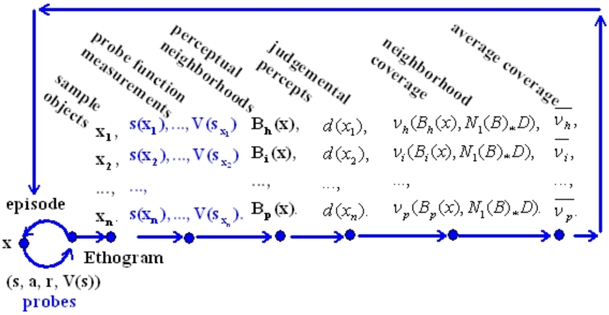

sample objects
probe function measurements
perceptual neighborhoods
judgemental percepts
neighborhood coverage
average coverage
x_1, $s(x_1), \ldots, V(s_{x_1})$ $B_h(x)$, $d(x_1)$, $\nu_h(B_h(x), N_1(B)_*D)$, $\overline{\nu}_h$,
x_2, $s(x_2), \ldots, V(s_{x_2})$ $B_i(x)$, $d(x_2)$, $\nu_i(B_i(x), N_1(B)_*D)$, $\overline{\nu}_i$,
…, …, …, …, …, …,
x_n. $s(x_n), \ldots, V(s_{x_n})$. $B_p(x)$. $d(x_n)$. $\nu_p(B_p(x), N_1(B)_*D)$. $\overline{\nu}_p$.
episode
x
Ethogram
(s, a, r, V(s))
probes

Fig. 2. Approximate Adaptive Learning Cycle

3.1 Percepts and Perception

The set $N_r(B)$ contains a set of percepts. A *percept* is a byproduct of perception, *i.e.*, something that has been observed [10]. For example, a member of $N_r(B)$ represents *what has been perceived about objects belonging to a neighborhood, i.e.*, observed objects with matching probe function values. Collectively, $N_r(B)$ represents a *perception*, a product of perceiving. Perception is defined as the extraction and use of information about one's environment [1]. This basic idea is represented in the *sample objects*, *probe function measurements*, *perceptual neighborhoods* and *judgemental percepts* columns in Fig. 2[6]. In this article, we focus on the perception of acceptable objects.

3.2 Sensing, Classifying, and Perceptual Judgement

Sensing provides a basis for probe function measurements commonly associated with features such as colour, contour, shape, arrangement, entropy, and so on [12,22]. A probe function can be thought of as a model for a sensor. Classification combines evaluation of a disposition of sensor measurements with judgement (apprehending the significance of a vector of probe function measurements for an observed object). The result is a higher level percept, which has been traditionally called a decision. In the context of percepts, the term *judgement* means a conclusion about an object's measurements rather than an abstract idea. This form of judgement is considered *perceptual.* Perceptual judgements provide a basis for the formulation of abstract ideas (models of perception, rules) about a class (type) of objects. Let D denote a feature called *decision* with a probe $d_B : X \times B \longrightarrow \{0, 1\}$, where X denotes a set of sample objects; B, a set of probe functions; 0, "reject perceived object" and 1, "accept perceived object". A set of objects d with matching perceptual judgements (*e.g.*, $d_B(x) = 1, x \in X$ for an acceptable object) is a mathematical model representing the abstract notion *acceptable.*

For each possible feature value j of a and $x \in U$, put $B_j(x) = [x]_B$ if, and only if, $a(x) = j$, and call $B_j(x)$ an *action block.* Put $\mathcal{B} = \{B_j(x) : a(x) = j, x \in U\}$,

[6] Subscripts h, i, p denote probe function values for a single feature, *i.e.*, where r = 1.

Algorithm 2. Approximate Adaptive Learning

```
Input  : States $s \in S$, Actions $a \in A$, Objects $x \in U$.
Output: Ethogram resulting from policy $\pi(s, a)$.
while True do
    Begin episode;
    Initialize $\bar{\nu_a}'(t)$, policy $\pi(s, a), s, V(s), sunset \leftarrow maxTime, episode \leftarrow true$;
    Insert experimental $(x, s, a, r, V(s), d(x))$ rows in ethogram, then continue ;
    Estimate $V(s') \leftarrow E[R_t]$;
    while $V(s) \leq V(s')$ do
        Perform action $a$ based on $\pi(s, a)$, observe $r(t)$ signal, compute $\gamma'(t)$;
        Update $a(x_t) \leftarrow \gamma'(t) \cdot r(t)$;
        Choose new $a$ from new $s$ according to policy $\pi(s, a)$ ;
        Estimate $V(s') \leftarrow E[R_t]$;
        $V(s) \longleftarrow V(s) + \bar{\nu_a}\ '(t) \cdot [r + max_a\{V(s')\} - V(s)]$;
        $sunset \leftarrow sunset - 1$;
        if $sunset > 1$ then
            if $V(s') > V(s)$ then
                Episode continues ;
                Add $(x, s, a, r, V(s), d(x))$ to ethogram ;
            end
        else
            $episode \leftarrow false$ {publish constructed ethogram} ;
            Compute learning rate $\bar{\nu_a}\ '(t)$ using ethogram, (1), & (2);
        end
    end
end
```

a set of blocks that "represent" action $a(x) = j$. Define $\bar{\nu}_a(t)$ (average rough coverage)[7] with respect to an action $a(x) = j$ at time t in (1).

$$\bar{\nu}_a(t) = \frac{1}{|\mathcal{B}|} \sum_{B_j(x) \in \mathcal{B}} \nu\left(B_j(x), N_r(B)_* D\right). \tag{1}$$

4 Approximate RT Adaptive Learning

Based on the introduction of families of neighbourhoods, there are different forms of adaptive learning that is influenced by the perceived behaviours recorded in episode ethograms. A behaviour is defined by the tuple

$$(s, a, r, V(s)),$$

where $V(s)$ is the estimated value of expectation $E[R_t]$. A Monte Carlo method [7,30] is used to estimate $E[R_t]$, which, in its simplest form, is a running average of the rewards received up to the current state.

[7] $\bar{\nu}_a(t)$ is computed at the end of each episode using an ethogram that is part of the adaptive learning cycle shown in Fig. 2.

The differential $\bar{\nu_a}\ '(t)$ of $\bar{\nu_a}(t)$[8] takes the place of learning rate α in Q-learning [40], where $\bar{\nu_a}\ '(t)$ reflects the rate of change of average action acceptability across adjacent episodes. Starting with the end of the second episode during approximate adaptive learning, it is possible to define a learning rate $\bar{\nu_a}\ '(t)$ shown in (2).

$$\bar{\nu}'(t) = \left.\frac{d(\bar{\nu_a}(t))}{dt}\right|_{t \leftarrow t_i} \approx \frac{|\bar{\nu_a}(t_i) - \bar{\nu_a}(t_{i-1})|}{|t_i - t_{i-1}|}, \tag{2}$$

where $\bar{\nu_a}(t_i), \bar{\nu_a}(t_{i-1})$ is the average action coverage at times t_i, t_{i-1} at the end of the current and the previous episodes, respectively. In other words, at the end of each episode, $\bar{\nu}'(t)$ is refreshed to reflect a varying learning rate (see Alg. 2). Other forms of Alg. 2 are possible, if we consider combinations of features in addition to the single-feature case, where multiple-feature families of neighborhoods are used to estimate average coverage. At present, a number of fairly intensive experiments with approximate adaptive learning in colonies of organisms (*e.g.*, fish and ants) and in computer vision, are being carried out [18,19].

5 Conclusion

This article considers a perception-based approach to adaptive learning. The early work of Zdzisław Pawlak and others on classification of objects and approximation spaces during the 1980s as well as more recent work on approximation spaces by Andrzej Skowron and Jarosław Stepaniuk provide a framework for observing the returns on episodic behaviour during learning. This work has also benefited from the work on adaptive learning by Oliver Selfridge and Chris J.C.H. Watkins, also during the 1980. It was Watkins who suggested a stopping time strategy for episodic behaviour based on the estimated value of state. The work on semi-martingales by Leo Doob introduced during the 1950s has also been helpful in the interpretation of what is happening during what is known as run-and-twiddle (RT) adaptive learning. It has been shown that a semi-martingale is constructed with a stopping time strategy during each adaptive learning episode. Future work will include various families of neighborhoods as a basis for defining a learning rate.

Acknowledgements

The author gratefully acknowledges the suggestions and insights by Andrzej Skowron, Jarosław Stepaniuk, Jay Kraut, Shabnam Shahfar, Christopher Henry, Dan Lockery and David Gunderson concerning topics in this paper. This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grant 185986.

[8] Average rough coverage $\bar{\nu_a}(t)$ is computed at time t marking the end of episode.

References

1. Audi, R. (ed.): The Cambridge Dictionary of Philosophy. 2nd, Cambridge University Press, UK (1999)
2. Berenji, H.R.: A convergent actor–critic-based FRL algorithm with application to power management of wireless transmitters. IEEE Trans. on Fuzzy Systems 11/4, 478–485 (2003)
3. Doob, J.L.: Stochastic Processes. Chapman & Hall, London (1953)
4. Gomolińska, A.: Approximation spaces based on similarity and dissimilarity. In: Lindemann, G., Schlingloff, H., Burkhard, H.-D., Czaja, L., Penczek, W., Salwicki, A., Skowron, A., Suraj, Z. (eds.): Concurrency, Specification and Programming (CS&P'06). Infomatik-Berichte, Nr. 206, pp. 446–457 (2006)
5. Geramifard, A., Bowling, M., Sutton, R.S.: Incremental least-squares temporal difference learning. In: Proc. 21st National Conf. on AI (AAA06), pp. 356–361 (2006)
6. Henry, C., Peters, J.F.: Image Pattern Recognition Using Approximation Spaces and Near Sets. In: Proc. 2007 Joint Rough Set Symposium (JRS07), Toronto, Canada 14-16 May 2007 (2007)
7. Hammersley, J.M., Handscomb, D.C.: Monte Carlo Methods. Methuen & Co Ltd, London (1964)
8. Mitzenmacher, M., Upfal, E.: Probability and Computing. Randomized Algorithms and Probabilistic Analysis. Cambridge University Press, New York (2005)
9. Orłowska, E.: Semantics of Vague Concepts, Applications of Rough Sets, Institute for Computer Science, Polish Academy of Sciences, Report 469, March 1982 (1982)
10. The Oxford English Dictionary. Oxford University Press, London (1933)
11. Pal, S.K., Polkowski, L., Skowron, A. (eds.): Rough-Neural Computing: Techniques for Computing with Words. Cognitive Technologies. Springer, Heidelberg (2004)
12. Pavel, M.: Fundamentals of Pattern Recognition, 2nd edn. Marcel Dekker, Inc, NY (1993)
13. Pawlak, Z.: Classification of Objects by Means of Attributes, Institute for Computer Science, Polish Academy of Sciences, Report 429, March 1981 (1981)
14. Pawlak, Z.: Rough Sets, Institute for Computer Science, Polish Academy of Sciences, Report 431, March 1981 (1981)
15. Pawlak, Z.: Rough sets. International J. Comp. Inform. Science 11, 341–356 (1982)
16. Pawlak, Z., Skowron, A.: Rudiments of rough sets, Information Sciences, ,177 (1), 3–27 (2007) ISSN 0020-0255
17. Peters, J.F.: Near sets. Special theory about nearness of objects. Fundamenta Informaticae 75(1-4), 407–433 (2007)
18. Peters, J.F.: Near Sets. Toward Approximation Space-Based Object Recognition. In: Proc. 2007 Joint Rough Set Symposium (JRS07), Toronto, Canada 14-16 May, 2007 (2007)
19. Peters, J.F., Borkowski, M., Henry, C., Lockery, D., Gunderson, D., Ramanna, S.: Line-Crawling Bots That Inspect Electric Power Transmission Line Equipment. In: Proc. 3rd Int. Conf. on Autonomous Robots and Agents (ICARA 2006), Palmerston North, NZ, 2006, pp. 39–44 (2006)
20. Peters, J.F., Skowron, A., Stepaniuk, J.: Nearness in approximation spaces. Lindemann, G., Schlilngloff, H., et al. (eds.): Proc. Concurrency, Specification & Programming (CS&P'2006). Informatik-Berichte Nr. 206, Humboldt-Universität zu Berlin, 2006, pp. 434–445 (2006)
21. Peters, J.F., Skowron, A., Stepaniuk, J.: Nearness of objects: Extension of approximation space model. Fundamenta Informaticae 77 (in press) (2007)

22. Peters, J.F.: Classification of objects by means of features. In: Kacprzyk, J., Skowron, A.: Proc. Special Session on Rough Sets, IEEE Symposium on Foundations of Computational Intelligence (FOCI07) (2007)
23. Peters, J.F., Henry, C.: Approximation spaces in off-policy Monte Carlo learning, Engineering Applications of Artificial Intelligence (in press) (2007)
24. Peters, J.F.: Rough ethology: Toward a Biologically-Inspired Study of Collective behaviour in Intelligent Systems with Approximation Spaces. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 153–174. Springer, Heidelberg (2005)
25. Peters, J.F., Henry, C.: Reinforcement learning with approximation spaces. Fundamenta Informaticae 71(2-3), 323–349 (2006)
26. Peters, J.F., Henry, C., Ramanna, S.: Rough Ethograms: Study of Intelligent System behaviour. In: Kłopotek, M.A., Wierzchoń, S., Trojanowski, K. (eds.): New Trends in Intelligent Information Processing and Web Mining (IIS05), Gdańsk, Poland, June 13-16, 2005, pp. 117–126 (2005)
27. Polkowski, L.: Rough Sets. Mathematical Foundations. Springer, Heidelberg (2002)
28. Polkowski, L., Skowron, A. (eds.): Rough Sets in Knowledge Discovery 2, Studies in Fuzziness and Soft Computing, vol. 19. Springer, Heidelberg (1998)
29. Precup, D., Sutton, R.S., Paduraru, C., Koop, A., Singh, S.: Off-policy with recognizers. Advances in Neural Information Processing Systems, pp. 1–8 (2006)
30. Rubinstein, R.Y.: Simulation and the Monte Carlo Method. John Wiley & Sons, Toronto (1981)
31. Selfridge, O.G.: Some themes and primitives in ill-defined systems. In: Selfridge, O.G., Rissland, E.L., Arbib, M.A. (eds.) Adaptive Control of Ill-Defined Systems, Plenum Press, London (1984)
32. Skowron, A., Stepaniuk, J.: Generalized approximation spaces. In: Lin, T.Y., Wildberger, A.M (eds.): Soft Computing, Simulation Councils, San Diego, 18–21 (1995)
33. Skowron, A., Swiniarski, R., Synak, P.: Approximation spaces and information granulation. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 175–189. Springer, Heidelberg (2005)
34. Skowron, A., Stepaniuk, J., Peters, J.F., Swiniarski, R.: Calculi of approximation spaces. Fundamenta Informaticae 72(1-3), 363–378 (2006)
35. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27(2-3), 245–253 (1996)
36. Skowron, A., Stepaniuk, J.: Information granules and rough-neural computing. In: Pal et al. [11] (2204), pp. 43–84
37. Stepaniuk, J.: Approximation spaces, reducts and representatives. In [28], pp. 109–126
38. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA (1998)
39. Wolski, M.: Similarity as nearness: Information quanta, approximation spaces and nearness structures. In: Proc. CS&P 2006. Infomatik-Berichte, pp. 424–433 (2006)
40. Watkins, C.J.C.H.: Learning from Delayed Rewards. Ph.D. Thesis, supervisor: Richard Young. King's College, Cambridge University, May 1989 (1989)
41. Watkins, C.J.C.H., Dayan, P.: Reinforcement learning. Encyclopedia of Cognitive Science. Macmillan, UK (2003)
42. Wawrzyński, P.: Intensive Reinforcement Learning, Ph.D. dissertation, supervisor: Andrzej Pacut, Institute of Control and Computational Engineering, Warsaw University of Technology, May 2005 (2005)
43. Whitehead, A.N.: Process and Reality. Macmillan, UK (1929)
44. Williams, D.: Probability with Martingales. Cambridge University Press, UK (1991)

Granulation of Knowledge in Decision Systems: The Approach Based on Rough Inclusions. The Method and Its Applications

Lech Polkowski

Polish –Japanese Institute of Information Technology
Koszykowa 86 02008 Warszawa, Poland
polkow@pjwstk.edu.pl

Abstract. Rough set approach to knowledge entails its granulation: knowledge represented as a collection of classifications by means of indiscernibility of objects consists of indiscernibility classes that form elementary granules of knowledge. Granules of knowledge that emerge as unions of elementary granules are also characterized as exact concepts that are described with certainty. Relaxing of indiscernibility relations has led to various forms of similarity relations. In this lecture, we discuss the approach to similarity rooted in mereological theory of concepts, whose primitive notion is that of a rough inclusion. Rough inclusions are predicates/relations of a part to a degree. Partial containment is the basic underlying phenomenon related to uncertainty, therefore rough inclusions allow for a formalization of a wide spectrum of contexts in which reasoning under uncertainty is effected.

Granules are formed by means of rough inclusions as classes of objects close to a specified center of the granule to a given degree; formally, they resemble neighborhoods formed with respect to a certain metric. Classes of objects in turn are defined by the class operator borrowed from mereology. The usage of mereological techniques based on the notion of a part is justified by its greater elegance and transparency in comparison to the naive theory of concepts based on the notion of an element.

At IEEE GrC 2005, 2006 the Author put forth the idea of a granular information/decision system whose objects are granules formed from the original information/decision system; the idea was issued along with the hypothesis that granular systems at sufficiently large radii of granulation, should preserve information about objects coded in the attribute–value language to a sufficiently high degree. This idea is here discussed along with results of some tests that bear it out.

The second application that is reflected in the lecture is about missing values; the approach discussed here is also based on granulation and the idea is to absorb objects with missing values into granules of knowledge in order to replace in a sense the missing value with a defined one decided by the granule.

Keywords: granulation of knowledge, rough sets, rough inclusions, granular decision systems.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 69–79, 2007.

1 The Idea of Computing Via Rough Sets

Classical ideas about representations of uncertainty, expressed respectively by Gottlob Frege and Max Black, found realization respectively in rough and fuzzy concept theories. Despite their formal differences and distinct starting points, both compute with granules of objects: rough sets with indiscernibility classes of objects, fuzzy sets with inverse images of fuzzy membership functions.

Rough sets represent knowledge by means of *information systems*, i.e., pairs of the form (U, A) where U is a set of *objects* and A is a set of *attributes* with each $a \in A$ a mapping $a : U \rightarrow V_a$ on U into the value set V_a. Objects are coded by their *information sets* of the form $inf(u) = \{(a = a(u)) : a \in A\}$. Objects u, v with $inf(u) = inf(v)$ are called *indiscernible* and they are regarded as identical with respect to the given set A. The *B–indiscernibility relation relative to a set* $B \subseteq A$ is $ind(B) = \{(u, v) : \forall a \in B.a(u) = a(v)\}$. Classes $[u]_B = \{v : (u, v) \in ind(B)\}$ are *B–elementary granules* of knowledge. Their unions are *B–granules* of knowledge.

A formula $(a = v)$ is an *elementary descriptor*; *descriptors* are formed as the smallest set containing all elementary descriptors and closed under sentential connectives $\vee, \wedge, \neg, \Rightarrow$. The meaning $[a = v]$ of an elementary descriptor is defined as the set $\{u : a(u) = v\}$ and it is recursively extended to meaning of descriptors [8].

Decision systems are information systems of the form $(U, A \cup \{d\})$ with a singled out attribute d called the *decision* that does represent a description of objects by an external informed source (say, an expert). Description of d in terms of *conditional* attributes in the set A is effected by means of *decision rules* [8] of the form

$$\bigwedge_{a \in B} (a = v_a) \Rightarrow (d = v). \tag{1}$$

The rule (1) is *true* whenever $[\bigwedge_{a \in B}(a = v_a)] \subseteq [d = v]$; otherwise it is partially true; see, e.g., [10] for a review of this topic.

2 Granulation of Knowledge

The issue of granulation of knowledge as a problem on its own, has been posed by L.A.Zadeh [23]. The issue of granulation has been a subject of intensive studies within rough set community, as witnessed by a number of papers, see, e.g., [17], [18].

Granules defined by indiscernibility and their direct generalizations to various similarity classes of tolerance, asymmetric similarity relations and general binary relations were subject to an intensive research, see, e.g. [7], [22]. Granulation of knowledge by means of rough inclusions was studied in [16].

Granulation of knowledge and applications to knowledge discovery in the realm of approximation spaces were studied, among others, in [20].

A study of granule systems was also carried out in [11], [12], [13], [14], in order to find general properties of granules. In proofs of those properties, techniques of mereology were applied as more simple and elegant than those of naive set theory.

2.1 The Technique of Mereology

Fundamental in mereology [6] is the relation of a *part*, π, that given a universe U, does satisfy the following conditions ,

$$1.\neg(x\pi x).2.x\pi y \wedge y\pi z \Rightarrow x\pi z, \tag{2}$$

i.e., it is transitive (cond. 2) and irreflexive (cond. 1).

The notion of an *element*, associated with the part relation π, is expressed with the help of the notion of an ingredient ing_π, informally an "improper part",

$$x\ ing_\pi\ y \Leftrightarrow\ x\ \pi\ y\ \vee\ x = y. \tag{3}$$

Mereology is a theory of individual objects, that decompose into parts, and passing to it from Ontology - theory of distributive concepts, is realized by means of the *set/class operator* [6]; given a non–empty collection F of objects, i.e., an ontological concept F, the individual representing F is given as the class of F, $Cls_\pi F$, subject to the following conditions,

$$\begin{gathered} 1.\ u \in\ F \Rightarrow u\ ing_\pi\ Cls_\pi F. \\ 2.\ u\ ing_\pi\ Cls_\pi F \Rightarrow \forall v.[v\ ing_\pi\ u \Rightarrow \exists w,t.\ w\ ing_\pi\ v, w\ ing_\pi\ t, t \in\ F]. \end{gathered} \tag{4}$$

In the sequel, the subscript π will be mostly omitted.

In plain words, $ClsF$ consists of those objects whose each part has a part in common with an object in F; the reader will easily recognize that the union $\bigcup F$ of a family F of sets is the class of F with respect to the part relation $\subset$.

2.2 Rough Inclusions

A *rough inclusion* is a generic term introduced in [16] for a class of relations on the universe U; any rough inclusion μ is a ternary relation, a subset of the product $U \times U \times [0, 1]$; see [11], [12], [13], [14], for details and discussion along with the extensive reference list.

A *rough inclusion* $\mu_\pi(x, y, r)$, where x, y are individual objects, $r \in [0, 1]$, does satisfy the following requirements, relative to a given part relation π on a set U of individual objects,

$$\begin{gathered} 1.\ \mu_\pi(x, y, 1) \Leftrightarrow x\ ing_\pi\ y; \\ 2.\ \mu_\pi(x, y, 1) \Rightarrow [\mu_\pi(z, x, r) \Rightarrow \mu_\pi(z, y, r)]; \\ 3.\ \mu_\pi(x, y, r) \wedge s < r \Rightarrow \mu_\pi(x, y, s). \end{gathered} \tag{5}$$

2.3 Examples of Rough Inclusions

Apart from a general theory, we give here some examples of rough inclusions, cf. [11], [13], [14].

1. **Rough inclusions from Archimedean t–norms.** They are induced from Archimedean t–norms, see, e.g, [3], [10]. We describe the one we are going to use in the sequel. The Łukasiewicz t–norm
$$L(x, y) = max\{0, x + y - 1\}, \tag{6}$$
admits a characterization,
$$L(x, y) = g(f(x) + f(y)), \tag{7}$$
with $f = 1 - x = g$. We define the set, $DIS(u, v) = \{a \in A : a(u) \neq a(v)\}$, and its complement $IND(u, v) = U \times U \setminus DIS(u, v)$.

   We define the rough inclusion μ_L,
$$\mu_L(u, v, r) \Leftrightarrow g(\frac{|DIS(u, v)|}{|A|}) \geq, \tag{8}$$
i.e.,
$$\mu_L(u, v, r) \Leftrightarrow \frac{|IND(u, v)|}{|A|} \geq r. \tag{9}$$
The formula (9) witnesses that the reasoning based on the rough inclusion μ_L is the probabilistic one. At the same time, we have given a logical proof for formulas like (9) that are very frequently applied in Data Mining and Knowledge Discovery.

   μ_L is *transitive* [11]: *$\mu_L(u, v, r)$ and $\mu_L(v, w, s)$ imply that $\mu_L(u, w, L(r, s))$.*
2. **Rough inclusions and metrics.** For a metric $d(u, v)$ on the set of objects U, i.e., 1. $d(u, u) = 0$; 2. $d(u, v) = d(v, u)$; 3. $d(u, v) \leq d(u, w) + d(w, v)$, we let $\mu_d(u, v, r) \Leftrightarrow d(u, v) \leq 1 - r$. Then, *$\mu_d$ is a rough inclusion*, transitive with respect to the t–norm L.

   Conversely, consider a transitive symmetric rough inclusion μ_T; let $d_\mu(u, v) \leq r \Leftrightarrow \mu(u, v, 1-r)$. Then, clearly, $d_\mu(u, u) = 0$, $d_\mu(u, v) = d_\mu(v, u)$; concerning triangle inequality 3., if $d_\mu(u, v) \leq r$ and $d_\mu(v, w) \leq s$, then by transitivity of μ, $d_\mu(u, w) \leq 1 - T(1 - r, 1 - s) = S_T(r, s)$, where S_T is the t–*conorm*, induced by T, see,e.g., [10]; thus, d_μ is a generalized metric. Particular cases encompass: in case of $T = min$, $S_T = max$, hence $d_{min}(u, w) \leq max\{d_{min}(u, v), d_{min}(v, w)\}$, i.e., d_{min} is an Archimedean metric; in case of L, $S_L(r, s) = min\{1, r+s\} \leq r+s$, i.e., d_L is a metric satisfying 3., restricted by 1.

2.4 Granules Induced from Rough Inclusions

The general scheme of our own for inducing granules is as follows. We fix an information system (U, A), and a rough inclusion μ on U.

For an object u and a real number $r \in [0, 1]$, we define the granule $g_\mu(u, r)$ about u of the radius r, relative to μ, by letting,
$$g_\mu(u, r) \text{ is } ClsF(u, r), \tag{10}$$

where the property $F(u, r)$ is satisfied with an object v if and only if $\mu(v, u, r)$ holds.

It was shown, see [11], Theorem 4, that in case of a transitive μ,

$$v \ ing \ g_{\mu}(u, r) \Leftrightarrow \mu(v, u, r). \tag{11}$$

By (11), the granule $g_{\mu_L}(u, r)$ consists of objects v such that $\mu_L(v, u, r)$, i.e, $|IND(u, v)| \geq r \cdot |A|$;

For a given granulation radius r, and the rough inclusion μ_L, we form the collection $U^G_{r,\mu_L} = \{g_{\mu_L}(u, r)\}$.

3 Granular Decision Systems

The idea of a granular decision system was posed in [13]; for a given information system (U, A), a rough inclusion μ, and $r \in [0, 1]$, the new universe $U^G_{r,\mu}$ is given. We apply a strategy $\mathcal{G}$ to choose a covering $Cov^G_{r,\mu}$ of the universe U by granules from $U^G_{r,\mu}$.

We apply a strategy $\mathcal{S}$ in order to assign the value $a^*(g)$ of each attribute $a \in A$ to each granule $g \in Cov^G_{r,\mu}$: $a^*(g) = \mathcal{S}(\{a(u) : u \in g\})$. The granular counterpart to the information system (U, A) is a tuple $(U^G_{r,\mu}, \mathcal{G}, \mathcal{S}, \{a* : a \in A\})$; analogously, we define granular counterparts to decision systems by adding the factored decision $d*$.

4 Factoring Classifiers Through Granular Systems

As objects in a granule are related one to another by similarity, the granule as a whole should determine a new object; and a judiciously chosen set of the new objects should guarantee the satisfactory quality of classification [13]. To test the validity of this hypothesis, experiments have been carried out with real data sets. We select here the Primary tumor data set [21] and we test it with exhaustive algorithm of RSES package [19] and with LEM2 algorithm with the parameter p=0.5 [2], [19]. We adopt random choice as the strategy $\mathcal{G}$, majority voting with random resolution of ties as $\mathcal{S}$, and train–and–test at ratio 1:1 as the method of test performing. Quality of classification is measured by *total accuracy* being the ratio of the number of correctly classified cases to the number of recognized cases, and *total coverage*, i.e, the ratio of the number of recognized test cases to the number of test cases. Results are given in Table 1. *nil* denotes results without granulation to which granular results are compared.

The procedure has been as follows.

1. the data table (U, A) has been input;
2. classification rules have been found on the training subtable of 50 percent of objects by means of each of the three algorithms;
3. classification of dataset objects in the test subtable of remaining 50 percent of objects has been found for each of the three classifications found at point 2;

4. given the granule radius, granules of that radius have been found on the training subtable;
5. a granular covering of the training subtable has been chosen;
6. the corresponding granular decision system has been determined;
7. granular classifiers have been induced from the granular system in point 6 by means of each of algorithms in point 2;
8. classifications of objects in the test subtable have been found by means of each of classifiers in point 7;
9. classifications from points 3,8 have been compared with respect to adopted global measures of quality: total accuracy and total covering.

Table 1. Primary tumor dataset:r=granule radius,tst=test sample size,trn=training sample size,rulex=number of rules with exhaustive algorithm, rullem=number of rules with LEM2, aex=total accuracy with exhaustive algorithm,cex=total coverage with exhaustive algorithm,alem=total accuracy with LEM2, clem=total coverage with LEM2

r	*tst*	*trn*	*rulex*	*rullem*	*aex*	*cex*	*alem*	*clem*
nil	170	169	4186	43	0.253	0.976	0.5	0.259
0.0	170	1	0	0	0.0	0.0	0.0	0.0
0.0588235	170	1	0	0	0.0	0.0	0.0	0.0
0.117647	170	1	0	0	0.0	0.0	0.0	0.0
0.176471	170	1	0	0	0.0	0.0	0.0	0.0
0.235294	170	1	0	0	0.0	0.0	0.0	0.0
0.294118	170	1	0	0	0.0	0.0	0.0	0.08
0.352941	170	1	0	0	0.0	0.0	0.0	0.0
0.411765	170	2	0	1	0.0	0.0	1.0	0.188
0.470588	170	3	0	1	0.0	0.0	1.0	0.188
0.529412	170	5	0	1	0.0	0.0	1.0	0.188
0.588235	170	8	0	1	0.0	0.0	1.0	0.188
0.647059	170	12	11	1	0.547	0.376	0.0	0.0
0.705882	170	17	40	3	0.457	0.476	0.667	0.035
0.764706	170	33	108	4	0.468	0.553	0.769	0.076
0.823529	170	54	1026	11	0.434	0.759	0.586	0.171
0.882353	170	75	3640	17	0.308	0.859	0.579	0.224
0.941176	170	107	4428	24	0.295	0.976	0.466	0.341

Conclusions for Primary tumor. For exhaustive algorithm,accuracy is better with granular than original training set from the radius of 0.647059 on where reduction in size of training set is 92.9 percent and reduction in size of rule set is almost 100 percent (11 versus 4186). Coverage falls within error bound of 22.3 percent from the radius of 0.823529 on, where reduction in training st size is 68.2 percent and reduction in size of rule set is 75.5 percent; it becomes the same as in non–granular case at $r = .941$ with reduction in object size of 36.7 percent.

LEM2 exceeds accuracy of classifier trained on original training table with accuracy of granular classifier from the radius of 0.705882 on where reduction in training set size is 89.95 percent and reduction in rule set size is 93 percent. Coverage for granular classifier is better or within error of 13.5 percent from the radius of 0.882353 where reduction in size of the training set is 55.6 percent and reduction in size of rule set is 60.5 percent.

Thus, granular approach provides results on par with those obtained in non–granular case.

5 A Granular Approach to Missing Values

An information/decision system is *incomplete* in case some values of conditional attributes from A are not known. Analysis of systems with missing values requires a decision on how to treat missing values; Grzymala–Busse in his work [2], analyzes nine such methods, among them, *4. assigning all possible values to the missing location*, *9. treating the unknown value as a new valid value*, etc. etc. Results in [2] indicate that methods *4,9* perform very well among all nine methods. In this work we consider and adopt two methods, i.e.*4, 9*. Analysis of this problem has been given also in Kryszkiewicz [4] and Kryszkiewicz–Rybinski [5].

We will use the symbol $*$ commonly used for denoting the missing value; we will use two methods *4, 9* for treating $*$, i.e, either $*$ is a *don't care* symbol meaning that any value of the respective attribute can be substituted for $*$, thus $* = v$ for each value v of the attribute, or $*$ is a new value on its own, i.e., if $* = v$ then v can be only $*$.

Table 2. Strategy A. CV–5; Hepatitis; exhaustive algorithm. r=radius,macc=mean accuracy, mcov=mean coverage, mrul=mean number of rules, mtrn=mean training granular sample size

r	*macc*	*mcov*	*mrul*	*mtrn*
0.0	0.0	0.0	0.0	1.0
0.0526316	0.0	0.0	0.0	1.0
0.105263	0.0	0.0	0.0	1.0
0.157895	0.0	0.0	0.0	1.0
0.210526	0.0	0.0	0.0	1.0
0.263158	0.0	0.0	0.0	1.4
0.315789	0.0	0.0	0.0	2.0
0.368421	0.0	0.0	0.0	2.4
0.421053	0.0	0.0	0.0	3.8
0.473684	0.2012	0.3548	6.4	3.4
0.526316	0.5934	1.0	29.6	7.4
0.578947	0.4992	0.7872	33.8	7.6
0.631579	0.5694	0.9872	176.2	20.0
0.684211	0.5852	0.9936	167.6	17.8
0.736842	0.6102	0.9936	263.0	22.8
0.789474	0.6130	1.0	911.0	49.4
0.842105	0.6258	1.0	989.6	46.8
0.894737	0.6386	1.0	1899.0	77.0
0.947368	0.6774	1.0	2836.2	105.8
1.0	0.6710	1.0	3286.4	123.4

Table 3. Strategy B. CV–5; Hepatitis; exhaustive algorithm. r=radius,macc=mean accuracy, mcov=mean coverage, mrul=mean number of rules, mtrn=mean training granular sample size

r	$macc$	$mcov$	$mrul$	$mtrn$
0.0	0.0	0.0	0.0	1.0
0.0526316	0.0	0.0	0.0	1.0
0.105263	0.0	0.0	0.0	1.0
0.157895	0.0	0.0	0.0	1.0
0.210526	0.0	0.0	0.0	1.2
0.263158	0.0	0.0	0.0	1.2
0.315789	0.0	0.0	0.0	1.6
0.368421	0.1104	0.1870	1.0	2.6
0.421053	0.0904	0.2000	1.6	3.4
0.473684	0.3938	0.5806	7.2	4.4
0.526316	0.4234	0.7936	26.2	7.6
0.578947	0.6302	0.9936	59.4	10.8
0.631579	0.6708	1.0	126.4	15.4
0.684211	0.6038	0.9742	253.4	24.4
0.736842	0.6292	0.9936	367.6	35.2
0.789474	0.6166	0.9936	947.0	52.2
0.842105	0.6324	1.0	1417.2	71.8
0.894737	0.6386	1.0	1797.0	79.6
0.947368	0.6450	1.0	3081.8	113.4
1.0	0.6646	1.0	3354.2	123.4

Our procedure for treating missing values is based on the granular structure $(U^G_{r,\mu}, \mathcal{G}, \mathcal{S}, \{a* : a \in A\})$; the strategy $\mathcal{S}$ is the majority voting, i.e., for each attribute a, the value $a^*(g)$ is the most frequent of values in $\{a(u) : u \in g\}$. The strategy $\mathcal{G}$ consists in random selection of granules for a covering.

For an object u with the value of $*$ at an attribute a, and a granule $g = g(v, r) \in U^G_{r,\mu}$, the question whether u is included in g is resolved according to the adopted strategy of treating $*$: in case $* = don't\ care$, the value of $*$ is regarded as identical with any value of a hence $|IND(u, v)|$ is automatically increased by 1, which increases the granule; in case $* = *$, the granule size is decreased. Assuming that $*$ is sparse in data, majority voting on g would produce values of a^* distinct from $*$ in most cases; nevertheless the value of $*$ may appear in new objects g^*, and then in the process of classification, such value is repaired by means of the granule closest to g^* with respect to the rough inclusion μ_L, in accordance with the chosen method for treating $*$.

In plain words, objects with missing values are in a sense absorbed by close to them granules and missing values are replaced with most frequent values in objects collected in the granule; in this way the method *3* or *4* in [2] is combined with the idea of a frequent value, in a novel way.

We have thus four possible strategies:

- Strategy A: in building granules $*$=*don't care*, in repairing values of $*$, $*$=*don't care*;

- Strategy B: in building granules $*$=*don't care*, in repairing values of $*$, $* = *$;
- Strategy C: in building granules $* = *$, in repairing values of $*$, $*$=*don't care*;
- Strategy D: in building granules $* = *$, in repairing values of $*$, $* = *$.

5.1 Results of Test with Real Data Set Hepatitis with Missing Values

We record here results of tests with Hepatitis data set [21] with 155 objects, 20 attributes and 167 missing values. We apply the exhaustive algorithm of RSES system [19] and 5–fold cross–validation (CV–5). Below we give averaged results for strategies A, B, C, and D. As before, radius *nil* indicates non–granulated case.

Now, we record in Tables 2–5 the results of classification for Hepatitis with exhaustive algorithm and CV–5 cross–validation for strategies A, B, C, D, respectively.

For comparison, we include results of tests with Hepatitis recorded in [1]; the method was modified LERS algorithm with additional parameters like strength and specificity of a rule and the approach 9. gave error rate of 0.1935 i.e. accuracy 0.8065. Best result given by strategy C based on the same treatment of $*$ is accuracy 0.6838. Naive LERS algorithm [1] gave for this data set and method 9 error of 0.3484 i.e. accuracy of 0.6516. Interestingly, granular method gives better than [1] results for Breast cancer data set, as reported in [15], these Proceedings.

Table 4. Strategy C. CV–5; Hepatitis; exhaustive algorithm. r=radius,macc=mean accuracy, mcov=mean coverage, mrul=mean number of rules, mtrn=mean training granular sample size

r	*macc*	*mcov*	*mrul*	*mtrn*
0.0	0.0	0.0	0.0	1.0
0.0526316	0.0	0.0	0.0	1.0
0.105263	0.0	0.0	0.0	1.2
0.157895	0.0	0.0	0.0	1.2
0.210526	0.0	0.0	0.0	1.8
0.263158	0.0	0.0	0.0	2.0
0.315789	0.2560	0.3936	2.4	4.0
0.368421	0.4486	0.6838	7.4	5.6
0.421053	0.4766	0.7870	19.2	7.8
0.473684	0.5806	1.0	58.4	10.6
0.526316	0.6580	1.0	136.6	17.4
0.578947	0.64902	0.9936	332.4	32.0
0.631579	0.6568	0.9936	991.6	47.4
0.684211	0.6646	1.0	1751.6	70.2
0.736842	0.6902	1.0	2648.8	93.2
0.789474	0.6322	1.0	3208.8	112.6
0.842105	0.6776	1.0	3297.8	120.2
0.894737	0.6710	1.0	3297.4	123.4
0.947368	0.6838	1.0	3305.4	124.0
1.0	0.6774	1.0	3327.2	124.0

Table 5. Strategy D. CV–5; Hepatitis; exhaustive algorithm. r=radius,macc=mean accuracy, mcov=mean coverage, mrul=mean number of rules, mtrn=mean training granular sample size

r	$macc$	$mcov$	$mrul$	$mtrn$
0.0	0.0	0.0	0.0	1.0
0.0526316	0.0	0.0	0.0	1.0
0.105263	0.0	0.0	0.0	1.0
0.157895	0.0	0.0	0.0	1.4
0.210526	0.0	0.0	0.0	1.6
0.263158	0.0	0.0	0.0	2.6
0.315789	0.3886	0.5162	6.0	3.8
0.368421	0.5730	0.9032	16.6	4.8
0.421053	0.6328	0.9418	23.8	6.8
0.473684	0.5740	0.9740	60.6	10.6
0.526316	0.6170	0.9936	120.6	16.8
0.578947	0.6888	0.9936	354.0	30.6
0.631579	0.6388	1.0	922.0	47.4
0.684211	0.6646	1.0	1828.6	70.8
0.736842	0.6450	1.0	2648.2	93.4
0.789474	0.6516	1.0	3182.0	112.4
0.842105	0.6710	1.0	3299.2	120.4
0.894737	0.6710	1.0	3333.8	123.4
0.947368	0.6646	1.0	3327.2	124.0
1.0	0.6710	1.0	3338.6	124.0

Conclusions for Hepatitis data set. Results for particular strategies compared radius by radius show that the ranking of strategies is $C > D > B > A$ with the average number of ranks respectively 1.3, 1.8., 3.1, 3.8; thus, the strategy C is most effective with D giving slightly worse results. Results by our granular approach are midway between results for naive and new LERS in [1] showing the potential of the method as well as the need for further development.

6 Conclusion

The results of tests reported in this work bear out the hypothesis that granulated data sets preserve information allowing for satisfactory classification. Also the novel approach to the problem of data with missing values has proved to be very effective. Further studies will lead to novel algorithms for rule induction based on granules of knowledge.

Acknowledgement

The author wishes to express his thanks to Professors A. Skowron and J. W. Grzymala–Busse who provided in the system RSES rule inducing algorithms used in this work.

References

1. Grzymala–Busse, J.W., Hu, M.: A comparison of several approaches to missing attribute values in Data Mining. Lecture Notes in Artificial intelligence, vol. 2005, pp. 378–385. Springer, Berlin (2000)
2. Grzymala–Busse, J.W.: Data with missing attribute values: Generalization of rule indiscernibility relation and rule induction. In: Transactions on Rough Sets I, pp. 78–95. Springer, Berlin (2004)
3. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
4. Kryszkiewicz, M.: Rules in incomplete information systems. Information Sciences 113, 271–292 (1999)
5. Kryszkiewicz, M., Rybiński, H.: Data mining in incomplete information systems from rough set perspective. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.) Rough Set Methods and Applications, pp. 568–580. Physica Verlag, Heidelberg (2000)
6. Leśniewski, S.: On the foundations of set theory. Topoi 2, 7–52 (1982)
7. Lin, T.Y.: Granular computing: Examples, intuitions, and modeling, in: [17], pp. 40–44
8. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)
9. Pawlak, Z.: Rough sets. Int. J. Computer and Information Sci. 11, 341–356 (1982)
10. Polkowski, L.: Rough Sets. Mathematical Foundations. Physica Verlag, Heidelberg (2002)
11. Polkowski, L.: Toward rough set foundations. Mereological approach (a plenary lecture). In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 8–25. Springer, Berlin (2004)
12. Polkowski, L.: Rough–fuzzy–neurocomputing based on rough mereological calculus of granules. Intern. J. Hybrid Intell. Systems 2, 91–108 (2005)
13. Polkowski, L.: Formal granular calculi based on rough inclusions (a feature talk). In: [17], pp. 57–62
14. Polkowski, L.: A model of granular computing with applications (a feature talk). In: [18], 9–16.
15. Polkowski, L., Artiemjew, P.: On granular rough computing with missing values. In: these Proceedings
16. Polkowski, L., Skowron, A.: Rough mereology: a new paradigm for approximate reasoning. International Journal of Approximate Reasoning 15(4), 333–365 (1997)
17. In: Proceedings of IEEE 2005 Conference on Granular Computing,GrC05, Beijing, China, July 2005, IEEE Press, New York (2005)
18. In: Proceedings of IEEE 2006 Conference on Granular Computing, GrC06, Atlanta, USA, May 2006, IEEE Press, New York (2006)
19. A. Skowron et al., RSES: A system for data analysis; available at http: logic.mimuw.edu.pl/ rses/
20. Skowron, A., Stepaniuk, J.: Information granules and rough–neural computing. In: Pal, S.K., Polkowski, L., Skowron, A. (eds.) Rough Neural Computing, pp. 43–84. Springer, Berlin (2004)
21. http://www.ics.uci.edu./~mlearn/databases/
22. Yao, Y.Y.: Perspectives of granular computing. In: [17], pp. 85–90
23. Zadeh, L.A.: Fuzzy sets and information granularity. In: Gupta, M., Ragade, R., Yager, R. (eds.): Advances in Fuzzy Set Theory and Applications. North–Holland, Amsterdam, pp. 3–18 (1979)

MIRAI: Multi-hierarchical, FS-Tree Based Music Information Retrieval System

Zbigniew W. Raś[1,2], Xin Zhang[1], and Rory Lewis[1]

[1] University of North Carolina, Dept. of Comp. Science, Charlotte, N.C. 28223, USA
[2] Polish-Japanese Institute of Information Technology, 02-008 Warsaw, Poland

Abstract. With the fast booming of online music repositories, there is a need for content-based automatic indexing which will help users to find their favorite music objects in real time. Recently, numerous successful approaches on musical data feature extraction and selection have been proposed for instrument recognition in monophonic sounds. Unfortunately, none of these methods can be successfully applied to polyphonic sounds. Identification of music instruments in polyphonic sounds is still difficult and challenging, especially when harmonic partials are overlapping with each other. This has stimulated the research on music sound separation and new features development for content-based automatic music information retrieval. Our goal is to build a cooperative query answering system (QAS), for a musical database, retrieving from it all objects satisfying queries like "find all musical pieces in pentatonic scale with a viola and piano where viola is playing for minimum 20 seconds and piano for minimum 10 seconds". We use the database of musical sounds, containing almost 4000 sounds taken from the MUMs (McGill University Master Samples), as a vehicle to construct several classifiers for automatic instrument recognition. Classifiers showing the best performance are adopted for automatic indexing of musical pieces by instruments. Our musical database has an FS-tree (Frame Segment Tree) structure representation. The cooperativeness of QAS is driven by several hierarchical structures used for classifying musical instruments.

1 Introduction

Broader research on automatic musical instrument sound classification goes back to last few years. So far, there is no standard parameterization used as a classification basis. The sound descriptors used are based on various methods of analysis of time and spectrum domain, with Fourier Transform amplitude spectra being most common. Also, wavelet analysis gains increasing interest for sound and especially for musical sound analysis and representation [21], [9]. Diversity of sound timbres is also used to facilitate data visualization via sonification, in order to make complex data easier to perceive [1].

Many parameterization and recognition methods, including pitch extraction techniques, applied in musical research come from speech and speaker recognition domain [5], [22]. Sound parameters applied in research performed in musical instrument classification include cepstral coefficients, constant-Q coefficients,

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 80–89, 2007.

spectral centroid, autocorrelation coefficients, and moments of the time wave [3], wavelet analysis [23], [13], root mean square (RMS), amplitude envelope and multidimensional scaling analysis trajectories [12], and various spectral and temporal features [14], [17], [23]. The sound sets used differ from experiment to experiment, with McGill University Master Samples (MUMS) CDs being most common [19], yet not always used [3], making comparison of results more difficult. Some experiments operate on a very limited set of data, like 4 instruments, or singular samples for each instrument. Even if the investigations are performed on MUMS data, every researcher selects different group of instruments, number of classes, and testing method is also different. Therefore, data sets used in experiments and the obtained results are not comparable. Additionally, each researcher follows different parameterization technique(s), which makes comparison yet more difficult. Audio features in our system [26], [15] are first categorized as MPEG7 descriptors and other/non-MPEG7 descriptors in the acoustical perspective of view, where both spectrum features and temporal features are included. We have built a derivative database of those features with single valued data for KD-based classification. The spectrum features have two different frequency domains: Hz frequency and Mel frequency. Frame size was carefully designed to be 120ms, so that the 0th octave G (the lowest pitch in our audio database) can be detected. The hop size is 40ms with a overlapping of 80ms. A hamming window was applied to all STFTs (Short Time Fourier Transforms) to avoid jittering in the spectrum. By the results from the experiments, it was shown that the non-MPEG features significantly improve the performance of the classifiers [28].

The classifiers, applied in research on musical instrument sound classification, represent practically all known methods. The most popular classifier is k-Nearest Neighbor (k-NN), see for example [12]. This classifier is relatively easy to implement and quite successful. Other reported results include Bayes decision rules, Gaussian mixture model [3], artificial neural networks [13], decision trees and rough set based algorithms [24], discriminant analysis [17] hidden Markov Models (HMM), support vector machines (SVM) and other. The obtained results vary depending on the size of the data set, with accuracy reaching even 100% for 4 classes. However, the results for more than 10 instruments, explored in full musical scale range, generally are below 80%. Extensive review of parameterization and classification methods applied in research on this topic, with obtained results, is given in [10]. The classifiers investigated in our project include k-NN, Bayesian Networks, and Decision Tree J-48. We also consider use of neural networks, especially time-delayed neural networks (TDNN), since they perform well in speech recognition applications [18].

Musical instrument sounds can be classified in various ways, depending on the instrument or articulation classification. In [25], we review a number of possible generalizations of musical instruments sounds classification which can be used to construct different hierarchical decision attributes. Each decision attribute leads to a new classifier and the same to a different system for automatic indexing of music by instrument sounds and their generalizations. Values of any decision attribute and their generalizations can be seen as atomic queries of a query language built for retrieving musical objects from musical database. When query fails, the cooperative strategy tries to find its lowest generalization which does not fail, taking into consideration all available hierarchical attributes. Paper [25] evaluates two hierarchical attributes (Hornbostel-Sachs classification and classification by articulation) upon the same dataset which contains 2628 distinct musical samples of 102 instruments. By cross checking the resulting schemes for both attributes, it was observed that the timbre estimation of instruments had higher accuracy than that of instruments from other families by the classification by articulation. Also, among the musical objects played by different articulations, the sounds played by lip-vibration tended to be less correctly recognized by Hornbostel-Sachs classification. This justifies the construction of atomic queries from values of more than one decision attribute.

2 Sound Data

This paper deals with recordings where for each channel there is only access to one-dimensional data, i.e. to single sample representing amplitude of the sound. Any basic information like pitch (or pitches, if there are more sounds), timbre, beginning and end of the sound must be extracted via digital signal processing. The audio database consists of stereo musical pieces from the MUMS samples. These audio data files are treated as mono-channel, where only left channel was taken into consideration, since successful methods for the left channel will also be successfully applied to the right channel. In the view of classification, these audio data can be categorized into two different types: one is monophonic sound note to generate training feature set; the other is polyphonic sound sequence for testing.

Our research is driven by the desire to identify the individual instrument types or instrument family categories of the predominant instruments in a music object. Timbre is a quality of sound that distinguishes one music instrument from another, while there are a wide variety of instrument families and individual categories. It is rather subjective quality, defined by ANSI as the attribute of auditory sensation, in terms of which a listener can judge that two sounds, similarly presented and having the same loudness and pitch, are different. Such definition is subjective and not of much use for automatic sound timbre classification. Therefore, musical sounds must be very carefully parameterized to allow automatic timbre recognition. The real use of timbre-based grouping of music is discussed in [2]. Evolution of sound features in time is essential for humans, therefore it should be reflected in sound description as well. The discovered temporal patterns may better express sound features than static features, especially that classic features can be very similar for sounds representing the same family or pitch, whereas changeability of features with pitch for the same instrument makes sounds of one instrument dissimilar.

Based on recent research performed in MIR area, MPEG proposed an MPEG-7 standard, in which it described a set of low-level sound temporal and spectral features. The low-level descriptors in MPEG-7 are intended to describe the time-variant information within an entire audio segment, where most of them are, like other STFT related acoustic features, in a form of either vector or matrix of large size, where an audio segment was divided into a set of frames and each row represents a power spectrum in the frequency domain within each analysis window. Therefore, these features are not suitable for traditional classifiers, which require single-value cell of input datasets. Researchers have been explored different statistical summations in a form of single value to describe signatures of music instruments within vectors or matrices in those features, such as Tristimulus parameters [20] or Brightness [6]. However, current features fail to sufficiently describe the audio signatures which vary in time within a whole sound segment, esp. where multiple audio signatures are overlapping with each other. It was widely observed that a sound segment of a note, which is played by a music instrument, has at least three states: onset (transient), quasi-steady state and offset (transient). Vibration pattern in a transient state is known to significantly differ from the one in a quasi-steady state. Consequently, the harmonic features in the transient states behavior significantly different from those in the quasi-steady state. Also, it has been observed that a human needs to know the beginning of the music sound in order to discern the type of an instrument. Identifying the boundary of the transient state enables accurate timbre recognition.

3 Feature Database Construction

Our research involves the construction of two main databases, one is a monophonic sound feature database, which is used for classifiers construction; the other is a polyphonic audio database, which is used for testing. The latter will have FS-tree structure driven by automatic indexing of audio files by music instruments and their classes. The monophonic sound feature database contains over 1022 attributes, where 1018 of them were computed from the digital monophonic sound files and four decision hierarchical attributes were manually labelled. There are many ways to categorize the audio features. In our research, computational audio features are first categorized as MPEG7 based descriptors and other/non-MPEG7 descriptors in the acoustical perspective of view, where both spectrum features and temporal features are included. Then, a derivative database of those features with single valued data features, for the purpose of learning classifiers, is constructed. The manually labelled decision attributes will be discussed in latter section. Spectrum features have different frequency domains: Hz frequency and Mel frequency. Frame size is chosen as 0.12 second, so that the 0th octave G (the lowest pitch in our audio database) can be detected, which is also within the range of estimates for temporal acuity of human ear. The hop size is 0.04 second with a overlapping of 0.08 second. Since the sampling frequency of all the music objects is 44,100Hz, there are 5292 sample data per frame in the waveform.

The list of MPEG7 features includes: Harmonic Upper Limit, Harmonic Ratio, Basis Functions, Log Attack Time, Temporal Centroid, Spectral Centroid, Spectrum Centroid/Spread I, Harmonic Parameters, Flatness. The list of extended MPEG7 features and other features includes: Tristimulus Parameters, Spectrum Centriod/Spread II, Flux, Roll Off, Zero Crossing, MFCC, Spectrum Centroid/Spread I, Harmonic Parameters, Flatness, Durations. Intermediate features include Harmonic Upper Limit and Projection.

4 Sound Separation

Our system consists of five modules: a quasi-steady state detector, a $STFT$ converter with hamming window, a pre-dominant fundamental frequency estimator, a sequential pattern matching engine (it will be replaced by a classifier) with connection to a feature database, a FFT subtraction device [27].

The quasi-steady state detector computes overall fundamental frequency in each frame by a cross-correlation function, and outputs the beginning and end positions of the quasi-steady state of the input sound.

The $STFT$ converter divides a digital audio object into a sequence of frames, applies $STFT$ transform to the mixed sample data of integers from time domain to frequency domain with a hamming window, and outputs $NFFT$ discrete points.

The pre-dominant fundamental frequency estimator identifies all the possible harmonic peaks, computes the likelihood value for each candidate peak, elects

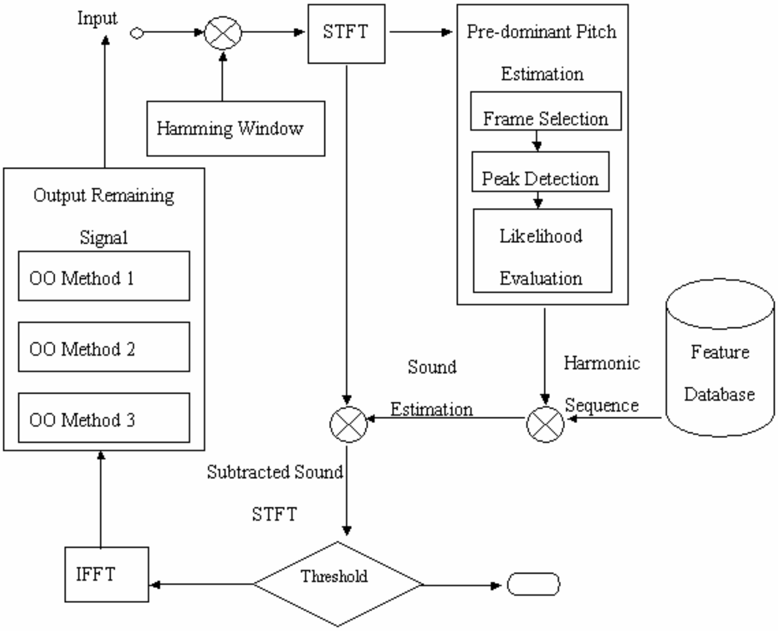


Fig. 1. Sound Separation System

the frequency with the maximum likelihood value as the fundamental frequency, and stores its normalized correspondence harmonic sequence.

The sequential-pattern matching engine computes the distance of each pair wise sequence of first N harmonic peaks, where N is set empirically, then outputs the sound with the minimum distance value for each frame, and finally estimates the sound object by the most frequent sound object among all the frames.

The FFT subtraction device subtracts the detected sound source from the spectrum, computes the imaginary and real part of the FFT point by the power and phase information, performs $IFFT$ for each frame, and outputs resultant remaining signals into a new audio data file.

5 Multi-way Hierarchic Classification

Classification of musical instrument sounds can be performed in various ways [11]. Paper [25] reviews several hierarchical classifications of musical instrument sounds but concentrates only on two of them: Hornbostel-Sachs classification of musical instruments and classification of musical instruments by articulation with 15 different articulation methods (seen as attribute values): blown, bowed, bowed vibrato, concussive, hammered, lip-vibrated, martele, muted, muted vibrato, percussive, picked, pizzicato, rubbed, scraped and shaken. Each hierarchical classification represents a unique decision attribute which leads us to a discovery of a new classifier and the same to a different system for automatic indexing of music by instruments and their certain generalizations.

The goal of each classification is to find descriptions of musical instruments or their classes (values of attribute d) in terms of values of attributes from A. Each classification results in a classifier which can be evaluated using standard methods like bootstrap or cross-validation.

In [25] authors concentrate on classifiers built by rule-based methods (for instance: *LERS, RSES, PNC2*) and next on classifiers built by tree-based methods (for instance: *See5, J48 Tree, Assistant, CART, Orange*).

Let us assume that $S = (X, A \cup \{d\}, V)$ is a decision system, where d is a hierarchical attribute. We also assume that $d_{[i_1,\ldots,i_k]}$ (where $1 \leq i_j \leq m_j$, $j = 1, 2\ldots, k$) is a child of $d_{[i_1,\ldots,i_{k-1}]}$ for any $1 \leq i_k \leq m_k$. Clearly, attribute d has $\Sigma\{m_1 \cdot m_2 \cdot \ldots \cdot m_j : 1 \leq j \leq k\}$ values, where $m_1 \cdot m_2 \cdot \ldots \cdot m_j$ shows the upper bound for the number of values at the level j of d. By $p([i_1, \ldots, i_k])$ we denote a path (d, $d_{[i_1]}$, $d_{[i_1,i_2]}$, $d_{[i_1,i_2,i_3]}$,..., $d_{[i_1,\ldots,i_{k-1}]}$, $d_{[i_1,\ldots,i_k]}$) leading from the root of the hierarchical attribute d to its descendant $d_{[i_1,\ldots,i_k]}$.

Let us assume that R_j is a set of classification rules extracted from S, representing a part of a rule-based classifier $R = \bigcup\{R_j : 1 \leq j \leq k\}$, and describing all values of d at level j. The quality of a classifier at level j of attribute d can be checked by calculating $Q(R_j) = \frac{\sum\{sup(r) \cdot conf(r) : r \in R_j\}}{\sum\{sup(r : r \in R_j\}}$, where $sup(r)$ is the support of the rule r in S and $conf(r)$ is its confidence. Then, the quality of the rule-based classifier R can be checked by calculating $Q(\bigcup\{R_j : 1 \leq j \leq k\}) = \frac{\sum\{Q(R_j) : 1 \leq j \leq k\}}{k}$.

The quality of a tree-based classifier can be given by calculating its quality for every node of a hierarchical decision attribute d. Let us take a node $d_{[i_1,...,i_k]}$ and the path $p([i_1,...,i_k])$ leading to that node from the root of d. There is a set of classification rules $R_{[i_1,...,i_m]}$, uniquely defined by the tree-based classifier, assigned to a node $d_{[i_1,...,i_m]}$ of a path $p([i_1,...,i_k])$, for every $1 \leq m \leq k$. Now, we define $Q(R_{[i_1,...,i_m]})$ as $\frac{\sum\{sup(r)\cdot conf(r):r\in R_{[i_1,...,i_m]}\}}{\sum\{sup(r):r\in R_{[i_1,...,i_m]}\}}$. Then, the quality of a tree-based classifier for a node $d_{[i_1,...,i_m]}$ of the decision attribute d can be checked by calculating $Q(d_{[i_1,...,i_m]}) = \prod\{Q(R_{[i_1,...,i_j]}) : 1 \leq j \leq m\}$. In our experiments, presented in Section 4 of this paper, we use *J48 Tree* as the tool to build tree-based classifiers. Also, their performance on level m of the attribute d is checked by calculating $Q(d_{[i_1,...,i_m]})$ for every node $d_{[i_1,...,i_m]}$ at the level m. Finally, the performance of both classifiers is checked by calculating $Q(\bigcup\{R_j : 1 \leq j \leq k\})$ (the first method we proposed).

Learning values of a decision attribute at different generalization levels is extremely important not only for designing and developing an automatic indexing system of possibly highest confidence but also for handling failing queries. Values of a decision attribute and their generalizations are used to construct atomic queries of a query language built for retrieving musical objects from *MIR* Database (see http://www.mir.uncc.edu). When query fails, the cooperative strategy [7], [8] may try to find its lowest generalization which does not fail. Clearly, by having a variety of different hierarchical structures available for d we have better chance not only to succeed but succeed with a possibly smallest generalization of an instrument class.

6 Flexible Query Answering System

Now, we discuss how a Flexible Query Answering System (see Figure 1) associated with a database D of music files works for a sample query which consists of two parts: a digital musical file F and an instrument T. The query should be read as: *Find all musical pieces, in the database D, which are played by the same instruments as the instruments used in F.* Also the duration time of all these instruments has to be the same (threshold value can be provided).

The digital musical file is divided into segments of equal length. Automatic indexing system operates on each segment piece and outputs a vector of features describing its content. Then a classifier estimates what instruments are present in each segment and what is their time duration and then searches the FS-tree to identify the musical pieces in database D satisfying the query. If query fails, then an instrument used in F which has the most similar timbre to the instrument T is identified and it is replaced by T assuming that its time duration is the same as the time duration of the replaced instrument. Finally, the closest musical file to the file requested by user is returned as the result of the query. Alternatively, the classifier of a higher level in the instrument family tree is assigned for timbre classification on its own level, and repeats the steps until a desire result is achieved or the root of the instrument family tree is reached. This

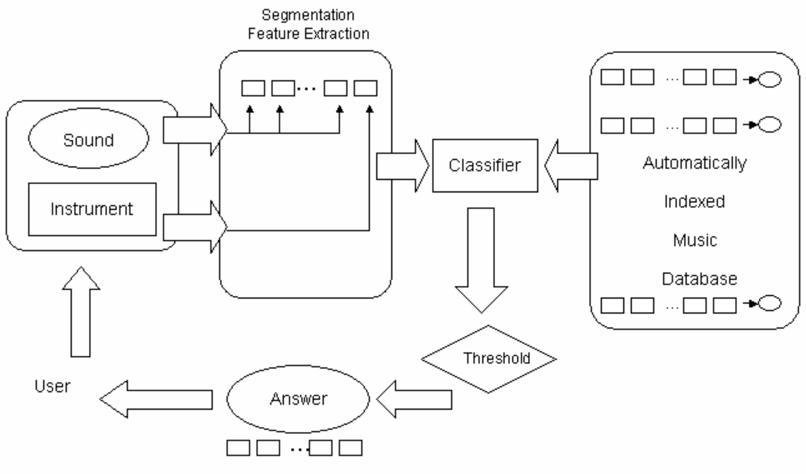


Fig. 2. Flexible Query Answering System based on MIR

approach especially benefits non-musician users who have limited information on music instrument classification schema.

7 Conclusion and Acknowledgement

The ultimate goal of this research is to build a cooperative system for automatic indexing of music by instruments or classes of instruments, use this system to build FS-tree type music database for storing automatically indexed musical files, and finally design and implement a Cooperative Query Answering System to handle user requests submitted to music database.

This research was supported by the National Science Foundation under grant IIS-0414815.

References

1. Ben-Tal, O., Berger, J., Cook, B., Daniels, M., Scavone, G., Cook, P.: SONART: The Sonification Application Research Toolbox. In: Proceedings of the 2002 International Conference on Auditory Display, Kyoto, Japan, July 2002 (2002)
2. Bregman, A.S.: Auditory scene analysis, the perceptual organization of sound. MIT Press, Cambridge (1990)
3. Brown, J.C., Houix, O., McAdams, S.: Feature dependence in the automatic identification of musical woodwind instruments. J. Acoust. Soc. of America 109, 1064–1072 (2001)
4. Cardoso, J.F., Comon, P.: Independent Component Analysis, a Survey of Some Algebraic methods. In: Proc. ISCAS Conference, Atlanta, May 1996, vol. 2, pp. 93–96 (1996)

5. Flanagan, J.L.: Speech Analysis, Synthesis and Perception. Springer, New York (1972)
6. Fujinaga, I., McMillan, K.: Real time Recognition of Orchestral Instruments. In: International Computer Music Conference, pp. 141–143 (2000)
7. Gaasterland, T.: Cooperative answering through controlled query relaxation. IEEE Expert 12(5), 48–59 (1997)
8. Godfrey, P.: Minimization in cooperative response to failing database queries. International Journal of Cooperative Information Systems 6(2), 95–149 (1993)
9. Goodwin, M.M.: Adaptive Signal Models: Theory, Algorithms, and Audio Applications, Ph.D. dissertation, University of California, Berkeley (1997)
10. Herrera, P., Amatriain, X., Batlle, E., Serra, X.: Towards instrument segmentation for music content description: a critical review of instrument classification techniques. In: Proc. of International Symposium on Music Information Retrieval (ISMIR 2000), Plymouth, MA (2000)
11. Hornbostel, E.M.V., Sachs, C.: Systematik der Musikinstrumente. Ein Versuch. Zeitschrift fur Ethnologie 46(4-5), 553–590 (1914), available at http://www.uni-bamberg.de/ppp/ethnomusikologie/HS-Systematik/HS-Systematik
12. Kaminskyj, I.: Multi-feature Musical Instrument Classifier, MikroPolyphonie 6, 2000, online journal at http://farben.latrobe.edu.au/
13. Kostek, B., Czyzewski, A.: Representing Musical Instrument Sounds for Their Automatic Classification. J. Audio Eng. Soc. 49(9), 768–785 (2001)
14. Kostek, B., Wieczorkowska, A.: Parametric Representation of Musical Sounds. Archive of Acoustics 22(1), 3–26 (1997)
15. Lewis, R., Zhang, X., Ras, Z.W.: Blind Signal Separation of Similar Pitches and Instruments in a Noisy Polyphonic Domain. In: Esposito, F., Raś, Z.W., Malerba, D., Semeraro, G. (eds.) ISMIS 2006. LNCS (LNAI), vol. 4203, pp. 228–237. Springer, Heidelberg (2006)
16. Manjunath, B.S., Salembier, P., Sikora, T. (eds.): Introduction to MPEG-7. Multimedia Content Description Interface. J. Wiley and Sons, New York (2002)
17. Martin, K.D., Kim, Y.E.: Musical instrument identification: a pattern-recognition approach. In: Proceedings of 136th Meeting of the Acoustical Society of America, Norfolk, VA, October 1998 (1998)
18. Meier, U., Stiefelhagen, R., Yang, J., Waibel, A.: Towards Unrestricted Lip Reading. International Journal of Pattern Recognition and Artificial Intelligence 14(5), 571–586 (2000)
19. Opolko, F., Wapnick, J.: MUMS - McGill University Master Samples, CD's (1987)
20. Pollard, H.F., Jansson, E.V.: A Tristimulus Method for the spectificaiton of Musical Timbre. Acustica (51), 162–171 (1982)
21. Popovic, I., Coifman, R., Berger, J.: Aspects of Pitch-Tracking and Timbre Separation: Feature Detection in Digital Audio Using Adapted Local Trigonometric Bases and Wavelet Packets Center for Studies in Music Technology, Yale University, Research Abstract, June 1995 (1995)
22. Rabiner, L., Schafer, R.: Digital Processing of Speech Signals. Prentice-Hall, Englewood Cliffs, New Jersey (1978)
23. Wieczorkowska, A.: Musical Sound Classification based on Wavelet Analysis. Fundamenta Informaticae Journal 47(1), 175–188 (2001)
24. Wieczorkowska, A.: The recognition efficiency of musical instrument sounds depending on parameterization and type of a classifier, PhD. thesis (in Polish), Technical University of Gdansk, Poland (1999)

25. Wieczorkowska, A., Raś, Z.W., Zhang, X., Lewis, R.: Multi-way Hierarchic Classification of Musical Instrument Sounds. In: Proceedings of the IEEE CS International Conference on Multimedia and Ubiquitous Engineering (MUE 2007) April 26-28, 2007, in Seoul, Korea (will appear)
26. Zhang, X., Raś, Z.W.: Differentiated Harmonic Feature Analysis on Music Information Retrieval For Instrument Recognition. In: Proceedings of IEEE International Conference on Granular Computing (IEEE GrC, 2006) May 10-12, 2006, Atlanta, Georgia, 578–581 (2006)
27. Zhang, X., Marasek, K., Raś, Z.W.: Maximum Likelihood Study for Sound Pattern Separation and Recognition. In: Proceedings of the IEEE CS International Conference on Multimedia and Ubiquitous Engineering (MUE, 2007) April 26-28, 2007, in Seoul, Korea (will appear)
28. Zhang, X., Raś, Z.W.: Analysis of Sound Features for Music Timbre Recognition. In: Proceedings of the IEEE CS International Conference on Multimedia and Ubiquitous Engineering (MUE 2007), April 26-28, 2007, in Seoul, Korea (will appear)

Medical Reasoning and Rough Sets

Shusaku Tsumoto

Department of Medical Informatics,
Faculty of Medicine, Shimane University
89-1 Enya-cho Izumo 693-8501 Japan
tsumoto@computer.org

Abstract. Pawlak showed that knowledge can be captured by data partition and proposed a rough set method where comparison between data partition gives knowledge about classification. Interestingly, thes approximations correspond to the focusing mechanism of differential medical diagnosis; upper approximation as selection of candidates and lower approximation as concluding a final diagnosis. This paper focuses on severl models of medical reasoning shows that core ideas of rough set theory can be observed in these diagnostic models.

1 Introduction

Medical reasoning always includes uncertainty[1], which is caused by the limitations of medical knowledge, available data and our recognition, compared with the complexities of human body. Thus, medical databases also have a certain degree of uncertainty: rules extracted from databases are also incomplete, which suggests that rule induction method should deal with uncertain rules.

According to this motivation, rule induction based on rough set theory have been applied to medical databases empirically[2,3], the results of which shows that rough-set-based methods are very useful to extract medical diagnostic rules.

This paper presents how medical diagnostic rules are modeled by the concepts of rough sets[4] in a more theoretical way. The key ideas are variable precision rough set model, which corresponds to a ordinal positive reasoning, and an upper approximation of a target concept, which corresponds to a focusing mechanism in medical reasoning. Acquired models show that the characteristics of medical reasoning reflect the concepts on approximation of rough sets, which explains why rough sets work well in medical domains. The paper is organized as follows: in Section 2, two important measures, accuracy and coverage are defined and a probabilistic rule is defined. Section 3 to 5 presents description of three types of medical reasoning: simple differential diagnosis, focusing mechanism and $m-$of$-n$ criteria, respectively. Section 6 concludes our paper.

2 Definition of Rules

2.1 Rough Sets

In the following sections, we use the following notations introduced by Grzymala-Busse and Skowron[5], which are based on rough set theory[4]. These notations

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 90–100, 2007.

Table 1. An Example of Dataset

No.	age	location	nature	prodrome	nausea	M1	class
1	50-59	occular	persistent	no	no	yes	m.c.h.
2	40-49	whole	persistent	no	no	yes	m.c.h.
3	40-49	lateral	throbbing	no	yes	no	migra
4	40-49	whole	throbbing	yes	yes	no	migra
5	40-49	whole	radiating	no	no	yes	m.c.h.
6	50-59	whole	persistent	no	yes	yes	psycho

DEFINITIONS. M1: tenderness of M1, m.c.h.: muscle contraction headache, migra: migraine, psycho: psychological pain.

are illustrated by a small dataset shown in Table 1, which includes symptoms exhibited by six patients who complained of headache.

Let U denote a nonempty, finite set called the universe and A denote a nonempty, finite set of attributes, i.e., $a : U \to V_a$ for $a \in A$, where V_a is called the domain of a, respectively.Then, a decision table is defined as an information system, $A = (U, A \cup \{d\})$. For example, Table 1 is an information system with $U = \{1, 2, 3, 4, 5, 6\}$ and $A = \{age, location, nature, prodrome, nausea, M1\}$ and $d = class$. For $location \in A$, $V_{location}$ is defined as $\{occular, lateral, whole\}$.

The atomic formulae over $B \subseteq A \cup \{d\}$ and V are expressions of the form $[a = v]$, called descriptors over B, where $a \in B$ and $v \in V_a$. The set $F(B, V)$ of formulas over B is the least set containing all atomic formulas over B and closed with respect to disjunction, conjunction and negation. For example, $[location = occular]$ is a descriptor of B.

For each $f \in F(B, V)$, f_A denote the meaning of f in A, i.e., the set of all objects in U with property f, defined inductively as follows.

1. If f is of the form $[a = v]$ then, $f_A = \{s \in U | a(s) = v\}$
2. $(f \wedge g)_A = f_A \cap g_A$; $(f \vee g)_A = f_A \vee g_A$; $(\neg f)_A = U - f_a$

For example, $f = [location = whole]$ and $f_A = \{2, 4, 5, 6\}$. As an example of a conjunctive formula, $g = [location = whole] \wedge [nausea = no]$ is a descriptor of U and f_A is equal to $g_{location,nausea} = \{2, 5\}$.

2.2 Classification Accuracy and Coverage

Definition of Accuracy and Coverage. By the use of the framework above, classification accuracy and coverage, or true positive rate is defined as follows.

Definition 1

Let R and D denote a formula in $F(B, V)$ and a set of objects which belong to a decision d. Classification accuracy and coverage(true positive rate) for $R \to d$ is defined as:

$$\alpha_R(D) = \frac{|R_A \cap D|}{|R_A|}(= P(D|R)), \ \textit{and}$$
$$\kappa_R(D) = \frac{|R_A \cap D|}{|D|}(= P(R|D)),$$

where $|S|$, $\alpha_R(D)$, $\kappa_R(D)$ and P(S) denote the cardinality of a set S, a classification accuracy of R as to classification of D and coverage (a true positive rate of R to D), and probability of S, respectively.

Figure 1 depicts the Venn diagram of relations between accuracy and coverage. Accuracy views the overlapped region $|R_A \cap D|$ from the meaning of a relation R. On the other hand, coverage views the overlapped region from the meaning of a concept D.

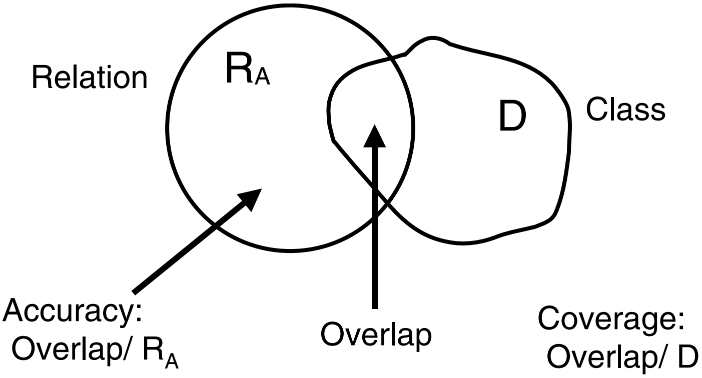


Fig. 1. Venn Diagram of Accuracy and Coverage

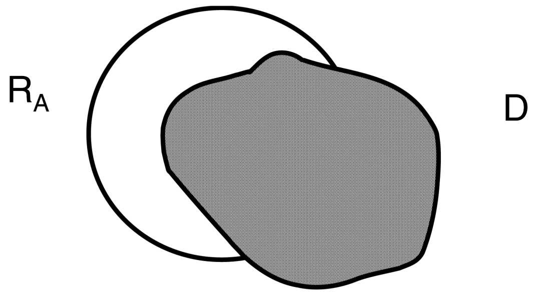
In the above example, when R and D are set to $[nau = yes]$ and $[class = migraine]$, $\alpha_R(D) = 2/3 = 0.67$ and $\kappa_R(D) = 2/2 = 1.0$.

It is notable that $\alpha_R(D)$ measures the degree of the sufficiency of a proposition, $R \rightarrow D$, and that $\kappa_R(D)$ measures the degree of its necessity. For example, if $\alpha_R(D)$ is equal to 1.0, then $R \rightarrow D$ is true. On the other hand, if $\kappa_R(D)$ is equal to 1.0, then $D \rightarrow R$ is true. Thus, if both measures are 1.0, then $R \leftrightarrow D$. Other characteristics of accuracy and coverage are shown in the appendix.

2.3 Probabilistic Rules

By the use of accuracy and coverage, a probabilistic rule is defined as:

$$R \overset{\alpha,\kappa}{\rightarrow} d \quad s.t. \ R = \wedge_j [a_j = v_k], \alpha_R(D) \geq \delta_\alpha \ \text{ and } \kappa_R(D) \geq \delta_\kappa,$$



$$\boxed{R \rightarrow D \quad s.t. \quad \alpha_R(D) > \delta_\alpha, \kappa_R(D) > \delta_\kappa}$$

Fig. 2. Venn Diagram for Probabilistic Rules

If the thresholds for accuracy and coverage are set to high values, the meaning of the conditional part of probabilistic rules corresponds the highly overlapped region. Figure 2 depicts the Venn diagram of probabilistic rules with highly overlapped region. This rule is a kind of probabilistic proposition with two statistical measures, which is an extension of Ziarko's variable precision model(VPRS) [6].[1]

3 Simplest Diagnostic Rules

3.1 Representation of Diagnostic Rules

The simplest probabilistic model is that which only uses classification rules which have high accuracy and high coverage. Such rules can be defined as:

$$\begin{aligned} R \overset{\alpha,\kappa}{\to} d \text{ s.t.} \quad & R = \vee_i R_i = \vee \wedge_j [a_j = v_k], \\ & \alpha_{R_i}(D) \geq \delta_\alpha \text{ and } \kappa_{R_i}(D) \geq \delta_\kappa, \end{aligned}$$

where δ_α and δ_κ denote given thresholds for accuracy and coverage, respectively. For the above example shown in Table 1, probabilistic rules for m.c.h. are given as follows (both δ_α and δ_κ are set to 0.75):

$$\begin{aligned} &[prod = 0] \rightarrow m.c.h.\ \alpha = 3/4 = 0.75,\ \kappa = 1.0, \\ &[nau = 0]\ \ \rightarrow m.c.h.\ \alpha = 3/3 = 1.0,\ \kappa = 1.0, \\ &[M1 = 1]\ \ \rightarrow m.c.h.\ \alpha = 3/4 = 0.75,\ \kappa = 1.0, \end{aligned}$$

3.2 An Rule Induction Algorithm

An rule induction algorithm is defined as Figure 1, which is discussed precisely in [8]. It is notable that rule induction of other type rules is derived by simple modification of this algorithm.

[1] This probabilistic rule is also a kind of *Rough Modus Ponens*[7].

```
procedure Induction of Classification Rules;
  var
    $i$ : integer;    $M, L_i$ : List;
  begin
    $L_1 := L_{er}$; /* $L_{er}$: List of Elementary Relations */
    $i := 1$;    $M := \{\}$;
    for $i := 1$ to $n$ do      /* $n$: Total number of attributes */
      begin
        while ( $L_i \neq \{\}$ ) do
          begin
            Sort $L_i$ with respect to the value of coverage;
            Select one pair $R = \wedge[a_i = v_j]$ from $L_i$,
            which have the largest value on coverage;
            $L_i := L_i - \{R\}$;
            if $(\kappa_R(D) \geq \delta_\kappa)$
              then do
                        if $(\alpha_R(D) \geq \delta_\alpha)$
                          then  do $S_{ir} := S_{ir} + \{R\}$; /* Include $R$ as Classification Rule */
                        $M := M + \{R\}$;
          end
        $L_{i+1}$ := (A list of the whole combination of the conjunction formulae in $M$);
      end
  end {Induction of Classification Rules };
```

Fig. 3. An Algorithm for Classification Rules

4 Focusing Mechanism

One of the characteristics in medical reasoning is a focusing mechanism, which is used to select the final diagnosis from many candidates[9,10]. For example, in differential diagnosis of headache, more than 60 diseases will be checked by present history, physical examinations and laboratory examinations. In diagnostic procedures, a candidate is excluded if a symptom necessary to diagnose is not observed.

This style of reasoning consists of the following two kinds of reasoning processes: exclusive reasoning and inclusive reasoning. Relations of this diagnostic model with another diagnostic model are discussed in [2]. The diagnostic procedure will proceed as follows (Figure 4): first, exclusive reasoning excludes a disease from candidates when a patient does not have a symptom which is necessary to diagnose that disease. Secondly, inclusive reasoning suspects a disease in the output of the exclusive process when a patient has symptoms specific to a disease. These two steps are modelled as usage of two kinds of rules, negative rules (or exclusive rules) and positive rules, the former of which corresponds to exclusive reasoning and the latter of which corresponds to inclusive reasoning. In the next two subsections, these two rules are represented as special kinds of probabilistic rules.

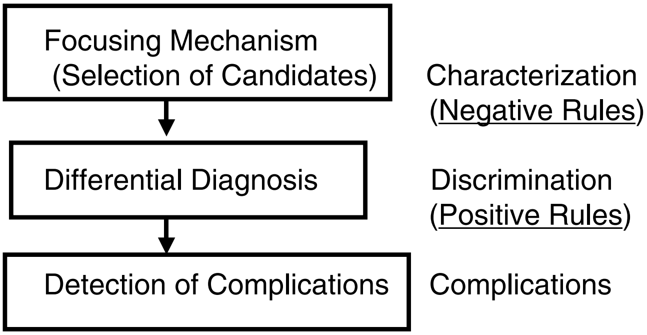


Fig. 4. Illustration of Focusing Mechanism

4.1 Positive Rules

A positive rule can be defined as a rule supported by only positive examples, which means that the classification accuracy of a rule is equal to 1.0. Thus, a positive rule is represented as:

$$R \to d \quad s.t. \qquad R = \wedge_j [a_j = v_k], \quad \alpha_R(D) = 1.0$$

In the above example, one positive rule of "m.c.h." is:

$$[nau = 0] \to m.c.h. \quad \alpha = 3/3 = 1.0.$$

This positive rule is often called deterministic rules. However, in this paper, we use a term, positive (deterministic) rules, because deterministic rules which is supported only by negative examples, called negative rules, is introduced as in the next subsection.

4.2 Negative Rules

Before defining a negative rule, let us first introduce an exclusive rule, the contrapositive of a negative rule[9]. An exclusive rule can be defined as a rule supported by all the positive examples, which means that the coverage of a rule is equal to 1.0.[2] Thus, an exclusive rule is represented as:

$$R \to d \quad s.t. \qquad R = \wedge_j [a_j = v_k], \quad \kappa_R(D) = 1.0.$$

In the above example, exclusive rule of "m.c.h." is:

$$[prod = 0] \wedge [nau = 0] \wedge [M1 = 1] \to m.c.h. \quad \kappa = 1.0,$$

It is notable that exclusive rule corresponds to an upper approximation of a target concept. For example, the set which supports the exclusive rule above is an upper approximation of m.c.h.

[2] Exclusive rules represent the necessity condition of a decision.

From the viewpoint of propositional logic, an exclusive rule should be represented as:

$$d \rightarrow \wedge_j [a_j = v_k],$$

because the condition of an exclusive rule correspond to the necessity condition of conclusion d. Thus, it is easy to see that a negative rule is defined as the contrapositive of an exclusive rule:

$$\vee_j \neg [a_j = v_k] \rightarrow \neg d,$$

which means that if a case does not satisfy any attribute value pairs in the condition of a negative rules, then we can exclude a decision d from candidates. For example, the negative rule of m.c.h. is:

$$\neg[prod = 0] \vee \neg[nau = 0] \vee \neg[M1 = 1] \rightarrow \neg m.c.h.$$

In summary, a negative rule is defined as:

$$\wedge_j \vee [a_j = v_k] \rightarrow \neg d \quad s.t. \quad \forall [a_j = v_k] \ \kappa_{[a_j = v_k]}(D) = 1.0,$$

where D denotes a set of samples which belong to a class d. It can be also called a deterministic rule, since a measure of negative concept, coverage is equal to 1.0.

In summary, positive and negative rules corresponds to positive and negative regions defined in rough sets. Figure 5 shows the Venn diagram of those rules.

4.3 Rule Induction Algorithm

An algorithm for induction of positive and negative rules is derived by simple modification of the algorithm in Figure 1: if the thresholds of accuracy and coverage is set to 0.0 and 1.0, respectively, the algorithm for negative rules will

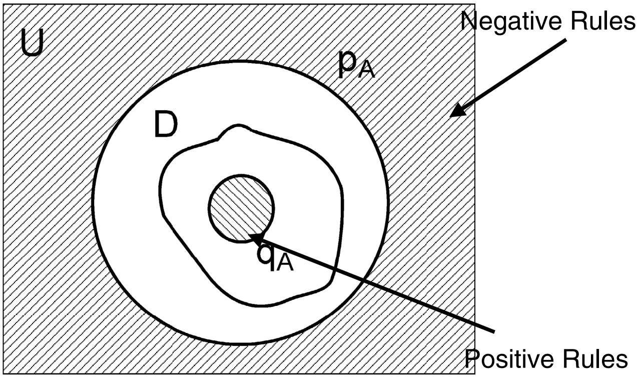


Fig. 5. Positive and Negative Rules as Overview

be obtained. On the other hand, if the thresholds of accuracy and coverage is set to 1.0 and 0.0, respectively, the algorithm for negative rules will be obtained.

It is notable that positive and negative rules can be extended to probabilistic versions, which is discussed precisely in [9].

5 Criteria Tables

5.1 Representation of Rules

Another characteristic reasoning in medicine is $m-$of$-n$ concepts, or criteria table, which is discussed in [11]. Criteria table for a disease d is described by n attributes, which are enough to make its diagnosis. If at least m attributes are observed in a patient, d should be suspected.

Langley discusses that this $m-$of$-n$ description can be rewritten as a simple linear combination of attribute-value pairs. Thus, he implements an induction of this description as an induction of threshold concepts.

However, a $m-$of$-n$ rule in medicine is not equivalent to a linear combination rule, which is a special kind of statistical discriminant functions[12]. Rather, this type of rule is based on relations between sets as follows.

1. If total n attributes are observed, a disease d is suspected with the highest accuracy. (The coverage is equal to 1.0).

2. If m attributes are satisfied, a disease d should be suspected with high accuracy. (The coverage is equal to 1.0).

3. If less than m attributes are satisfied, the probability of d is low. However, the coverage is equal to 1.0. Thus, $m-$of$-n$ concept is described as combination of exclusive rules (below, we call them *unit rules*) with the constraint that their accuracies are high:

$$\begin{aligned} R \to d \ s.t. \ R = \wedge_{j=1}^{i}[a_j = v_k](m \leq i \leq n) \\ \alpha_R(D) \geq \delta_\alpha, \kappa_{[a_j=v_k]}(D) = 1.0, \end{aligned}$$

which also satisfies that: if R is represented as $\wedge_{j=1}^{i}(i < m)$, then $\alpha_R(D) < \delta_\alpha$ holds.

For the above example in Table 1, exclusive rule of m.c.h. is:

$$[prod = 0] \wedge [nau = 0] \wedge [M1 = 1] \to m.c.h. \quad \kappa = 1.0, \alpha = 1.0$$

This attains the highest accuracy. If the threshold for accuracy is set to 0.75, then

$$\begin{aligned} [prod = 0] \to m.c.h. \ \kappa = 1.0, \alpha = 0.75, \\ [nau = 0] \to m.c.h. \ \kappa = 1.0, \alpha = 0.75, \ and \\ [M1 = 1] \to m.c.h. \ \ \kappa = 1.0, \alpha = 1.0. \end{aligned}$$

So, diagnostic rules for m.c.h. can be viewed as $1-$of-3 concept. In this way, combination of accuracy and coverage is also important to represent $m-$of$-n$ type rules.

5.2 Rule Induction Algorithm

An algorithm for induction of unit rules is derived by simple modification of the algorithm in Figure 1: if the thresholds of accuracy and coverage is set to δ and 1.0, respectively, then the algorithm for induction of each unit rule will be obtained. In this model, we should only add integration of unit rules after rule induction to obtain the total algorithm, which is not shown for the limitation of the space.

6 Conclusion

In this paper, rough set framework is introduced to model medical diagnostic rules. Acquired models show that the characteristics of medical reasoning reflect the concepts on approximation of rough sets, which explains why rough sets work well in medical domains.

References

1. Buchnan, B., Shortliffe, E.: Rule-Based Expert Systems. Addison-Wesley, New York (1984)
2. Tsumoto, S.: Automated extraction of medical expert system rules from clinical databases on rough set theory. Inf. Sci. 112, 67–84 (1998)
3. Tsumoto, S.: Extraction of experts decision rules from clinical databases using rough set model. Intelligent Data Analysis 2 (1998)
4. Pawlak, Z.: Rough Sets. Kluwer Academic Publishers, Dordrecht (1991)
5. Skowron, A., Grzymala-Busse, J.: From rough set theory to evidence theory. In: Yager, R., Fedrizzi, M., Kacprzyk, J. (eds.) Advances in the Dempster-Shafer Theory of Evidence, pp. 193–236. John Wiley & Sons, New York (1994)
6. Ziarko, W.: Variable precision rough set model. J. Comput. Syst. Sci. 46, 39–59 (1993)
7. Pawlak, Z.: Rough modus ponens. In: Proceedings of International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems 98, Paris (1998)
8. Tsumoto, S., Tanaka, H.: Automated knowledge acquisition from medical databases and its evaluation (1998)
9. Tsumoto, S., Tanaka, H.: Automated discovery of medical expert system rules from clinical databases based on rough sets. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining 96, Palo Alto, pp. 63–69. AAAI Press, California (1996)
10. Tsumoto, S.: Modelling medical diagnostic rules based on rough sets. In: Polkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS (LNAI), vol. 1424, pp. 475–482. Springer, Heidelberg (1998)
11. Langley, P.: Elements of Machine Learning. Morgan Kaufmann, San Francisco (1996)
12. McLachlan, G.: Discriminant Analysis and Statistical Pattern Recognition. John Wiley and Sons, New York (1992)

A Fundamentals of Accuracy and Coverage

A.1 Statistical Dependence

Let $P(R)$ and $P(D)$ be defined as: $P(R) = \frac{|R_A|}{|U|}$ *and* $P(D) = \frac{|D|}{|U|}$, where U denotes the total samples. Then, a index for statistical dependence ς_c is defined as:

$$\varsigma_R(D) = \frac{|R_A \cap D|}{|R_A||D|} = \frac{|U|P(R,D)}{P(R)P(D)},$$

where $P(R,D)$ denotes a joint probability of R and D $(P(R,D) = |R_A \cap D|/|U|)$. Since the formula $P(R,D) = P(R)P(D)$ is the definition of statistical independence, $\varsigma_R(D)$ measures the degree of statistical dependence. That is, If $\varsigma_R(D) > 1.0$, then R and D are dependent, other R and D are independent; especially, if $\varsigma_R(D)$ is equal to 1.0, they are statistically independent.

Theorem 1. *Lower approximation and upper approximation gives (strong) statistical dependent relations.*

Proof. Since $\alpha_R(D) = 1.0$ *for the lower approximation,* $\varsigma_R(D) = \frac{1}{P(D)} > 1.0$ *In the same way, for the upper approximation,* $\varsigma_R(D) = \frac{1}{P(R)} > 1.0$ □

Definition 2. *Let* U *be described by* n *attributes. A conjunctive formula* $R(i)$ *is defined as:* $R(i) = \bigwedge_{k=1}^{i} [a_i = v_i]$, *where index* i *is sorted by a given criteria, such as the value of accuracy. Then, the sequence of a conjunction is given as:* $R(i+1) = R(i) \wedge [a_{i+1} = v_{j+1}]$.

Since $R(i+1)_A = R(i)_A \cap [a_{i+1} = v_{i+1}]_A$, for this sequence, the following proposition will hold: $R(i+1)_A \subseteq R(i)_A$ Thus, the following theorem is obtained.

Theorem 2. *When we consider a sequence of conjunctive formula such that the value of accuracy should be increased, the statistical dependence will increase.*
Proof.

$$\varsigma_{R(i+1)}(D) = \frac{\alpha_{R(i+1)}(D)}{P(D)} \geq \frac{\alpha_{R(i)}(D)}{P(D)} = \varsigma_{R(i)}(D)$$

A.2 Tradeoff Between Accuracy and Coverage

Theorem 3 (Monotonicity of Coverage). *Let a sequence of conjunctive formula* $R(i)$ *given with* n *attributes. Then,*

$$\kappa_{R(i+1)}(D) \leq \kappa_{R(i)}(D).$$

Then, since accuracy and coverage has the following relation:

$$\frac{\kappa_R(D)}{\alpha_R(D)} = \frac{P(R)}{P(D)}. \qquad (1)$$

Since $P(R)$ will decrease with the sequence of conjunction, the following theorem is obtained.

Theorem 4. *Even if a sequence of conjunction for* R *is selected such that the value of accuracy increases monotonically,* $\kappa_R(D)$ *will decrease. That is, the decrease of* $\kappa_R(D)$ *is larger than the effect of the increase of* $\alpha_R(D)$*.* □

The Art of Granular Computing

Yiyu Yao

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
yyao@cs.uregina.ca

Abstract. The current research in granular computing is dominated by set-theoretic models such as rough sets and fuzzy sets. By recasting the existing studies in a wider context, we propose a unified framework of granular computing. The new framework extends results obtained in the set-theoretic setting and extracts high-level common principles from a wide range of scientific disciplines. The art of granular computing for problem solving emerges from the resulting common philosophy, methodology and information processing paradigm. Granular computing stresses not only the need for rigor, structure, conciseness and clarity, but also the importance of conscious effects and wisdom in using powerful strategies and heuristics in stating and solving problems.

Keywords: Granular computing triangle, structured thinking, structured problem solving, structured information processing.

1 Introduction

The advances of rough set theory have greatly influenced the development of granular computing [4,5,19,33,34,36,43,46,47,51,52,62,65,66]. Specifically, the philosophy and methodology of rough sets, centralized on the notions of indiscernibility and knowledge granularity, are fundamental to granular computing. It is fair to say that the plentiful results and applications of the theory of rough sets motivate many researchers to study granular computing.

An underlying notion of rough set theory is an equivalence relation representing indiscernibility of objects and the induced partition of a universe [45,46,47]. Suppose a finite universe of objects is described by a finite set of attributes in the form of an information table. Different equivalence relations can be constructed based on distinct subsets of attributes [45]. One may interpret a partition as a simple flat granulated view of the universe with each equivalence class as a granule. Under this view, rough set analysis deals with approximation and reasoning with partitions of different levels of granularity [45,66].

The basic ideas of partitioning a universe for problem solving have also been used in many studies such as the partition model of databases [31], the theory of granularity [21], and the quotient space theory [74,75]. Each of these studies is formulated differently to deal with a different type of problems. In spite of their differences, they all share two common features with the rough set theory. First,

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 101–112, 2007.

they all consider different descriptions of the same problem at multiple levels of granularity. This allows us to focus on solving a problem at the most appropriate level of granularity by ignoring unimportant and irrelevant details. The second feature is that multilevel descriptions are linked together to form a hierarchical structure. In other words, levels with differing granularity are partially ordered. This allows us to change granularity easily at different stages of problem solving.

These two features are common to problem solving activities across many branches of science [68,69,70]. Although scientists in different disciplines study different subject matters and use different formulations, they all employ remarkably common structures for describing problems and apply common principles, strategies, and heuristics for problem solving [7,39]. Our understanding and formulation of granular computing is based on such high-level features [66,67,68,69]. We attempt to extract the common domain-independent principles, strategies and heuristics that have been applied either explicitly or implicitly in many disciplines. The results enable us to arrive at a unified framework of granular computing for problem solving from three perspectives [69].

The main objective of this paper is to explore granular computing as the creative art of problem solving. We propose and examine a trinity framework of granular computing.

2 Granular Computing as a New Field of Study

In the past ten years, many researchers have focused their efforts on the development of a new research field under the umbrella name of granular computing [4,23,35,43,46,48,51,52,63,64,72,73]. Extensive results and applications demonstrate the need for and the potential of granular computing. However, the advance of granular computing suffers from the lack of a conceptual framework that enables us to answer some of the fundamental questions. In order to justify the existence of granular computing as a new field in its own right, we need to address these questions.

It is a well-accepted fact that the basic ideas, principles and strategies of granular computing appear in many branches of science and different fields of computer science [4,70,72]. This immediately raises the following questions:

- What is new and unique in granular computing?
- What are the contributions of granular computing?
- What are the relations between granular computing and other fields?
- What are the scopes and goals for the study of granular computing?

The answers to these questions show the necessity for the study of granular computing, provide the context to which granular computing fits, and set the goals of research on granular computing.

People solve different problems by using some common principles. However, one can make several important observations regarding their actual usages. First, these principles are scattered over many places in isolation without being synthesized into an integrated whole. Second, they are normally explained with reference to domain-specific knowledge and thus are buried deeply in minute details.

Third, the same principles are discussed in different languages and notations. Fourth, these principles are typically used either implicitly or subconsciously, for a formal documentation does not exist. They are not readily accessible for many people to use. Sometimes, the same principles are reinvented time and again in the same or different fields. By introducing granular computing as a new field of study, we attempt to resolve such problems.

To a large extent, the emergence of granular computing is motivated by the same reasons that led to the introduction of general systems theory several decades ago [9,29,44,56]. As a new field of research, granular computing is a study of the art of problem solving. It has two unique tasks. One is to extract high-level commonalities of different disciplines and to synthesize their results into an integrated whole by ignoring low-level details. The other is to make explicit ideas hidden in discipline-specific discussions in order to arrive at a set of discipline-independent principles.

What makes granular computing new and unique is not its individual principles, methodologies, and strategies. Each of them has been extensively studied by authors in many fields. Granular computing contributes by synthesizing, integrating, and studying them in a uniform way. Through granular computing, we attempt to achieve the following goals:

- to make implicit principles explicit,
- to make invisible principles visible,
- to make domain-specific principles domain-independent,
- to make subconscious effects conscious.

It is possible to empower more people with effective strategies for problem solving tasks. One can consciously apply the principles of granular computing in solving a wide range of problems. It is also possible to prevent a waste of research efforts rediscovering or reinventing these principles.

Granular computing is a multidisciplinary study that emerged from existing disciplines and fields of study. For example, in addition to rough sets [45,46,47] and fuzzy sets [72,73], granular computing can draw results from the following:

- philosophy and philosophy of science [37,49],
- research methods [7,39],
- cognitive science and cognitive psychology [53,58],
- human problem solving [42],
- general systems theory [9,29,56],
- synectics [18],
- hierarchy theory [1,44,55,56,60],
- cluster analysis [2],
- social networks [3,24],
- artificial intelligence [16,20,21,27,74],
- learning [11,50,57],
- computer programming [12,28,30,61],
- information processing [4,26,38],

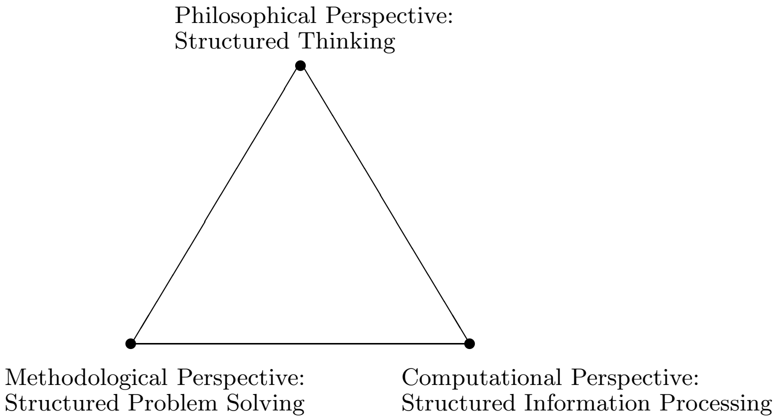

Philosophical Perspective:
Structured Thinking
Methodological Perspective:
Structured Problem Solving
Computational Perspective:
Structured Information Processing

Fig. 1. The granular computing triangle

- teaching and instruction [13,54],
- rhetoric and writing [14,22,41,71].

This list is not intended to be exhaustive, but an illustration to show the diversity of disciplines where the principles of granular computing can be observed.

A theory of granular computing may be established by extracting, sorting, integrating, synthesizing, and interpreting a set of generally applicable principles, methods, and strategies for problem solving. In the past few years, many researchers have made significant progress on concrete models and methods of granular computing. In the meantime, one can also observe a number of studies that simply restate existing results using the terminology of granular computing or reinvent them in a different context. A conceptual study of granular computing may free us from similar pitfalls.

3 The Granular Computing Triangle

Granular computing can be studied from three perspectives that are unified and based on the notion of granular structures. The granular computing triangle of Figure 1 represents this trinity view. In the philosophical perspective, granular computing deals with structured thinking. It attempts to extract and formalize human thinking. In the methodological perspective, granular computing concerns structured problem solving. It aims to study methods and techniques for systematic problem solving. In the computational perspective, granular computing is a paradigm of structured information processing. It addresses the problems of information processing in the abstract, in the brain, and in machines. Each perspective supports the other two perspectives. What integrates them is the granular structures that represent the real world at multiple levels of granularity. By emphasizing on structures, granular computing leads to structured solutions to real-world problems.

3.1 Granular Structures

A primitive notion of granular computing is a granule representing a part of a whole. Like systems theory, granular computing explores the composition of parts, their interrelationships, and connections to the whole. A real-world problem normally consists of a web of interacting and interrelated parts [9]. In order to have a practical understanding and solution, it is necessary to extract approximate structures that are tractable and easy to analyze. Granular computing exploits structures in terms of granules, levels, and hierarchies based on multilevel and multiview representations [69].

A granule plays two distinctive roles. It may be an element of another granule and is considered to be a part forming the other granule. It may also consist of a family of granules and is considered to be a whole. Its particular role is determined by our focal points at different stages of problem solving. This part-whole relationship suggests a partial ordering of granules. It is possible to derive a hierarchical structure. The term hierarchy is used to denote such a structure that consists of a family of interacting and interrelated granules, and each of them can be, in turn, a hierarchical structure. Trees and lattices are typical examples of hierarchical structures. Another example is the notion of rule complex, introduced and elaborated by Burns and Gomolińska [8,17] within the generalized game theory. We may view a hierarchy as a structure of (partially) ordered multiple levels. Each level is made up of a family of granules. Hierarchical structures not only make a complex problem more easily understandable, but also lead to efficient, although perhaps approximate, solutions.

In building a hierarchical structure, we need to have a vertical separation of levels and a horizontal separation of granules at the same hierarchical level. These separations explore the notion of approximations and a loose coupling of parts [9,56]. In forming a granule, one may ignore the subtle differences between its elements as well as their individual connections to others. That is, a group of elements may be treated approximately as a whole when studying their relations to others. Each level may be viewed as a representation of a problem at a specific level of granularity. The relationship between levels can be interpreted in terms of abstraction, control, complexity, detail, resolution, etc.

A hierarchy represents the results of a study of a problem from one particular angle or point-of-view. Some useful information may be lost with a hierarchy instead of a web. For the same problem, many interpretations and descriptions may co-exist [6,10]. It may be necessary to construct and compare multiple hierarchies [24]. A comparative study of those hierarchies may provide a complete understanding of the problem.

In summary, granular computing exploits multilevel and multiview representations in problem solving. A hierarchy represents one view of a problem with multiple levels of granularity. Depending on different contexts of applications, we may have data granulation, information granulation, and knowledge granulation corresponding to granular data structures, granular information structures, and granular knowledge structures.

3.2 Structured Thinking

Granular computing, as structured thinking, integrates two complementary philosophical views dealing with the complexity of real-world problems, namely, the traditional reductionist thinking and the more recent systems thinking. It stresses the importance of conscious effects in thinking with hierarchical structures.

According to reductionist thinking, a complex system or problem can be divided into simpler and more fundamental parts, and each part can be further divided. An understanding of the system can be reduced to the understanding of its parts. In other words, we can deduce fully the properties of the system based solely on the properties of its parts. In contrast, systems thinking shifts from parts to the whole, in terms of connectedness, relationships, and context [9,29]. A complex system is viewed as an integrated whole consisting of a web of interconnected, interacting, and highly organized parts. The properties of the whole are not present in any of its parts, but emerge from the interactions and relationships of the parts.

The reductionist thinking and systems thinking agree on the modeling of a complex system in terms of the whole and parts, but differ in how to make inference with the parts. Based on this common hierarchical structure, granular computing attempts to unify reductionist thinking and systems thinking into structured thinking.

3.3 Structured Problem Solving

Structured thinking leads to a perception and understanding of a real-world problem in terms of multilevel and multiview representations. These structures play a crucial role in problem solving. Granular computing is structured problem solving guided by structured thinking.

Structured problem solving methods and strategies have been extensively studied by many authors. A convincing way to show the effectiveness of granular computing is to present a set of principles and to demonstrate the working of these principles in real-world applications. We present three such principles:

- the principle of multilevel granularity,
- the principle of focused effort,
- the principle of granularity conversion.

The first principle emphasizes the importance of modeling in terms of hierarchical structures. Once such structures are obtained, the second principle calls for attention on the focal point at a particular stage of problem solving. The third principle links the different stages in this process.

Although principles of granular computing are named differently in different disciplines, they are indeed the same at a more abstract level. We briefly summarize the applications of such ideas and principles in several related areas:

Concept formulation and learning: A concept represents a basic unit of human thought and is commonly labeled by a word of a natural language. Hierarchical structures are commonly used in organizing human knowledge [49,54]. To a large extent, human learning is a good example to demonstrate the working principles of granular computing, i.e., attention and changing of attention.

Structured programming: Hierarchical structures are a central notion to structured programming [12,30]. In this context, a granule may be viewed as a program module. The stepwise refinement process explores multilevel development of a full program, from a brief high-level description to the final complete program [61].

Structured proofs: Following the results of structured programming, several authors studied structured methods for developing, teaching and communicating mathematical proofs [15,32]. In particular, a structured method arranges the proof in levels and proceeds in a top-down manner. A level consists of short autonomous modules, each embodying one major idea of the proof to be further concretized in the subsequent levels. The process continues by supplying more details of the higher levels until a complete proof is reached.

Structured writing: Writing may be viewed as a problem solving process and task [14,41,71]. A simple idea is described by a paragraph consisting of several sentences. A point-of-view is jointly described and supported by several ideas. A theme emerges from a few different points-of-view. The units of writing are sometimes referred to as information blocks [22], issues [14], ideas [41], and units of experience [71]. For effective communication, one needs to organize them into a hierarchical structure, referred to as an issue tree [14], a pyramid structure of ideas [41], or a hierarchically structured system of units of experience [71]. Like structured programming, structured writing may be viewed as a stepwise refinement that produces a full article.

Two important features can be observed from these studies. One is the construction of building blocks (i.e., granules) and the other is the arrangement of blocks into a hierarchical structure. The ideas, principles, proverbs, maxims, and strategies from these fields can be easily transferred to each other.

Human concept formation and learning determine, in principle, the ways to produce easily-understandable solutions to a problem. For example, the chunking principle underlying human memory [40] suggests a hierarchical structure used in writing [22,41]. The hierarchical structures of complex systems [56] are applicable to the process of writing if one considers an article to be a complex system that has evolved through time [71]. On the other hand, the styles of programming [25,30] are influenced by styles of writing English prose [59]. Structured programming in turn offers solutions to structured mathematical proofs [15,32]. In summary, these examples provide us convincing evidence that supports the study of granular computing. Instead of reinventing the same principles and strategies, one can focus on their applications across many disciplines.

3.4 Structured Information Processing

In the information processing paradigm [4], granular computing works with a pyramid consisting of different-sized information granules. This structured information processing is a necessary feature of any knowledge-intensive system.

Two notions of structured information processing are representation and process [38]. A representation is a formal system that makes explicit certain entities or types of information and a specification of how the system does it. The result is called a description of the entity in the representation. A process may simply be interpreted as actions or procedures for carrying out information processing tasks. In general, a representation determines the effectiveness of processes.

A representation of granules must capture their essential features and make explicit a particular aspect of their physical meanings. It needs to be closely connected to the representations of granular structures with respect to granules, levels, and hierarchies. Processes of granular computing may be broadly divided into the two classes: granulation and computation with granules [64,70]. Granulation processes involve the construction of the building blocks and structures, namely, granules, levels, and hierarchies. Computation processes systematically explore the granular structures. This involves two-way communications up and down in a hierarchy, as well as switching between levels.

Structured information processing is a stepwise refinement process. At a higher level, one may produce an approximate, a partial, or a schematic solution. The latter is to be made more precise, complete, and detailed at a lower level. The process stops when a desirable (approximate) solution is obtained.

4 Conclusion

A new understanding of granular computing is presented. Granular computing draws extensive results from existing disciplines and offers its own insights and solutions. Its future depends critically on a right balance between the two. We need to draw results from classical thinking and explore new ways of creative thinking. The proposed trinity framework casts granular computing into a wider context. The art of granular computing can be fully appreciated from the philosophical perspective as structured thinking, from the methodological perspective as structured problem solving, and from the computational perspective as structured information processing.

References

1. Ahl, V., Allen, T.F.H.: Hierarchy Theory, A Vision, Vocabulary and Epistemology. Columbia University Press, New York (1996)
2. Anderberg, M.R.: Cluster Analysis for Applications. Academic Press, New York (1973)
3. Arrow, H., McGrath, J.E., Berdahl, J.L.: Small Groups as Complex Systems: Formation, Coordination, Development, and Applications. Sage Publications, Thousand Oaks (2000)

4. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Boston (2002)
5. Bargiela, A., Pedrycz, W.: The roots of granular computing. In: Proceedings of 2006 IEEE International Conference on Granular Computing, pp. 806–809 (2006)
6. Bateson, G.: Mind and Nature: A Necessary Unity, E.P. Dutton, New York (1979)
7. Beveridge, W.I.B.: The Art of Scientific Investigation, Vintage Books, New York (1967)
8. Burns, T.R., Gomolińska, A.: The theory of socially embedded games: the mathematics of social relationships, rule complexes, and action modalities. Quality and Quantity 34, 379–406 (2000)
9. Capra, F.: The Web of Life, Anchor Books, New York (1997)
10. Chen, Y.H., Yao, Y.Y.: Multiview intelligent data analysis based on granular computing. In: Proceedings of 2006 IEEE International Conference on Granular Computing, pp. 281–286 (2006)
11. Conway, C.M., Christiansen, M.H.: Sequential learning in non-human primates. Trends in Cognitive Sciences 12, 539–546 (2001)
12. Dahl, O.-J., Dijkstra, E.W., Hoare, C.A.R.: Structured Programming. Academic Press, New York (1972)
13. Doignon, J.P., Falmagne, J.C.: Knowledge Spaces. Springer, Berlin (1999)
14. Flower, L.: Problem-Solving Strategies for Writing, Harcourt Brace Jovabovich, Inc. New York (1981)
15. Friske, M.: Teaching proofs: a lesson from software engineering. American Mathematical Monthly 92, 142–144 (1995)
16. Giunchglia, F., Walsh, T.: A theory of abstraction. Artificial Intelligence 56, 323–390 (1992)
17. Gomolińska, A.: Fundamental mathematical notions of the theory of socially embedded games: a granular computing perspective. In: Pal, S.K., Polkowski, L., Skowron, A. (eds.) Rough-Neural Computing: Techniques for Computing with Words, pp. 411–434. Springer, Berlin (2004)
18. Gordon, W.J.J.: Synectics: The Development of Creative Capacity, Harper and Row, New York (1961)
19. Grzymala-Busse, J.W., Rzasa, W.: Local and global approximations for incomplete data. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 244–263. Springer, Heidelberg (2006)
20. Hawkins, J., Blakeslee, S.: On Intelligence, Henry Holt and Company, New York (2004)
21. Hobbs, J.R.: Granularity. In: Proceedings of the Ninth International Joint Conference on Artificial Intelligence, pp. 432–435 (1985)
22. Horn, R.E.: Structured writing as a paradigm. In: Romiszowski, A., Dills, C. (eds.) Instructional Development: State of the Art, Educational Technology Publications, Englewood Cliffs (1998)
23. Inuiguchi, M., Hirano, S., Tsumoto, S. (eds.): Rough Set Theory and Granular Computing. Springer, Berlin (2003)
24. Jeffries, V., Ransford, H.E.: Social Stratification: A Multiple Hierarchy Approach, Allyn and Bacon, Boston (1980)
25. Kernighan, B.W., Plauger, P.J.: The Elements of Programming Style. McGraw-Hill, New York (1978)
26. Klahr, D., Kotovsky, K. (eds.): Complex Information Processing: The Impact of Herbert A. Simon. Lawrence Erlbaum Associates, Hillsdale (1989)

27. Knoblock, C.A.: Generating Abstraction Hierarchies: An Automated Approach to Reducing Search in Planning. Kluwer Academic Publishers, Boston (1993)
28. Knuth, D.E.: The Art of Computer Programming, 3rd edn. Addison-Wesley, Matssachusetts (1997)
29. Laszlo, E.: The Systems View of the World: The Natural Philosophy of the New Developments in the Science, George Brasiller, New York (1972)
30. Ledgard, H.F., Gueras, J.F., Nagin, P.A.: PASCAL with Style: Programming Proverbs, Hayden Book Company, Rechelle Park, New Jersey (1979)
31. Lee, T.T.: An information-theoretic analysis of relational databases – part I: data dependencies and information metric. IEEE Transactions on Software Engineering SE-13, 1049–1061 (1987)
32. Leron, U.: Structuring mathematical proofs. American Mathematical Monthly 90, 174–185 (1983)
33. Lin, T.Y.: Granular Computing on binary relations I: data mining and neighborhood systems, II: rough set representations and belief functions. In: Skowron, A., Polkowski, L. (eds.) Rough Sets In Knowledge Discovery, pp. 107–140. Physica-Verlag (1998)
34. Lin, T.Y.: Granular computing: structures, represenations, and applications. In: Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.) RSFDGrC 2003. LNCS (LNAI), vol. 2639, pp. 16–24. Springer, Heidelberg (2003)
35. Lin, T.Y., Yao, Y.Y., Zadeh, L.A. (eds.): Data Mining, Rough Sets and Granular Computing. Physica-Verlag, Heidelberg (2002)
36. Liu, Q., Wang, Q.Y.: Granular logic with closeness relation "$\sim_\lambda$" and its reasoning. In: Ślęzak, D., Wang, G., Szczuka, M., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 709–717. Springer, Heidelberg (2005)
37. Losee, J.: A Historical Introduction to the Philosphy of Science, 3rd edn. Oxford University Press, Oxford (1993)
38. Marr, D.: Vision, A Computational Investigation into Human Representation and Processing of Visual Information, W.H. Freeman and Company, San Francisco (1982)
39. Martella, R.C., Nelson, R., Marchard-Martella, N.E.: Research Methods: Learning to Become a Critical Research Consumer, Allyn and Bacon, Boston (1999)
40. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological Review 63, 81–97 (1956)
41. Minto, B.: The Pyramid Princile: Logic in Writing and Thinking. Prentice Hall/Financial Times, London (2002)
42. Newell, A., Simon, H.A.: Human Problem Solving. Prentice-Hall, Englewood Cliffs (1972)
43. Nguyen, H.S., Skowron, A., Stepaniuk, J.: Granular computing: a rough set approach. Computational Intelligence 17, 514–544 (2001)
44. Pattee, H.H. (ed.): Hierarchy Theory, The Challenge of Complex Systems, George Braziller, New York (1973)
45. Pawlak, Z.: Rough Sets, Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
46. Pawlak, Z.: Granularity, multi-valued logic, Bayes' theorem and rough sets. In: Lin, T.Y., Yao, Y.Y., Zadeh, L.A. (eds.) Data Mining, Rough Sets and Granular Computing, pp. 487–498. Physica-Verlag, Heidelberg (2002)
47. Pawlak, Z., Skowron, A.: Rough sets: some extensions. Information Science 177, 28–40 (2007)
48. Pedrycz, W. (ed.): Granular Computing: An Emerging Paradigm. Physica-Verlag, Heidelberg (2001)

49. Peikoff, L.: Objectivism: The Philosophy of Ayn Rand, Dutton, New York (1991)
50. Poggio, T., Smale, S.: The mathematics of learning: dealing with data. Notices of the AMS 50, 537–544 (2003)
51. Polkowski, L.A: model of granular computing with applications: granules from rough inclusions in information systems. In: Proceedings of 2006 IEEE International Conference on Granular Computing, pp. 9–16 (2006)
52. Polkowski, L., Skowron, A.: Towards adaptive calculus of granules. In: Proceedings of 1998 IEEE International Conference on Fuzzy Systems, pp. 111–116 (1998)
53. Posner, M.I. (ed.): Foundations of Cognitive Science. MIT Press, Cambridge (1989)
54. Reif, F., Heller, J.: Knowledge structure and problem solving in physics. Educational Psychologist 17, 102–127 (1982)
55. Salthe, S.N.: Evolving Hierarchical Systems, Their Structure and Representation. Columbia University Press, New York (1985)
56. Simon, H.A.: The Sciences of the Artificial. The MIT Press, Massachusetts (1969)
57. Skowron, A., Synak, P.: Hierarchical information maps. In: Ślęzak, D., Wang, G., Szczuka, M., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 622–631. Springer, Heidelberg (2005)
58. Solso, R.L., MacLin, M.K., MacLin, O.H.: Cognitive Psychology, 7th edn. Allyn and Bacon, New York (2005)
59. Strunk, W., White, E.B.: The Elements of Style, Allyn and Bacon, Needham Heights, Massachusetts (2000)
60. Whyte, L.L., Wilson, A.G., Wilson, D. (eds.): Hierarchical Structures. American Elsevier Publishing Company, Inc, New York (1969)
61. Wirth, N.: Program development by stepwise refinement. Communications of the ACM 14, 221–227 (1971)
62. Yao, J.T.: Information granulation and granular relationships. In: Proceedings of the IEEE Conference on Granular Computing, pp. 326–329 (2005)
63. Yao, Y.Y.: Granular computing using neighborhood systems. In: Roy, R., Furuhashi, T., Chawdhry, P.K. (eds.) Advances in Soft Computing: Engineering Design and Manufacturing, pp. 539–553. Springer, London (1999)
64. Yao, Y.Y.: Granular computing: basic issues and possible solutions. In: Proceedings of the 5th Joint Conference on Information Sciences, pp. 186–189 (2000)
65. Yao, Y.Y.: Information granulation and rough set approximation. International Journal of Intelligent Systems 16, 87–104 (2001)
66. Yao, Y.Y.: A partition model of granular computing. LNCS Transactions on Rough Sets. 1, 232–253 (2004)
67. Yao, Y.Y.: Granular computing. Computer Science (Ji Suan. Ji. Ke. Xue) 31, 1–5 (2004)
68. Yao, Y.Y.: Perspectives of granular computing. In: Proceedings of 2005 IEEE International Conference on granular computing, vol. 1, pp. 85–90 (2005)
69. Yao, Y.Y.: Three perspectives of granular computing. Journal of Nanchang Institute of Technology 25, 16–21 (2006)
70. Yao, Y.Y.: Granular computing for data mining. In: Proceedings of SPIE Conference on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, pp. 1-12 (paper no. 624105) (2006)
71. Young, R.E., Becker, A.L., Pike, K.L.: Rhetoric: Discovery and Change, Harcourt Brace Jovabovich, Inc. New York (1970)
72. Zadeh, L.A.: Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets. and Systems 19, 111–127 (1997)

73. Zadeh, L.A.: Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. Soft. Computing 2, 23–25 (1998)
74. Zhang, B., Zhang, L.: Theory and Applications of Problem Solving, North-Holland, Amsterdam (1992)
75. Zhang, L., Zhang, B.: The quotient space theory of problem solving. Fundamenta Informatcae 59, 287–298 (2004)

Dependencies in Structures of Decision Tables

Wojciech Ziarko

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2

Abstract. The presentation is focused on the introduction and the investigation of probabilistic dependencies between attribute-defined partitions of a universe in hierarchies of probabilistic decision tables learned from data. The dependencies are expressed through two measures: the probabilistic generalization of the Pawlak's measure of the dependency between attributes and the expected certainty gain measure. The expected certainty gain measure reflects the subtle grades of probabilistic dependence of events. The measures are reviewed and it is shown how they can be extended to dependencies existing in hierarchical structures of decision tables.

1 Introduction

In applications, the decision tables are typically used for making predictions about the value of the decision attribute based on combinations of values of condition attributes, as measured on new, previously unseen objects. However, the tables often suffer from the following problems related to the fact that they are computed based on the proper subset of the universe.

Firstly, The decision table may have excessive decision boundary, often due to poor quality of condition attributes, which may be weakly correlated with the decision attribute. The excessive decision boundary leads to the excessive number of incorrect predictions. Secondly, The decision table may be highly incomplete, i.e. excessively many new measurement vectors of condition attributes of new objects are not matched by any combination of condition attribute values present in the decision table. Such a highly incomplete decision table leads to an excessive number of new unrepresented observations, for which the prediction of the decision attribute value is not possible.

With weak condition attributes, increasing their number does not rectify the problem (1). This is due to the fact that increasing the number of attributes results in the exponential explosion of the complexity of learning of the decision table, leading to the rapid increase of the degree of the decision table incompleteness [8]. In general, the decision boundary reduction problem is conflicting with the decision table incompleteness minimization problem. To deal with these fundamental difficulties, an HDTL approach was proposed [6]. The approach is focused on learning hierarchical structures of decision tables rather than learning individual tables, subject to learning complexity constraints . In this approach, a linear hierarchy of decision tables is formed, in which the parent layer decision

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 113–121, 2007.

boundary defines a universe of discourse for the child layer table. The decision tables on each layer are size-limited by reducing the number of condition attributes, thus bounding their learning complexity [8]. Each layer contributes a degree of decision boundary reduction, while providing a shrinking decision boundary to the next layer. In this way, even in the presence of relatively weak condition attributes, a significant total boundary reduction can be achieved, while preserving the constraints on the complexity of learning on each level.

Similar to single layer decision table, the hierarchy of decision tables needs to be evaluated from the point of view of its quality as a potential classifier of new observations. The primary evaluative measure for decision tables introduced by Pawlak is the measure of partial functional dependency between attributes [1] and its probabilistic extension [7]. Another measure is the recently introduced expected gain measure which captures more subtle probabilistic associations between attributes [7]. In this paper, these measures are reviewed and generalized to the hierarchical structures of decision tables. A simple recursive method of their computation is also discussed. The measures, referred to as γ and λ measures respectively, provide a tool for non-experimental assessment of decision table-based classifiers derived from data.

2 Attribute-Based Approximation Spaces

In this section, we briefly review the essential assumptions, definitions and notations of the rough set theory in the context of probability theory. We assume that all subsets $X \subseteq U$ under consideration are measurable with $0 < P(X) < 1$ i.e. they are likely to occur but their occurrence is not certain. We also assume that observations about objects are expressed through values of *attributes*, which are functions $a : U \rightarrow V_a$, where V_a is a finite set of values called the *domain*. The attributes represent some properties of the objects in U. The attributes fall into two disjoint categories: C called *condition attributes*, and $D = \{d\}$ called *decision attributes.* In many applications, attributes are functions obtained by discretizing values of real-valued variables representing measurements taken on objects $e \in U$.

As individual attributes, any non-empty subset of attributes $B \subseteq C \cup D$ defines a mapping from the set of objects U into the set of vectors of values of attributes in B. This leads to the idea of the equivalence relation on U, called indiscernibility relation $IND_B = \{(e_1, e_2) \in U : B(e_1) = B(e_2)\}$. According to this relation, objects having identical values of attributes in B are equivalent, that is, indistinguishable in terms of values of attributes in B . The collection of classes of identical objects will be denoted as U/B and the pair $(U, U/B)$ will be called an *approximation space.*

The object sets $G \in U/C \cup D$, will be referred to as *atoms.* The sets $E \in U/C$ will be referred to as *elementary sets.* The sets $X \in U/D$ will be called *decision categories.* Each elementary set $E \in U/C$ and each decision category $X \in U/D$ is a union of some atoms. That is, $E = \cup\{G \in U/C \cup D : G \subseteq E\}$ and

$X = \cup\{G \in U/C \cup D : G \subseteq F\}$. Each atom $G \in U/C \cup D$ is assigned a *joint probability* $P(G)$, which is normally estimated from collected data.

From our initial assumption and from the basic properties of the probability measure P, follows that for all atoms $G \in U/C \cup D$, we have $0 < P(G) < 1$ and $\sum_{G \in U/C \cup D} P(G) = 1$. Based on the joint probabilities of atoms, probabilities of elementary sets E and of a decision category X can be calculated by $P(E) = \sum_{G \subseteq E} P(G)$. The probability $P(X)$ of the decision category X in the universe U is the *prior probability* of the category X. It represents the confidence in the occurrence of the decision category X in the absence of any information expressed by attribute values. The *conditional probability* of a decision category X, $P(X|E) = \frac{P(X \cap E)}{P(E)}$, conditioned on the occurrence of the elementary set E, represents the degree of confidence in the occurrence of the decision category X, given information indicating that E occurred. The conditional probability can be expressed in terms of joint probabilities of atoms by $P(X|E) = \frac{\sum_{G \subseteq X \cap E} P(G)}{\sum_{G \subseteq E} P(G)}$. This allows for simple computation of the conditional probabilities of decision categories.

3 Variable Precision Rough Set Model

One of the main objectives of rough set theory is the formation and analysis of approximate definitions of otherwise undefinable sets [1]. The approximate or rough definitions, in the form of lower approximation and boundary area of a set, allow for determination of an object's membership in a set with varying degrees of certainty. The lower approximation permits for uncertainty-free membership determination, whereas the boundary defines an area of objects which are not certain, but possible, members of the set [1]. The variable precision model of rough sets (VPRSM)[5][7] extends upon these ideas by parametrically defining the positive region as an area where the certainty degree of an object's membership in a set is relatively high, the negative region as an area where the certainty degree of an object's membership in a set is relatively low, and by defining the boundary as an area where the certainty of an object's membership in a set is deemed neither high nor low.

The defining criteria in the VPRSM are expressed in terms of conditional probabilities and of the prior probability $P(X)$ of the set X in the universe U. In the context the attribute-value representation of sets of the universe U, as described in the previous section, we will assume that the sets of interest are decision categories $X \in U/D$. Two *precision control* parameters are used: the *lower limit* l, $0 \leq l < P(X) < 1$, representing the highest acceptable degree of the conditional probability $P(X|E)$ to include the elementary set E in the *negative region* of the set X; and the *upper limit* u, $0 < P(X) < u \leq 1$, reflecting the least acceptable degree of the conditional probability $P(X|E)$ to include the elementary set E in the positive region, or *u-lower approximation* of the set X. The *l-negative region* of the set X, denoted as $NEG_l(X)$ is defined by:

$$NEG_l(X) = \cup\{E : P(X|E) \leq l\} \tag{1}$$

The l-negative region of the set $X$is a collection of objects for which the probability of membership in the set X is *significantly lower* than the prior probability $P(X)$. The u-positive region of the set X, $POS_u(X)$ is defined as

$$POS_u(X) = \cup\{E : P(X|E) \geq u\}. \tag{2}$$

The u-positive region of the set X is a collection of objects for which the probability of membership in the set X is *significantly higher* than the prior probability $P(X)$. The objects which are not classified as being in the u-positive region nor in the l-negative region belong to the (l, u)-boundary region of the decision category X, denoted as

$$BNR_{l,u}(X) = \cup\{E : l < P(X|E) < u\}. \tag{3}$$

The boundary is a specification of objects about which it is known that their associated probability of belonging, or not belonging to the decision category X, is not much different from the prior probability of the decision category $P(X)$.

4 Structures of Decision Tables Acquired from Data

To describe functional or partial functional connections between attributes of objects of the universe U, Pawlak introduced the idea of decision table acquired from data [1]. The probabilistic decision tables and their hierarchies extend this idea into probabilistic domain by forming representations of probabilistic relations between attributes.

For the given decision category $X \in U/D$ and the set values of the VPRSM lower and upper limit parameters l and u, we define the *probabilistic decision table* $DT^{C,D}_{l,u}$ as a mapping $C(U) \rightarrow \{POS, NEG, BND\}$ derived from the classification table as follows:

The mapping is assigning each tuple of values of condition attribute values $t \in C(U)$ to its unique designation of one of VPRSM approximation regions $POS_u(X)$, $NEG_l(X)$ or $BND_{l,u}(X)$, the corresponding elementary set E_t is included in, along with associated elementary set probabilities $P(E_t)$ and conditional probabilities $P(X|E_t)$:

$$DT^{C,D}_{l,u}(t) = \begin{cases} (P(E_t), P(X|E_t), POS) \Leftrightarrow E_t \subseteq POS_u(X) \\ (P(E_t), P(X|E_t), NEG) \Leftrightarrow E_t \subseteq NEG_l(X) \\ (P(E_t), P(X|E_t), BND) \Leftrightarrow E_t \subseteq BND_{l,u}(X) \end{cases} \tag{4}$$

The probabilistic decision table is an approximate representation of the probabilistic relation between condition and decision attributes via a collection of uniform size probabilistic rules corresponding to rows of the table. An example probabilistic decision table is shown in Table 1. The probabilistic decision tables are most useful for decision making or prediction when the relation between condition and decision attributes is largely non-deterministic. However, they suffer from the inherent contradiction between the accuracy and completeness. In the presence of boundary region, higher accuracy, i.e. reduction of boundary region,

Table 1. An example of probabilistic decision table

a	b	c	$P(E)$	$P(X\|E)$	**Region**
1	1	2	0.23	1.00	**POS**
1	0	1	0.33	0.61	**BND**
2	2	1	0.11	0.27	**BND**
2	0	2	0.01	1.00	**POS**
0	2	1	0.32	0.06	**NEG**

can be achieved either by adding new condition attributes or by increasing the precision of existing ones (for instance, by making the discretization procedure finer). Both solutions lead to the exponential growth in the maximum number of attribute-value combinations to be stored in the decision table [8]. It practice, it results in such negative effects as excessive size of the decision table, likely high degree of table incompleteness (in the sense of missing many combinations), weak data support for elementary sets represented in the table and, consequently, unreliable estimates of probabilities. The use of hierarchies of decision tables rather than individual tables in the process of classifier learning from data provides a partial solution to these problems [6].

Since the VPRSM boundary region $BND_{l,u}(X)$ is a definable subset of the universe U, it allows to structure the decision tables into hierarchies by treating the boundary region $BND_{l,u}(X)$ as sub-universe of U, denoted as $U' = BND_{l,u}(X)$. The "child" sub-universe U' so defined can be made completely independent from its "parent" universe U, by having its own collection of condition attributes C' to form a "child" approximation sub-space $(U, U/C')$. As on the parent level, in the approximation space $(U, U/C')$, the decision table for the subset $X' \subseteq X$ of the target decision category X, $X' = X \cap BND_{l,u}(X)$ can be derived by adapting the formula (4). By repeating this step recursively, a linear hierarchy of probabilistic decision tables can be grown until either boundary area disappears in one of the child tables, or no attributes can be identified to produce non-boundary decision table at the final level.

The nesting of approximation spaces obtained as a result of recursive computation of decision tables, as described above, creates a new approximation space on U. The resulting *hierarchical approximation space* (U, R) cannot be expressed by the indiscernibility relation, as defined in Section 2, in terms of the attributes used to form the local sub-spaces on individual levels of the hierarchy. This leads to the question: how to measure the degree of dependency between the *hierarchical partition* R of U and the partition $(X, \neg X)$ corresponding to the decision category $X \subseteq U$. Some answers to this question are explored in the next section.

5 Dependencies in Decision Table Hierarchies

There are several ways dependencies between attributes can be defined in decision tables. In Pawlak's early works functional and partial functional dependencies were explored [1]. The probabilistic generalization of the dependencies was defined and investigated in the framework of the variable precision rough set model. All these dependencies represent the relative size of the positive and negative regions of the target set X. They reflect the quality of approximation of the target category in terms of the elementary sets of the approximation space. Following the original Pawlak's terminology, we will refer to these dependencies as *Γ-dependencies.*

Other kind of dependencies, based on the notion of the certainty gain measure, reflect the average degree of change of the certainty of occurrence of the decision category X relative to its prior probability $P(X)$ [7] (see also [2] and [4]). We will refer to these dependencies as *Λ-dependencies.* The Γ-dependencies and Λ-dependencies can be extended to hierarchies of probabilistic decision tables, as described below. Because there is no single collection of attributes defining the partition of U, the dependencies of interest in this case are dependencies between the *hierarchical partition* R generated by the decision table hierarchy, forming the approximation space (U, R), and the partition $(X, \neg X)$, defined by the target set.

The partial functional dependency between attributes, referred here as *γ-dependency* $\gamma(D|C)$ measure, was introduced by Pawlak [1]. It can be expressed in terms of the probability of positive region of the partition U/D defining decision categories:

$$\gamma(D|C) = P(POS^{C,D}(U)) \tag{5}$$

where $POS^{C,D}(U)$ is a positive region of the partition U/D in the approximation space induced by the partition U/C. In the binary case of two decision categories, X and $\neg X$, the $\gamma(D|C)$-dependency can be extended to the VPRSM by defining it as the combined probability of the u-positive and l-negative regions:

$$\gamma_{l,u}(X|C) = P(POS_u(X) \cup NEG_l(X). \tag{6}$$

This dependency measure reflects the proportion of objects in U, which can be classified with sufficiently high certainty as being members, or non-members of the set X. In the case of the approximation space obtained by forming it via hierarchical classification process, the γ-dependency between the hierarchical partition R and the partition $(X, \neg X)$ can be computed directly by analyzing all classes of the hierarchical partition. However, an easier to implement recursive computation is also possible. This is done by recursively applying, starting from the leaf table of the hierarchy and going up to the root table, the following formula (7) for computing the dependency of the parent table $\gamma^U_{l,u}(X|R)$ in the hierarchical approximation space (U, R), if the dependency of a child level table $\gamma^{U'}_{l,u}(X|R')$ in the sub-approximation space (U', R') is given:

$$\gamma^U_{l,u}(X|R) = \gamma^U_{l,u}(X|C) + P(U')\gamma^{U'}_{l,u}(X|R'), \tag{7}$$

where C is collection of attributes inducing the approximation space U and $U' = BND_{l,u}(X)$. The dependency measure represents the fraction of objects that can be classified with acceptable certainty into decision categories X or

$\neg X$ by applying the decision tables in the hierarchy. The dependency of the whole structure of decision tables, that is the last dependency computed by the recursive application of formula (7), will be called a *global γ-dependency.*

Based on the probabilistic information contained in data, as given by the joint probabilities of atoms, it is also possible to evaluate the degree of probabilistic dependency between any elementary set and a decision category. The dependency measure is called *absolute certainty gain* [7] ($gabs$). It represents the degree of influence the occurrence of an elementary set E has on the likelihood of occurrence of the decision category X. The occurrence of E can increase, decrease, or have no effect on the probability of occurrence of X. The probability of occurrence of X, in the absence of any other information, is given by its prior probability $P(X)$. The degree of variation of the probability of X, due to occurrence of E, is reflected by *the absolute certainty gain function:*

$$gabs(X|E) = |P(X|E) - P(X)|, \tag{8}$$

where $| * |$ denotes absolute value function. The values of the absolute gain function fall in the range $0 \leq gabs(X|E) \leq max(P(\neg X), P(X)) < 1$. In addition, if sets X and E are independent in the probabilistic sense, that is if $P(X \cap E) = P(X)P(E)$, then $gabs(X|E) = 0$. The definition of the absolute certainty gain provides a basis for the definition of the probabilistic dependency measure between attributes. This dependency can be expressed as the average degree of change of occurrence certainty of the decision category X, or of its complement $\neg X$, due to occurrence of any elementary set [7], as defined by the *expected certainty gain* function:

$$egabs(X|C) = \sum_{E \in U/C} P(E)gabs(X|E), \tag{9}$$

where $X \in U/D$. The expected certainty gain $egabs(X|C)$ can be computed directly from joint probabilities of atoms. It can be proven [7] that the expected gain function falls in the range $0 \leq egabs(X|C) \leq 2P(X)(1 - P(X))$, where $X \in U/D$. Because the strongest dependency occurs when the decision category X is definable, i.e. when the dependency is functional, then the dependency in this deterministic case can be used as a normalization factor. The following normalized expected gain function $\lambda(X|C)$ measures the expected degree of the probabilistic dependency between elementary sets and the decision categories belonging to U/D [7]:

$$\lambda(X|C) = \frac{egabs(X|C)}{2P(X)(1 - P(X))}, \tag{10}$$

where $X \in U/D$. The dependency function reaches its maximum $\lambda(X|C) = 1$ only if the dependency is deterministic (functional). The value of the $\lambda(X|C)$ dependency function can be easily computed from the joint probabilities of atoms. As opposed to the generalized $\gamma(X|C)$ dependency, the $\lambda(X|C)$ dependency has the *monotonicity property* [3], that is, $\lambda(X|C) \leq \lambda(X|C \cup \{a\})$, where a is an extra condition attribute outside the set C. This monotonicity property allows for dependency-preserving reduction of attributes leading to the notion of probabilistic λ-reduct of attributes [3].

The λ-dependencies can be computed based on any partitioning of the universe U. In the case when the approximation space is formed through hierarchical classification, the λ-dependency between the partition R so created and the target category X can be computed via a recursive formula derived below. Let

$$egabs_{l,u}(X|C) = \sum_{E \in POS_u \cup NEG_l} P(E)gabs(X|E) \tag{11}$$

denote the conditional expected gain function, i.e. restricted to the union of positive and negative regions of the target set X in the approximations space generated by attributes C. The maximum value of $egabs_{l,u}(X|C)$, achievable in deterministic case, is $2P(X)(1 - P(X))$. Thus, the normalized *conditional λ-dependency* function, can be defined as:

$$\lambda_{l,u}(X|C) = \frac{egabs_{l,u}(X|C)}{2P(X)(1 - P(X))}. \tag{12}$$

As γ-dependencies, λ-dependencies between the target partition $(X, \neg X)$ and the hierarchical partition R can be computed recursively. The following formula (13) describes the relationship between λ-dependency computed in the approximation space (U, R), versus the dependency computed over the approximation sub-space (U, R'), where R and R' are hierarchical partitions of universes U and $U' = BND_{l,u}(X)$, respectively. Let $\lambda_{l,u}(X|R)$ and $\lambda_{l,u}(X|R')$ denote λ-dependency measures in the approximation spaces (U, R) and (U', R'), respectively. The λ-dependencies in those approximation spaces are related by the following:

$$\lambda_{l,u}(X|R) = \lambda_{l,u}(X|C) + P(BND_{l,u}(X))\lambda_{l,u}(X|R'). \tag{13}$$

The proof of the formula follows directly from the Bayes's equation. In practical terms, the formula (13) provides a practical method for efficient computation of λ-dependency in a hierarchical arrangement of probabilistic decision tables. According to this method, to compute λ-dependency for each level of the hierarchy, it suffices to compute the conditional λ-dependency and to know "child" $BND_{l,u}(X)$-level λ-dependency.

6 Concluding Remarks

Learning and evaluation of hierarchical structures of probabilistic decision tables is the main focus of the article. The earlier introduced measures of gamma and lambda dependency between attributes [7] for learned decision tables are not directly applicable to approximation spaces corresponding to hierarchical structures of decision tables. The main contribution of this work is the extension of the measures to the hierarchies and the derivation of recursive formulas for their easy computation. The gamma dependency measure allows for the assessment of the prospective ability of the classifier based on the hierarchy of decision tables to correctly predict the values of decision attribute on required level of certainty. The lambda dependency measure captures the relative degree of probabilistic correlation between classes of the partitions corresponding to condition and decision attributes, respectively. Jointly, both measures enable the user to evaluate the progress of learning with the addition of new training data and to assess the quality of the empirical classifier.

Acknowledgment. The support of the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

References

1. Pawlak, Z.: Rough sets - Theoretical Aspects of Reasoning About Data. Kluwer, Dordrecht (1991)
2. Greco, S., Matarazzo, B., Slowinski, R.: Rough membership and Bayesian confirmation measures for parametrized rough sets. In: Ślęzak, D., Wang, G., Szczuka, M., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 314–324. Springer, Heidelberg (2005)
3. Slezak, D., Ziarko, W.: The Investigation of the Bayesian rough set model. International Journal of Approximate Reasoning 40, 81–91 (2005)
4. Yao, Y.: Probabilistic approaches to rough sets. Expert Systems 20(5), 287–291 (2003)
5. Ziarko, W.: Variable precision rough sets model. Journal of Computer and Systems Sciences 46(1), 39–59 (1993)
6. Ziarko, W.: Acquisition of hierarchy-structured probabilistic decision tables and rules from data. In: Proc. of IEEE Intl. Conf. on Fuzzy Systems, Honolulu, pp. 779–784 (2002)
7. Ziarko, W.: Probabilistic rough sets. In: Ślęzak, D., Wang, G., Szczuka, M., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 283–293. Springer, Heidelberg (2005)
8. Ziarko, W.: On learnability of decision tables. In: RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 394–401. Springer, Heidelberg (2004)

Rough Sets and Vague Sets

Zbigniew Bonikowski[1] and Urszula Wybraniec-Skardowska[2]

[1] Institute of Mathematics and Informatics, University of Opole,
Oleska 48, 45-052 Opole, Poland
zbonik@math.uni.opole.pl

[2] Autonomous Section of Applied Logic,
Poznań School of Banking, Department in Chorzów, Poland
uws@uni.opole.pl

Abstract. The subject-matter of the consideration touches the problem of vagueness. The notion of the rough set, originated by Zdzisław Pawlak, was constructed under the influence of vague information and methods of shaping systems of notions leading to conceptualization and representation of vague knowledge, so also systems of their scopes as some vague sets. This paper outlines some direction of searching for a solution to this problem. In the paper, in connection to the notion of the rough set, the notion of a vague set is introduced. Some operations on these sets and their properties are discussed. The considerations intend to take into account a classical approach to reasoning, based on vague premises, and suggest finding a logic of vague sentences as a non-classical logic in which all counterparts of tautologies of classical logic are laws.

1 Introduction

Logicians and philosophers have been interested in the problem area of vague knowledge for a long time, looking for some logical bases of a theory of vague notions (terms) constituting such knowledge. Recently it has become the subject of investigations of computer scientists interested in the problems of AI, in particular, in problems of reasoning on the basis of incomplete or vague information and applications of computers to support and represent such reasoning in the computer memory. Significant results obtained by computer scientists in the scope of imprecision and vagueness: the Zadeh's fuzzy set theory [20], the Shafer's theory of evidence [17] and the Pawlak's rough sets theory [14] greatly contributed to actualization and intensification of research into vagueness.

The present paper proposes a new approach to vagueness and considers the problem of denotations of vague notions (terms) from the logical and computer sciences perspective. It yields logical foundations to a theory of vague notions (terms) and should be an essential contribution to that problem.

The paper consists of four sections. In Section 2, we introduce the notion of unit information (unit knowledge) and vague information (vague knowledge). The main notion of the vague set, inspired by the Pawlak's notion of a rough set is defined in Section 3. In Section 4 some operations on vague sets and their

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 122–132, 2007.

algebraic properties are given. A view on the problem of logic of vague concepts (terms) is discussed in Section 5. The paper ends with Section 6 including some final remarks.

2 Knowledge and Vague Knowledge

In the process of cognition of a definite fragment of reality, the cognitive agent (a man, an expert, a group of men or experts, a robot) attempts to discover information contained in it, and properly, about its objects. Each fragment of reality recognized by the agent can be understood as the following relational structure:

$$\Re = \langle \mathcal{U}, \mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n \rangle,$$

where $\mathcal{U}$, the *universe of objects of reality* $\Re$, is a nonempty set, and $\mathcal{R}_i$, for $i = 1, 2, \ldots, n$, is the set of i-ary relations on $\mathcal{U}$. One-ary relations are regarded as subsets of $\mathcal{U}$ and understood as properties of objects of $\mathcal{U}$, and multi-argument relations as relationships among its objects. Formally, every k-ary relation of $\mathcal{R}_k$ is a subset of $\mathcal{U}^k$.

We assume that reality $\Re$ is objective in relation to cognition. The objective knowledge about it consists of pieces of unit information (knowledge) about objects of $\mathcal{U}$ in relation to all particular relations of $\mathcal{R}_k$ $(k = 1, 2, \ldots, n)$.

We introduce the notion of knowledge and vague knowledge in accordance with some conceptions of the second author of this paper ([19]).

Definition 1. ***Unit information (knowledge)*** *about the object $o \in \mathcal{U}$ with respect to the relation $R \in \mathcal{R}_k$ $(k = 1, 2, \ldots, n)$ is the image $\overrightarrow{R}(o)$ of the object o with respect to the relation R*[1].

Discovering unit knowledge about objects of reality $\Re$ is realized through asking questions including certain aspects called ***attributes*** of the objects of its universe $\mathcal{U}$. Then, as the universe we usually choose a finite set $U \subseteq \mathcal{U}$ and we put it forward as generalized *attribute-value system* Σ called also an *information system* (cf. Codd [3]; Pawlak [11], [13], [14]; Marek and Pawlak [9]). Its definition is the following:

Definition 2. *Σ is an* ***information system*** *iff it is an ordered system*

$$\Sigma = \langle U, A_1, A_2, \ldots, A_n \rangle,$$

where $U \subseteq \mathcal{U}$, card(U) $< \omega$ and A_k $(k = 1, 2, \ldots, n)$ is the set of k-ary attributes understood as k-ary functions, i.e.

$$\forall_{a \in A_k} a \colon U^k \to V_a,$$

where V_a is the set of all values of the attribute a.

[1] $\overrightarrow{R}(o) = \begin{cases} R, \text{ if } o \in R, \\ \emptyset, \text{ otherwise.} \end{cases}$ for $R \in \mathcal{R}_1$.

$\overrightarrow{R}(o) = \{\langle x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k \rangle : \langle x_1, \ldots, x_{i-1}, o, x_{i+1}, \ldots, x_k \rangle \in R\}$ for $R \in \mathcal{R}_k$ $(k = 2, \ldots, n)$.

Example 1. Let us consider the following information system:

$$\mathbf{S} = \langle S, A_1, A_2 \rangle,$$

where $S = \{s_1, s_2, \ldots, s_5\}$ is a set of 5 scientists and $A_1 = \{$PUBLICATION ACTIVITY (PA), QUOTATIONS (Q)$\}$, $A_2 = \{$SCIENTIFIC COLLABORATION (SC)$\}$. The attribute PA is a function which assigns to every scientist of S a number of papers published by him. We assume that $V_{PA} = \{1, 2, \ldots, 1000\}$. The value of the attribute Q for any scientist of S is the number of quotations of his papers. We assume that $V_Q = \{0, 2, \ldots, 2000\}$. We also assume that $V_{SC} = \{0, 1, 2, 3\}$, where 0 is a value for cases, when arguments of the function SC are the same, and for any different s_n and s_m from S, 1 means that they do not collaborate, 2 means that they collaborate but they have not published any common paper, 3 means that they collaborate and have at least one common paper published.

The information system $\mathbf{S}$ can be clearly presented in the following tables:

	PA	Q
s_1	203	250
s_2	145	245
s_3	198	200
s_4	105	150
s_5	203	245

SC	s_1	s_2	s_3	s_4	s_5
s_1	0	1	3	1	3
s_2	1	0	2	3	1
s_3	3	2	0	2	2
s_4	1	3	2	0	1
s_5	3	1	2	1	0

Every attribute of the information system Σ and every value of this attribute explicitly indicate a relation belonging to the so-called ***relational system determined by*** Σ. The unit information (knowledge) about an object $o \in U$ should be considered with respect to relations of the system.

Definition 3. *$\Re(\Sigma)$ is a* ***system determined by the information system*** *Σ iff*

$$\Re(\Sigma) = \langle U, \{R_{a,W} : a \in A_1, \emptyset \neq W \subseteq V_a\}, \ldots, \{R_{a,W} : a \in A_n, \emptyset \neq W \subseteq V_a\} \rangle,$$

where $R_{a,W} = \{(o_1, o_2, \ldots, o_k) \in U^k : a((o_1, o_2, \ldots, o_k)) \in W\}$ for any $k \in \{1, 2, \ldots, n\}$, $a \in A_k$, $\emptyset \neq W \subseteq V_a$.

Let us see that $\bigcup\{R_{a,\{v\}} : a \in A_1, v \in V_a\} = U$, i.e. the family $\{R_{a,\{v\}} : a \in A_1, v \in V_a\}$ is a covering of U.

It is easy to see that

Fact 1. *The system $\Re(\Sigma)$ is uniquely determined by the system Σ.*

Example 2. Let $\mathbf{S}$ is the above given information system. Then the system determined by this system is $\Re(\mathbf{S}) = \langle U, R_{A_1}, R_{A_2} \rangle$, where $R_{A_1} = \{R_{PA,S}\}_{\emptyset \neq S \subseteq V_{PA}} \cup \{R_{Q,S}\}_{\emptyset \neq S \subseteq V_Q}$ and $R_{A_2} = \{R_{SC,S}\}_{\emptyset \neq S \subseteq V_{SC}}$.

For any attribute a of system $\mathbf{S}$ and any $i, j \in N$ we can accept the following notation:

$S_i^j = \{v \in V_a : i \leq v \leq j\}$, $S^j = \{v \in V_a : v \leq j\}$, $S_i = \{v \in V_a : i \leq v\}$.

Then, in particular, we can easily state that: $R_{PA,S_{145}^{145}} = R_{PA,\{145\}} = \{s_2\}$, $R_{PA,S_{145}^{200}} = \{s_2, s_3\}$, $R_{PA,S^{210}} = \{s_1, s_2, s_3, s_4, s_5\}$, $R_{Q,S_{150}} = \{s_1, s_2, s_3, s_4, s_5\}$, $R_{Q,S_{200}} = \{s_1, s_2, s_3, s_5\}, R_{Q,S_{245}} = \{s_1, s_2, s_5\}, R_{Q,S_{250}} = \{s_1\}$ and $R_{SC,\{2\}} = \{(s_2, s_3), (s_3, s_2), (s_3, s_4), (s_4, s_3), (s_3, s_5), (s_5, s_3)\}$, $R_{SC,\{0\}} = \{(s_i, s_i)\}_{i=1,...,5}$.

The notion of knowledge about the attributes of the system Σ depends on the cognitive agent discovering the fragment of reality Σ. According to Skowron's understanding a notion, of knowledge determined by any unary attribute (cf. Pawlak [12], Skowron et all [18], Demri, Orlowska [5] pp.16–17), we can accept the following definition of the notion of ***knowledge*** K_a ***about any k-ary attribute a***:

Definition 4. *Let Σ be the information system and $a \in A_k$ $(k = 1, 2, \ldots, n)$. Then*

(a) $K_a = \{((o_1, o_2, \ldots, o_k), V_{a,u}) : u = (o_1, o_2, \ldots, o_k) \in U^k\}$,

where $V_{a,u} \subseteq P(V_a)$, $V_{a,u}$ is the family of all sets of possible values of the attribute a for the object u from the point of view of the agent and $P(V_a)$ is the family of all subsets of V_a.

(b) *The knowledge K_a of the agent about the attribute a and its value for the object u is*

(0) ***empty*** *if card*$(V_{a,u}) = 0$*,*
(1) ***definite*** *if card*$(V_{a,u}) = 1$*,*
(> 1) ***imprecise****, in particular* ***vague****, if card*$(V_{a,u}) > 1$*.*

Let us observe that the vague knowledge about some attribute of the information system Σ is connected with assignation of a ***vague value*** to the object u.

Example 3. Let us consider again the information system **S**. The knowledge K_{PA}, K_Q, K_{SC} of the agent about the attributes from the information system **S** can be characterized by means of the following tables:

	$V_{PA,s}$	$V_{Q,s}$
s_1	$\{S_{150}^{200}, S_{170}^{220}, S^{220}\}$	$\{S_{250}, S_{300}, S_{350}, S_{400}\}$
s_2	$\{S_{100}^{150}, S_{100}^{200}, S_{150}^{180}\}$	$\{S_{200}^{250}, S_{250}^{300}, S_{200}^{300}\}$
s_3	$\{S_{150}^{160}, S_{160}^{170}, S_{170}^{180}, S_{180}^{190}, S_{190}^{200}\}$	$\{S_{150}, S_{170}, S_{190}, S_{210}, S_{230}, S_{250}, S_{300}\}$
s_4	$\{S_{105}^{105}\}$	$\{S_{200}, S_{250}^{500}, S_{400}^{800}, S_{500}\}$
s_5	$\{S_{180}^{220}, S_{200}^{240}\}$	$\{S_{200}^{250}\}$

$V_{SC,(s,s')}$	s_1	s_2	s_3	s_4	s_5
s_1	$\{\{0\}\}$	$\{\{1\}, \{2\}\}$	$\{\{3\}\}$	$\{\{1\}\}$	$\{\{3\}\}$
s_2	$\{\{1\}, \{2\}\}$	$\{\{0\}\}$	$\{\{2\}\}$	$\{\{1, 3\}\}$	$\{\{1\}\}$
s_3	$\{\{3\}\}$	$\{\{2\}\}$	$\{\{0\}\}$	$\{\{2\}\}$	$\{\{1, 3\}, \{2, 3\}\}$
s_4	$\{\{1\}\}$	$\{\{1, 3\}\}$	$\{\{2\}\}$	$\{\{0\}\}$	$\{\{1\}\}$
s_5	$\{\{3\}\}$	$\{\{1\}\}$	$\{\{1, 3\}, \{2, 3\}\}$	$\{\{1\}\}$	$\{\{0\}\}$

From Definitions 1 and 3 we get:

Fact 2. *Unit information (knowledge) about the object $o \in U$ with respect to a relation R of the system $\Re(\Sigma)$ is the image $\overrightarrow{R}(o)$ of the object o with respect to the relation R, from the point of view of the agent.*

Contrary to the objective unit knowledge $\overrightarrow{R}(o)$ about the object o of U in the reality $\Re$ with regard to its relation R, the subjective unit knowledge about the object o of U in the reality $\Re(\Sigma)$ depends on an attribute of Σ determining the relation R and its possible values from the point of view of knowledge of the agent discovering $\Re(\Sigma)$. The subjective unit knowledge $\overrightarrow{R}(o)$ from the point of view of the agent depends on his ability to solve the following equation:

$$\overrightarrow{R}(o) = x, \tag{e}$$

where x is an unknown quantity.

Solutions of (e) for k-ary relation R should be images of the object o with respect to k-ary relations $R_{a,W}$ from $\Re(\Sigma)$, where $\emptyset \neq W \in V_{a,o}$. Let us note, that for unary relation R solutions of (e) are unary relations from $\Re(\Sigma)$.

A solution of the equation (e) can be correct (then the agent's knowledge about object o is ***exact***). If the knowledge is ***inexact*** then at least one solution of (e) is not the image of the object o with respect to relation R.

Definition 5. *Unit knowledge about the object $o \in U$ in $\Re(\Sigma)$ with respect to its relation R is*

(0) ***empty*** *iff the equation (e) does not have a solution for the agent (the agent knows nothing about the value of the function $\overrightarrow{R}$ for the object o),*

(1) ***definite*** *iff the equation (e) has exactly one solution for the agent (either the agent's knowledge is exact – the agent knows the value of the function $\overrightarrow{R}$ for the object o – or he accepts only one, but not necessarily proper, value of the function),*

(> 1) ***imprecise*** *iff the equation (e) has at least two solutions for the agent (the agent allows at least two possible values of the function $\overrightarrow{R}$ for the object o).*

From Definitions 4 and 5 it follows that:

Fact 3. *The unit knowledge about the object $o \in U$ in $\Re(\Sigma)$ with respect to its relation R is*

(0) *empty if the knowledge K_a of the agent about the attribute a and its value for the object o is empty,*

(1) *definite if the knowledge K_a of the agent about the attribute a and its value for the object o is definite,*

(> 1) *imprecise if the knowledge K_a of the agent about the attribute a and its value for the object o is imprecise.*

When the unit knowledge of the agent about the object o is imprecise then most often we replace the unknown quantity x in (e) by a vague value.

Example 4. If in the system $\Re(\mathbf{S})$ we consider the relation $R = R_{Q,S_{200}}$, i.e. the set of all scientists of S that have at least 200 quotations of their papers (the property of possessing at least 200 quotations) then the unit knowledge about the scientist s_3 with respect to R can be the following vague information:

$$\overrightarrow{R}(s_3) = NUMEROUS, \qquad (e_1)$$

where $NUMEROUS$ is an unknown, indefinite, vague quantity, and the unit information about s_3 with respect to R, from the point of view of the agent, is certainly *imprecise* and *vague* if (e_1) has for him different solutions in different situations. Then the agent points to the scientist s_3 non-uniquely, possibly from his point of view different images $\overrightarrow{R}(s_3)$ of the scientist s_3 with respect to the relation R. Then the equation (e_1) has usually, for him, at least two solutions. On the basis of *Example 3* a solution of (e_1) can be each relation $R_{Q,S_{150}}, R_{Q,S_{170}}, R_{Q,S_{190}}, R_{Q,S_{210}}, R_{Q,S_{230}}$, $R_{Q,S_{250}}, R_{Q,S_{300}}$. Let us observe that $R_{Q,S_{150}} = \{s_1, s_2, s_3, s_4, s_5\}$, $R_{Q,S_{170}} = R_{Q,S_{190}} = \{s_1, s_2, s_3, s_5\}$, $R_{Q,S_{210}} = R_{Q,S_{230}} = \{s_1, s_2, s_5\}$, $R_{Q,S_{250}} = \{s_1\}$, $R_{Q,S_{300}} = \emptyset$.

3 Vague Sets and Rough Sets

In order to simplify the further considerations, we will limit ourselves to the unary relation R (property) – a subset of U.

Let $\Re(\Sigma)$ be the system determined by the information system Σ, R be its unary relation and $o \in U$.

Definition 6. *The* ***unit knowledge*** *about the object o in $\Re(\Sigma)$ with respect to R is* ***inexact*** *iff the equation (e) has the form:*

$$\overrightarrow{R}(o) = X, \qquad (ine)$$

where X is an unknown quantity from the point of view of the agent, and (ine) has for him at least one solution and at least one of the solutions is not the image $\overrightarrow{R}(o)$.

The equation (ine) can be called the *equation of inexact knowledge of the agent.* All solutions of (ine) are unary relations in the system $\Re(\Sigma)$.

Definition 7. *The* ***unit knowledge*** *about the object o in $\Re(\Sigma)$ with respect to R is* ***vague*** *iff the equation (ine) has the form:*

$$\overrightarrow{R}(o) = VAGUE, \qquad (ve)$$

where $VAGUE$ is an unknown quantity and (ve) has at least two different solutions for the agent.

The equation (ve) can be called the *equation of vague knowledge of the agent.*

Fact 4. *Vague unit knowledge is a particular case of inexact unit knowledge.*

Definition 8. *The family of all solutions (sets) of* (*ine*)*, respectively* (*ve*)*, such that at least one of them includes R, is called the* ***vague set for the object o approximated by*** *R, respectively the* ***proper vague set for the object o approximated by*** *R.*

Example 5. The family of all solutions of (e_1) from *Example 4* is a vague set $\mathbf{V}_{s_3}$ for the scientist s_3 approximated by $R_{Q,S_{200}}$ and $\mathbf{V}_{s_3} = \{R_{Q,S_{150}}, R_{Q,S_{170}}, R_{Q,S_{190}}, R_{Q,S_{210}}, R_{Q,S_{230}}, R_{Q,S_{250}}, R_{Q,S_{300}}\}$.

Vague sets, so also proper vague sets, determined by a set R are here some generalizations of sets approximated by representations (see Bonikowski [2]). They are non-empty families of unary relations from $\Re(\Sigma)$ (such that at least one of them includes R) and subfamilies of the family $P(U)$ of all subsets of the set U, determined by the set R. They have the greatest lower bound (the *lower limit*) and the least upper bound (the *upper limit*) in $P(U)$ with respect to inclusion. We will denote the greatest lower bound of any family $\mathbf{X}$ by $\underline{\mathbf{X}}$. The least upper bound of $\mathbf{X}$ will be denoted by $\overline{\mathbf{X}}$. So, we can note

Fact 5. *For each vague set $\mathbf{V}$ approximated by the set (property) R*

$$\mathbf{V} \subseteq \{Y \in P(U) : \underline{\mathbf{V}} \subseteq Y \subseteq \overline{\mathbf{V}}\}.$$

The idea of vague sets was conceived upon the idea of Pawlak's rough sets [14], who defined them by means of the operations of the *lower approximation*: $\underline{\ \ }$, and the upper approximation: $\overline{\ \ }$, defined on subsets of U. The lower approximation of a set is defined as a union of indiscernibility classes of a given relation in U^2, which are included in this set, whereas the upper approximation of a set is defined as a union of the indiscernibility classes of the relation, which have non-empty intersection with this set.

Definition 9. *A* ***rough set*** *determined by a set $R \subseteq U$ is a family $\mathbf{P}$ of all sets satisfying the following condition:*

$$\mathbf{P} = \{Y \in P(U) : \underline{Y} = \underline{R} \wedge \overline{Y} = \overline{R}\}^{2}.$$

Let us observe that because $R \subseteq R \in \mathbf{P}$, the family $\mathbf{P}$ is a non-empty family of sets such that at least one of them includes R (cf. Definition 8). By analogy to Fact 5 we have

Fact 6. *For each rough set $\mathbf{P}$ determined by the set (property) R*

$$\mathbf{P} \subseteq \{Y \in P(U) : \underline{R} \subseteq Y \subseteq \overline{R}\}.$$

It is obvious that

Fact 7. *If $\mathbf{V}$ is a vague set and $\underline{X} = \underline{\mathbf{V}}$ and $\overline{X} = \overline{\mathbf{V}}$ for any $X \in \mathbf{V}$, then $\mathbf{V}$ is a subset of a rough set determined by any set of $\mathbf{V}$.*

[2] Some authors define a rough set as a pair of sets (lower approximation, upper approximation)(cf. e.g. Iwiński [7], Pagliani [10]).

For every rough set $\mathbf{P}$ determined by R we have: $\underline{\mathbf{P}} = \underline{R}$ and $\overline{\mathbf{P}} = \overline{R}$. So we can consider the following generalization of the notion of the rough set:

Definition 10. *A non-empty family* $\mathbf{G}$ *of subsets of* U *is called a* ***generalized rough set*** *determined by a set* R *iff it satisfies the following condition:*

$$\underline{\mathbf{G}} = \underline{R} \text{ and } \overline{\mathbf{G}} = \overline{R}.$$

It is easily seen that

Fact 8. *Every rough set determined by a set* R *is a generalized rough set determined by* R.

Fact 9. *If* $\mathbf{V}$ *is a vague set and there exists a set* $X \subseteq U$ *such, that* $\underline{X} = \underline{\mathbf{V}}$ *and* $\overline{X} = \overline{\mathbf{V}}$, *then* $\mathbf{V}$ *is a generalized rough set determined by the set* X.

4 Operations on Vague Sets

Let us denote by $\mathcal{V}$ the family of all vague sets approximated by relations in system $\Re(\Sigma)$. In the family $\mathcal{V}$ we can define an operation of negation $\neg$ on vague sets, a union operation $\oplus$ and an intersection operation $\odot$ on any two vague sets.

Definition 11. *Let* $\mathbf{V_1} = \{R_i\}_{i \in I}$ *and* $\mathbf{V_2} = \{R_j\}_{j \in J}$ *be vague sets determined by sets* $R \subseteq U$ *and* $S \subseteq U$, *respectively.*

(a) $\mathbf{V_1} \oplus \mathbf{V_2} = \{R_i\}_{i \in I} \oplus \{R_j\}_{j \in J} = \{R_i \cup R_j\}_{i \in I, j \in J}$,
(b) $\mathbf{V_1} \odot \mathbf{V_2} = \{R_i\}_{i \in I} \odot \{R_j\}_{j \in J} = \{R_i \cap R_j\}_{i \in I, j \in J}$,
(c) $\neg \mathbf{V_1} = \neg \{R_i\}_{i \in I} = \{U \setminus R_i\}_{i \in I}$.

The family $\mathbf{V_1} \oplus \mathbf{V_2}$ *is called the union of vague sets* $\mathbf{V_1}$ *and* $\mathbf{V_2}$ *determined by relation* $R \cup S$, *the family* $\mathbf{V_1} \odot \mathbf{V_2}$ *is called the intersection of vague sets* $\mathbf{V_1}$ *and* $\mathbf{V_2}$ *determined by relation* $R \cap S$ *and the family* $\neg \mathbf{V_1}$ *is called the negation of vague set* $\mathbf{V_1}$ *determined by relation* $U \setminus R$.

Theorem 1. *Let* $\mathbf{V_1} = \{R_i\}_{i \in I}$ *and* $\mathbf{V_2} = \{R_j\}_{j \in J}$ *be vague sets determined by sets* R *and* S, *respectively.*

(a) $\underline{\mathbf{V_1} \oplus \mathbf{V_2}} = \underline{\mathbf{V_1}} \cup \underline{\mathbf{V_2}} = \bigcap \{R_i \cup R_j\}_{i \in I, j \in J}$ *and*
$\overline{\mathbf{V_1} \oplus \mathbf{V_2}} = \overline{\mathbf{V_1}} \cup \overline{\mathbf{V_2}} = \bigcup \{R_i \cup R_j\}_{i \in I, j \in J}$,
(b) $\underline{\mathbf{V_1} \odot \mathbf{V_2}} = \underline{\mathbf{V_1}} \cap \underline{\mathbf{V_2}} = \bigcap \{R_i \cap R_j\}_{i \in I, j \in J}$ *and*
$\overline{\mathbf{V_1} \odot \mathbf{V_2}} = \overline{\mathbf{V_1}} \cap \overline{\mathbf{V_2}} = \bigcup \{R_i \cap R_j\}_{i \in I, j \in J}$,
(c) $\underline{\neg \mathbf{V_1}} = U \setminus \overline{\mathbf{V_1}}$ *and* $\overline{\neg \mathbf{V_1}} = U \setminus \underline{\mathbf{V_1}}$.

Theorem 2. *The structure* $\mathfrak{B} = (\mathcal{V}, \oplus, \odot, \neg, \mathbf{0}, \mathbf{1})$ *is a Boolean algebra, where* $\mathbf{0} = \{\emptyset\}$ *and* $\mathbf{1} = \{U\}$.

We can easily observe that the above-defined operations on vague sets differ from Zadeh's operations on fuzzy sets, from standard operations in any field of sets and, in particular, also from operations on rough sets defined in papers of

Pomykala [16] and Bonikowski [1]. In the last cases the family of all rough sets with operations defined in these papers is Stone algebra.

5 On Logic of Vague Terms

How to solve the problem of logic of vague terms, logic of vague sentences (*vague logic*) based on the vague sets characterized in the previous sections? An answer to this question requires describing briefly the problem of language representation of unit knowledge.

On the basis of our examples let us consider two pieces of unit information about the scientist s_3, with respect to the set R of all scientists that have at least 200 quotations of their papers:

first, exact unit knowledge

$$\overrightarrow{R}(s_3) = \{s_1, s_2, s_3, s_5\}, \tag{ee}$$

next, vague unit knowledge:

$$\overrightarrow{R}(s_3) = \mathit{NUMEROUS}. \tag{e_1}$$

Let s_3 be the designator of the proper name a, R – denotation (extension) of the name-predicate P (‘*a scientist who has at least 200 quotations of his papers*’) and the vague name-predicate V (‘*a scientist who has numerous quotations of his papers*’) be a language representation of the vague quantity *NUMEROUS*. Then a representation of the first equation (ee) is the logical atomic sentence

$$a \text{ is } P \tag{re}$$

and a representation of the second equation (e_1) is the vague sentence

$$a \text{ is } V. \tag{re_1}$$

In an equivalent way we can represent, respectively, (ee) and (e_1) by means of a logical atomic sentence:

$$aP \text{ or } P(a), \tag{re'}$$

where P is the predicate (*‘has at least 200 quotations of his papers’*) and by means of a vague sentence

$$aV \text{ or } V(a), \tag{re'_1}$$

where V is the vague predicate (*‘has numerous quotations of his papers’*).

The sentence (re_1) (res. the sentence (re'_1)) is not a logical sentence, but it can be treated as a *sentential form*, which represents all logical sentences, in particular the sentence (re) (respectively sentence (re')) that arises by replacing the vague name-predicate (res. vague predicate) V by allowable sharp name-predicates (res. sharp predicates), whose denotations (extensions) constitute the vague set $\mathbf{V}_{s_3}$ that is the denotation of V and simultaneously the set of solutions the equation (e_1) from the agent's point of view.

By analogy we can consider every atomic vague sentence with the form $V(a)$, where a is an individual term and V — its vague predicate, as a *sentential form with V as a vague variable*, run over all denotations of sharp predicates that can be substituted for V in order to get precise, true or false, logical sentences from the form $V(a)$. Then, the scope of the variable V is the vague set $\mathbf{V}_o$ determined by the designator o of the term a.

All the above remarks lead to a 'conservative', classical approach in searching for logic of vague terms or vague sentences, called here *vague logic* (cf. Fine [6], Cresswell [4]). It is easy to see that all counterparts of laws of classical logic are laws of *vague logic*, even if for the fact that vague sentences have an interpretation in Boolean algebra $\mathfrak{B}$ of vague sets (see Theorem 2).

It should be noticed that sentential connectives for *vague logic* should not satisfy *standard conditions* (see Malinowski [8]). For example, an alternative of two vague sentences $V(a)$ and $V(b)$ can be a 'true' vague sentence (sentential form) despite the fact that its arguments $V(a)$ and $V(b)$ are neither 'true' or 'false' sentential form, i.e. they represent in certain cases true and in other cases false sentences. It is not contrary to the statement that all vague sentential forms which we obtain by suitable substitution of sentential variables (resp. predicate variables) by vague sentences (resp. vague predicates) in laws of classical logic always represent true sentences. Thus they are laws of vague logic.

6 Final Remarks

1. The concept of vagueness was defined here as a certain indefinite, vague quantity or property corresponding to the agent knowledge discovering a fragment of reality. It was given by means of the *equation of inexact knowledge of the agent*. A vague set was defined as a set (a family) of all possible solutions (sets) of this equation and although our considerations were limited to the case of unary relations, they can easily be generalized to the cases of any k-ary relations.
2. The idea of *vague sets* was taken here from the idea of rough sets originating from Zdzisław Pawlak, because Pawlak's theory of rough sets takes a non-numerical, qualitative approach, to the issue vagueness, as opposed to the quantitative characteristics of vagueness phenomenon by Lotfi Zadeh.
3. Vague sets, like rough sets, are based on the idea of a set approximation by two sets called the lower and the upper limits of this set. These two kinds of sets are families of sets approximated by suitable limits.
4. Pawlak's approach and the approach discussed in this paper are connected with a reference to the concept of a cognitive agent's knowledge about the objects of the investigated reality (see Pawlak [15]) This knowledge is determined by the system of concepts, that is determined by a system of their extensions (denotations). When the concept is vague, its denotation, in Pawlak's sense, is a rough set, while in the authors' sense – a vague set which at some conditions is a subset of the rough set.
5. In language representation the *equation of inexact, vague knowledge of the agent* can be expressed by means of vague sentences containing a vague predicate. Its denotation (extension) is a family of all scopes of sharp predicates which can be substituted for the vague predicate from the point of view of the agent. The denotation is simultaneously the vague set of all solutions of the equation of the vague agent's knowledge.

6. Because vague sentences can be treated as sentential forms in which variables are vague predicates, all counterparts of tautologies of classical logic are laws of *vague logic* (logic of vague sentences).
7. *Vague logic* is based on classical logic but it is many-valued logic, because its sentential connectives are intensional.

References

1. Bonikowski, Z.: A Certain Copnception of the Calculus of Rough Sets. Notre Dame J. Formal Logic 33, 412–421 (1992)
2. Bonikowski, Z.: Sets Approximated by Representations. In: Polish, the doctoral dissertation prepared under the supervision of Prof. U.Wybraniec-Skardowska, Warszawa (1996)
3. Codd, E.F.: A Relational Model of Data for Large Shared Data Banks. Comm. ACM 13, 377–387 (1970)
4. Cresswell, M.J.: Logics and Languages. Methuen, London (1973)
5. Demri, S., Orłowska, E.: Incomplete Information: Structure, Inference, Complexity. Springer, Berlin (2002)
6. Fine, K.: Vagueness, Truth and Logic. Synthese 30, 265–300 (1975)
7. Iwiński, T.: Algebraic Approach to Rough Sets. Bull. Pol. Acad. Sci. Math. 35, 673–683 (1987)
8. Malinowski, G.: Many-Valued Logics. Oxford University Press, Oxford (1993)
9. Marek, W., Pawlak, Z.: Rough Sets and Information Systems. ICS PAS Report 441 (1981)
10. Pagliani, P.: Rough Set Theory and Logic-Algebraic Structures. In: Orłowska, E. (ed.) Incomplete Information: Rough Set Analysis, pp. 109–190. Physica Verlag, Heidelberg (1998)
11. Pawlak, Z.: Information Systems. ICS PAS Report 338 (1979)
12. Pawlak, Z.: Information Systems – Theoretical Foundations (in Polish). PWN – Polish Scientific Publishers, Warsaw (1981)
13. Pawlak, Z.: Information Systems – Theoretical Foundations. Information Systems 6, 205–218 (1981)
14. Pawlak, Z.: Rough Sets. Intern. J. Comp. Inform. Sci. 11, 341–356 (1982)
15. Pawlak, Z.: Rough Sets. In: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht (1991)
16. Pomykała, J., Pomykała, J.A.: The Stone Algebra of Rough Sets. Bull. Pol. Acad. Sci. Math. 36, 495–508 (1988)
17. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
18. Skowron, A., Komorowski, J., Pawlak, Z., Polkowski, L.: Rough Sets Perspective on Data and Knowledge. In: Klösgen, W., Żytkow, J.M. (eds.) Handbook of Data Mining and Knowlewdge Discovery, pp. 134–149. Oxford University Press, Oxford (2002)
19. Wybraniec-Skardowska, U.: Knowledge, Vagueness and Logic. Int. J. Appl. Math. Comput. Sci. 11, 719–737 (2001)
20. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)

Consistency-Degree Between Knowledges[*]

M.K. Chakraborty[1] and P. Samanta[2]

[1] Department of Pure Mathematics, University of Calcutta
35, Ballygunge Circular Road, Kolkata-700019, India
mihirc99@vsnl.com
[2] Department of Mathematics
Katwa College, Katwa, Burdwan, West Bengal, India
pulak_samanta06@yahoo.co.in

Abstract. A knowledge is considered to be a partition of the Universe into classes of objects indiscernible with respect to the information available. So, knowledge of two agents may differ. In this paper a consistency-measure between knowledges is obtained. It is shown that the complement of consistency-measure, viz. the inconsistency-measure, is a metric under certain restrictions. Some initial axioms of a logic of consistency are proposed.

Keywords: Rough Sets, elementary category, knowledge, dependency degree, consistency degree.

1 Introduction

In Pawlak's book 'Rough Sets' [cf. [7]] there is a chapter entitled 'Dependencies in Knowledge Base'. He has described dependency of knowledge in the following manner : a knowledge Q depends on a knowledge P if and only if all the elementary categories of Q can be defined in terms of some elementary categories of P i.e the indiscernibility relation generated by P is finer than that of Q, in other words, the granules or equivalence classes formed by Q are split further into smaller granules formed by P. Two knowledges are equivalent if they generate the same granules and are independent if neither P nor Q is dependent on the other. He took a next step forward - a quite natural one - to define partial dependency of knowledge which is a graded notion and was first defined by Novotný and Pawlak in [4]. The formal definition of partial dependency in terms of dependency degree of knowledges is given later in definition [4]. The basic idea is, however, to give a kind of natural measure(estimate) of one knowledge being dependent on another when not all the equivalence classes of one could be obtained as unions of equivalence classes of the other.

Based upon the notion of partial dependency, we, in this paper propose a concept of partial consistency of knowledges. This is also a graded concept that

[*] The second author acknowledges the financial support from the University Grants Commission, Government of India.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 133–141, 2007.

reduces to the concept of ordinary, crisp consistency or inconsistency. By a knowledge we shall not mean a set P of equivalence relations on an Universe U, as understood by Pawlak but the intersection $\bigcap P$ or $IND(P)$, the indiscernibility relation caused by P. This intersection is also an equivalence relation. So, for our purpose (and also of Pawlak's) it is enough to call an equivalence relation on an Universe a knowledge about that Universe - and that reduces to the partition caused by the relation. The Universe is taken to be finite and finiteness is essential for the concepts in this paper. If knowledge is ultimately characterized by equivalence classes / elementary categories / granules caused by it in the universe of discourse, two knowledges are consistent if and only if both of them generate the same granules. We shall be justified in calling two such knowledges 'fully consistent' instead of consistent. On the other extreme there are two knowledges that may be called 'fully inconsistent' where no equivalence class due to one is contained in that due to the other. In other words, for each object 'x', not all objects indiscernible with it with respect to one knowledge are indiscernible with it with respect to the other. Thus the elementary categories are totally disparate. In the former situation it is quite justified to say that the consistency degree of the two knowledges is 1 and in the latter case it is 0. While the former is the case which according to Pawlak constitute 'equivalent knowledges', the latter case is not what Pawlak calls 'independent knowledges'. Being fully inconsistent is a more demanding concept than what has been said to be 'independent' by him. It is quite expected that some kind of a measure of consistency in the case of intermediate situations should emerge. It is also natural that this measure should be composed of the dependency measures proposed by Pawlak since we have noticed that knowledges P and Q turn out to be fully consistent if and only if P depends on Q and Q depends on P. In the current context the sentence 'P depends on Q' being a sentence that admits values(truth) other than 0(false) and 1(true), in order to define consistency, a conjunction operation on the value set [0,1] is sought for. A natural candidate could be a t-norm [cf. [2]] but we shall see later that such an operation does not serve the purpose. So another binary operator C on [0.1] has been proposed. The connection between C and t-norms shall be discussed in some of the propositions that follow.

It would be appropriate to look into the notions of consistency that have appeared in rough-set literature. In [3] a kind of [0,1]-valued functions assigning values called 'consistency degrees' to profiles (i.e finite multi-sets on an universe U) is defined satisfying certain consistency postulates. This definition presupposes an underlying 'distance' function on U. These consistency measuring functions are utilized to deal with conflict situations. As it appears and stands now, there is hardly any link with rough set theory and this idea of consistency degree.

On the other hand, in [8] a notion of the consistency of an object x with another object y in an information system involving condition attributes CON and decision attributes DEC is defined. x is said to be consistent with y iff whenever x and y are indiscernible w.r.t all the CON attributes they shall also be so w.r.t all the DEC attributes. x is called consistent if and only if it is consistent with every y. This means that either x is not indiscernible with any other element by the condition attributes or if there is any such y, the objects x,y shall be indiscernible by the decision attributes also. In terms of consistent pairs of elements, the authors have defined a dependency degree viz. $deg(CON, DEC)$ among attributes of the information system.

2 Dependency of Knowledge

We would accept the basic philosophy that a knowledge of an agent about an universe is her ability to categorize objects inhabiting it through information received from various sources or perception in the form of attribute-value data. For our purpose it is enough to start with the indiscernibility relation caused by the attribute-value system. So, knowledge is defined as follows.

Definition 1. *Knowledge : A knowledge is a pair, $< U, P >$ where U is a non-empty finite set and P is an equivalence relation on U. P will also denote the partition generated by the equivalence relation.*

Definition 2. *Finer and Coarser Knowledge : A knowledge P is said to be finer than the knowledge Q if every block of the partition P is included in some block of the partition Q. In such a case Q is said to coarser than P. We shall write it as $P \preceq Q$.*

We recall a few notions due to Pawlak (and others) e.g P-positive region of Q and based upon it dependency-degree of knowledges.

Definition 3. *Let P and Q be two equivalence relations over U. The P-positive region of Q, denoted by $Pos_P(Q)$ is defined by $Pos_P(Q) = \bigcup_{X \in U/Q} \underline{P}X$, where $\underline{P}X = \{\bigcup Y \in U/P : Y \subseteq X\}$ called P-lower approximation of X.*

Definition 4. *Dependency degree : Knowledge Q depends in a degree k $(0 \leq k \leq 1)$ on knowledge P , written as $P \Rightarrow_k Q$, iff $k = \frac{CardPos_P(Q)}{CardU}$ where card denotes cardinality of the set.*
If k = 1 , we say that Q totally depends on P and we write $P \Rightarrow Q$; and if k = 0 we say that Q is totally independent of P.

Viewing from the angle of multi-valuedness one can say that the sentence 'The knowledge Q depends on the knowledge P' instead of being only 'true'(1) or 'false'(0) may receive other intermediate truth-values, the value k being determined as above. This approach justifies the term 'partial dependency' as well.

In propositions 1,2 and 3, we enlist some elementary, often trivial, properties of dependency degree some of them being newly exercised but most of which are present in [4,7]. Some of these properties e.g. proposition 3(v) will constitute the basis of definitions and results of the next section.

Proposition 1. *(i)* $[x]_{P_1 \cap P_2} = [x]_{P_1} \cap [x]_{P_2}$,
(ii) $P \Rightarrow Q$ *and* $P \preceq R$, *then* $R \Rightarrow Q$,

(iii)If $P \Rightarrow Q$ and $R \preceq Q$ then $P \Rightarrow R$,
(iv)If $P \Rightarrow Q$ and $Q \Rightarrow R$ then $P \Rightarrow R$,
(v)If $P \Rightarrow R$ and $Q \Rightarrow R$ then $P \cap Q \Rightarrow R$,
(vi) If $P \Rightarrow R \cap Q$ then $P \Rightarrow R$ and $P \Rightarrow Q$,
(vii) If $P \Rightarrow Q$ and $Q \cap R \Rightarrow T$ then $P \cap R \Rightarrow T$,
(viii) If $P \Rightarrow Q$ and $R \Rightarrow T$ then $P \cap R \Rightarrow Q \cap T$.

Proposition 2. *(i) If $P\prime \preceq P$ then $\underline{P\prime}X \supseteq \underline{P}X$,*
(ii) If $P \Rightarrow_a Q$ and $P\prime \preceq P$ then $P\prime \Rightarrow_b Q$ where $b \geq a$,
(iii) $P \Rightarrow_a Q$ and $P \preceq P\prime$ then $P\prime \Rightarrow_b Q$ where $b \leq a$,
(iv) $P \Rightarrow_a Q$ and $Q\prime \preceq Q$ then $\Rightarrow_b Q\prime$ where $b \leq a$,
(v) $P \Rightarrow_a Q$ and $Q \preceq Q\prime$ then $P \Rightarrow_b Q\prime$ where $a \leq b$.

Proposition 3. *(i) If $R \Rightarrow_a P$ and $Q \Rightarrow_b P$ then $R \cap Q \Rightarrow_c P$ for some $c \geq$ Max(a,b),*
(ii) If $R \cap P \Rightarrow_a Q$ then $R \Rightarrow_b Q$ and $P \Rightarrow_c Q$ for some b,c $\leq$ a,
(iii) If $R \Rightarrow_a Q$ and $R \Rightarrow_b P$ then $R \Rightarrow_c Q \cap P$ for some c $\leq$ Min(a,b),
(iv) If $R \Rightarrow_a Q \cap P$, then $R \Rightarrow_b Q$ and $R \Rightarrow_c P$, for some b,c $\geq$ a,
(v) If $R \Rightarrow_a P$ and $P \Rightarrow_b Q$ then $R \Rightarrow_c Q$ for some c $\geq$ a+b-1.

3 Consistency of Knowledge

Two knowledges P and Q on U may be considered as fully consistent if and only if $U/P = U/Q$, that is P,Q generate exactly the same granules. This is equivalent to $P \Rightarrow Q$ and $Q \Rightarrow P$. So, a natural measure of consistency degree of P and Q might be the truth-value of the non-classical sentence "Q depends on $P \wedge P$ depends on Q" computed by a suitable conjunction operator applied on the truth-values of the two component sentences Thus a binary predicate *Cons* may be created such that $Cons(P, Q)$ will stand for the above conjunctive sentence and a triangular norm (or t-norm) used in fuzzy-literature and many-valued logic scenario is a potential candidate for computing $\wedge$. A t-norm is a mapping $t : [0, 1] \rightarrow [0, 1]$ satisfying (i) t(a,1) = a, (ii) b $\leq$ d implies t(a,b) $\leq$ t(a,d), (iii) t(a,b) = t(b,a), (iv) t(a,t(b,d)) = t(t(a,b),d). It follows that t(a,0)=0. Typical examples of t-norm are :
min(a,b) (Gödel),
max(0,a+b-1) (Lukasicwicz),
$a \times b$ (Godo,Hajek).
These are conjunction operators used extensively and are in some sense the basic t-norms [cf. [1]]. With $1 - x$ as negation operator the De-Morgan dual of t-norms called s-norms are obtained as $s(a, b) = 1 - t(1 - a, 1 - b)$. Values of disjunctive sentences are computed by s-norms.

There is however a difficulty in using a t-norm in this context. We would like to have the following assumptions to hold.

Assumption 1. Knowledges P,Q shall be fully consistent iff they generate the same partition.

Assumption 2. Knowledges P,Q shall be fully inconsistent iff no granule generated by one is contained in any granule generated by the other.

The translation of the above demands in mathematical terms is that the conjunction operator C should fulfill the conditions :
$C(a,b) = 1$ iff $a = 1, b = 1$
and $C(a,b) = 0$ iff $a = 0, b = 0$.

No t-norm satisfies the second. So we define consistency degree as follows:

Definition 5. *Let P and Q be two knowledges such that $P \Rightarrow_a Q$ and $Q \Rightarrow_b P$. The consistency degree between the two knowledges denoted by $Cons(P,Q)$ is given by $Cons(P,Q) = \frac{a+b+nab}{n+2}$, where n is a non negative integer.*

Definition 6. *Two knowledges P and Q are said to be fully consistent if $Cons(P,Q) = 1$. Two knowledge P and Q are said to be fully inconsistent if $Cons(P,Q) = 0$.*

Example 1. (i) Let $U = \{1,2,3,4,5,6,7,8\}$ and the partitions be taken as $P = \{\{1,3,5\},\{2,4,6\},$
$\{7,8\}\}$ and $Q = \{\{1,2,7\},\{3,4,8\},\{5,6\}\}$. Then $P \Rightarrow_0 Q$ and $Q \Rightarrow_0 P$. So, $Cons(P,Q) = 0$.
(ii) Let $U = \{1,2,3,4,5,6,7,8\}$ and partitions $P = \{\{1,3,5\},\{2,4,6\},\{7,8\}\}$ and $Q = \{\{1,3,5\},\{2,4,6\},\{7,8\}\}$. Then $P \Rightarrow_1 Q$ and $Q \Rightarrow_1 P$. So, $Cons(P,Q)$ $= 1$.
(iii) Let $U = \{1,2,3,4,5,6,7,8\}$ and partitions $P = \{\{1,4,5\},\{2,8\},\{6,7\},\{3\}\}$ and $Q = \{\{1,3,5\},\{2,4,7,8\},\{6\}\}$. Then $P \Rightarrow_{\frac{3}{8}} Q$ and $Q \Rightarrow_{\frac{1}{8}} P$. So, $Cons(P,Q)$ $= \frac{\frac{3}{8}+\frac{1}{8}+n\frac{3}{8}\frac{1}{8}}{n+2}$, where n is a non-negative integer.

Although any choice of n satisfies the initial requirements, some special values for it may be of special significance e.g $n = 0$, $n = Card(U)$ and n as defined in proposition 5. We shall make discussions on two of such values latter. 'n' shall be referred to as a 'consistency constant' or simply 'constant' in the sequel. The constant is a kind of constraint on consistency measure as shown in the next proposition.

Proposition 4. *For two knowledges P and Q if $n_1 \leq n_2$ then $Cons_1(P,Q) \geq Cons_2(P,Q)$ where $Cons_i(P,Q)$ is the consistency degree when n_i is the constant taken.*
Proof : Let $P \Rightarrow_a Q$ and $Q \Rightarrow_b P$. Since $n_1 \leq n_2$, so, $n_2 - n_1 \geq 0$. So $Cons_1(P,Q)$ $= \frac{a+b+n_1ab}{n_1+2}$ and $Cons_2(P,Q) = \frac{a+b+n_2ab}{n_2+2}$. Now, $\frac{a+b+n_1ab}{n_1+2} - \frac{a+b+n_2ab}{n_2+2} = \frac{(n_2-n_1)(a+b-2ab)}{(n_1+2)(n_2+2)} \geq 0$ iff $(n_2 - n_1)(a+b-2ab) \geq 0$ iff (a+b-2ab) ≥ 0 iff a+b $\geq$ 2ab. Now, $\frac{a+b}{2} \geq \sqrt{ab} \geq ab$. So a+b $\geq$ 2ab holds. This shows that $Cons_1(P,Q) \geq Cons_2(P,Q)$.

Proposition 5. *If n = the number of elements a$\in$ U such that $[a]_P \nsubseteq [a]_Q$ and $[a]_Q \nsubseteq [a]_P$, then n = Card U - [Card $\bigcup_{X \in U/Q} \underline{P}X$. + Card $\bigcup_{X \in U/P} \underline{Q}X$ - Card($\bigcup_{X \in U/Q} \underline{P}X \bigcap \bigcup_{X \in U/P} \underline{Q}X$)].*

Proof: Here the number of elements a$\in$ *U such that* $[a]_P \subseteq [a]_Q = Card \bigcup_{X\in U/Q} \underline{P}X \ldots (i)$*. Now the number of elements a*$\in$ *U such that* $[a]_Q \subseteq [a]_P = Card \bigcup_{X\in U/P} \underline{Q}X \ldots (ii)$*. So the number of elements common to (i) and (ii)* $= Card$ $(\bigcup_{X\in U/Q} \underline{P}X \bigcap \bigcup_{X\in U/P} \underline{Q}X)]$ *...(iii) . From (i), (ii) and (iii) the proposition follows.*

One can observe that the definition of a consistent object in [3](cf. Introduction) may be generalized relative to any pair (P, Q) of partitions of the Universe, not only restricted to the partitions caused due to the pair (CON, DEC). With this extension of the notion, n is the count of all those objects a such that a is not consistent relative to both the pairs (P, Q) and (Q, P). In the following examples n is taken to be this number.

Example 2. (i) Let U=$\{1, 2, 3, 4, 5, 6, 7, 8\}$ and partitions $P = \{\{1, 3, 5\}, \{2, 4, 6\}, \{7, 8\}\}$ and $Q = \{\{1, 2, 7\}, \{3, 4, 8\}, \{5, 6\}\}$. Then $P \Rightarrow_0 Q$ and $Q \Rightarrow_0 P$. Here n=8. So, $Cons(P, Q) = \frac{0+0+8.0.0}{8+2} = 0$.
(ii) Let U $= \{1, 2, 3, 4, 5, 6, 7, 8\}$ and partitions $P = \{\{1, 3, 5\}, \{2, 4, 6\}, \{7, 8\}\}$ and $Q = \{\{1, 3, 5\}, \{2, 4, 6\}, \{7, 8\}\}$. Then $P \Rightarrow_1 Q$ and $Q \Rightarrow_1 P$. Here n=0. So, $Cons(P, Q) = \frac{1+1+0.1.1}{0+2} = 1$.
(iii) Let U $= \{1, 2, 3, 4, 5, 6, 7, 8\}$ and partitions $P = \{\{1, 4, 5\}, \{2, 8\}, \{6, 7\}, \{3\}\}$ and $Q = \{\{1, 3, 5\}, \{2, 4, 7, 8\}, \{6\}\}$. Then $P \Rightarrow_{\frac{3}{8}} Q$ and $Q \Rightarrow_{\frac{1}{8}} P$. Here n=4. So, $Cons(P, Q) = \frac{\frac{3}{8}+\frac{1}{8}+4.\frac{3}{8}.\frac{1}{8}}{4+2} = \frac{11}{96}$.

If the t-norm is taken to be max(0,a+b-1), then the corresponding s-norm is min(1,a+b). For the t-norm min(a,b), the s-norm is max(a,b). There is an order relation in the t-norms/ s-norms, viz.

any t-norm $\leqq$ min $\leqq$ max $\leqq$ any s-norm.

In particular

max(o,a+b-1) $\leqq$ min(a,b) $\leqq$ max(a,b) $\leqq$ min(1,a+b).

Where does the Cons function situate itself in this chain - might be an interesting and useful query. The following proposition answers this question.

Proposition 6. *$max(0, a + b - 1) \leqq Cons(P, Q) \leqq max(a, b)$ if $P \Rightarrow_a Q$ and $Q \Rightarrow_b P$.*

To compare $Cons(P, Q)$ and $min(a, b)$, we have,

Proposition 7. *Let P and Q be two knowledges and $P \Rightarrow_a Q$ and $Q \Rightarrow_b P$. Then (i) a=b=1 iff min(a,b)=$Cons(P, Q)$=1,*
(ii) If either a=1 or b=1 then min(a,b) $\leq Cons(P, Q)$,
(iii) min(a,b)= a $\leq Cons(P, Q)$ iff $n \leq \frac{a-b}{a(b-1)}, a \neq 0, b \neq 1$,
(iv) min(a,b)= a $\geq Cons(P, Q)$ iff $n \geq \frac{a-b}{a(b-1)}, a \neq 0, b \neq 1$,
(v) max(0,a+b-1) $\leq Cons(P, Q) \leq$ max(a,b) $\leq$ s(a,b)=min(1,a+b).

The Cons function seems to be quite similar to a t-norm but not the same. So a closer look into the function is worthwhile.

We define a function C:[0,1]×[0,1]→[0,1] as follows $C(a, b) = \frac{a+b+nab}{n+2}$ where n is a non-negative integer.

Proposition 8. *(i)* $0 \leq C(a,b) \leq 1$,
(ii) If $a \leq b$ *then* $C(a,b) \leq C(a,c)$,
(iii) $C(a,b) = C(b,a)$,
(iv) $C(a,C(b,c)) = C(C(a,b),c)$ *iff a=c ;*
$C(a,C(b,c)) \leq C(C(a,b),c)$ *iff* $a \leq c$;
$C(a,C(b,c)) \geq C(C(a,b),c)$ *iff* $a \geq c$,
(v) $C(a,1) \geq a$, *equality occurring iff a=1,*
(vi) $C(a,0) \leq a$, *equality occurring iff a=0,*
(vii) $C(a,b) = 1$ *iff a=b=1 and* $C(a,b) = 0$ *iff a=b=0,*
(viii) $C(a,a) = a$ *iff either a=0 or a=1,*

Definition 7. *We define* $1 - C(1-a, 1-b) = D(a,b)$.

Proposition 9. *(i)*$C(a,b) \leq$ *1-*$C(1-a,1-b)$= $D(a,b)$
(ii) $0 \leq D(a,b) \leq 1$,
(iii) If $a \leq b$ *then* $D(a,b) \leq D(a,c)$,
(iv) $D(a,b) = D(b,a)$,
(v) $D(a,D(b,c)) = D(D(a,b),c)$ *iff a=c ,*
$D(a,D(b,c)) \leq D(D(a,b),c)$ *iff* $a \leq c$,
$D(a,D(b,c)) \geq D(D(a,b),c)$ *iff* $a \geq c$,
(vi) $D(a,1) \geq a$, *equality occurring iff a=1,*
(vii) $D(a,0) \leq a$, *equality occurring iff a=0,*
(viii) $D(a,b) = 1$ *iff a=b=1 and* $D(a,b) = 0$ *iff a=b=0,*
(ix) $D(a,a) = a$ *iff either a=0 or a=1,*
(x) $D(a,b) \leq \min(1,a+b)$.

One can immediately observe similarity between the function D and Lukasiewicz s-norm.

The consistency function Cons gives a measure of similarity between two knowledges. It would be natural to define a measure of inconsistency or dissimilarity now. In [4] a notion of distance is available.

Definition 8. *If* $P \Rightarrow_a Q$ *and* $Q \Rightarrow_b P$ *then the distance function is denoted by* $\rho(P,Q)$ *and defined as* $\rho(P,Q)=\frac{2-(a+b)}{2}$.

Proposition 10. *The distance function* ρ *satisfies the conditions :*
(i) $o \leq \rho(P,Q) \leq 1$
(ii) $\rho(P,P) = 0$
(iii) $\rho(P,Q) = \rho(P,Q)$
(iv) $\rho(P,R) \leq \rho(P,Q) + \rho(Q,R)$.
For proof the reader is referred to [4].

Definition 9. *We now define a measure of inconsistency by :*
$InCons(P,Q)$ = *1-*$Cons(P,Q)$

Proposition 11. *(i)* $o \leq InCons(P,Q) \leq 1$,
(ii) $InCons(P,P) = 0$,
(iii) $InCons(P,Q) = InCons(P,Q)$,

(iv) $InCons(P,R) \leq InCons(P,Q) + InCons(Q,R)$ *for a fixed constant* n.
Proof of (iv): Let $P \Rightarrow_x R$, $R \Rightarrow_y P$, $P \Rightarrow_a Q$, $Q \Rightarrow_b P$, $Q \Rightarrow_l R$, $R \Rightarrow_m Q$ *...(i). Now* $InCons(P,R) = \frac{n+2-x-y-nxy}{n+2} \leq InCons(P,Q) + InCons(Q,R) = \frac{n+2-a-b-nab}{n+2} + \frac{n+2-l-m-nlm}{n+2} = \frac{2(n+2)-n(ab+lm)-(a+b+l+m)}{n+2}$
iff $n+2-x-y-nxy \leq 2(n+2)-n(ab+lm)-(a+b+l+m)$
iff $n(ab+lm-xy-1) \leq 2+x+y-(a+b+l+m)$*...(ii).*
From (i) by Proposition 3(v) we have $x \geq (a+m-1)$ *and* $y \geq (b+l-1)$.
Hence $(ab+lm-xy-1) \leq (ab+lm-(a+m-1)(b+l-1)-1)$
$= (a(1-l)+b(1-m)+(m-1)+(l-1))$
$\leq (1-l+1-m+m-1+l-1)(\because 0 \leq a,b \leq 1)$ *=0. ...(iii)*
Now, $2+x+y-(a+b+l+m) = 2(\frac{2-a-b}{2} + \frac{2-l-m}{2} - \frac{2-x-y}{2})$
$= 2(\rho(P,Q)+\rho(Q,R)-\rho(P,R)) \geq 0$. *...(iv)[by Proposition 10(iv)]*
Thus the left hand side of inequality (ii) is negative and the right hand side of (ii) is positive. So (iv) i.e triangle inequality is established.

Proposition 11 shows that for any fixed n the inconsistency measure of knowledge is a metric. It is also a generalization of the distance function ρ in [4]; InCons reduces to ρ when $n = 0$. n is again a kind of constraint on the inconsistency measure - as n increases, the inconsistency increases too.

4 Towards a Logic of Consistency of Knowledge

We are now at the threshold of a logic of consistency (of knowledge). Along with the usual propositional connectives the language shall contain two binary predicates, '*Cons*' and '*Dep*' for consistency and dependency respectively. At least the following features of this logic are present.

(i) $0 \leq Cons(P,Q) \leq 1$,
(ii) $Cons(P,P) = 1$,
(iii) $Cons(P,Q) = Cons(Q,P)$,
(iv) $Cons(P,Q) = 0$ iff $Dep(P,Q) = 0$ and $Dep(Q,P) = 0$
and $Cons(P,Q) = 1$ iff $Dep(P,Q) = 1$ and $Dep(Q,P) = 1$
(v) $Cons(P,Q)$ and $Cons(Q,R)$ implies $Cons(P,R)$.

(i) shows that the logic is many-valued; (ii) and (ii i) are natural expectations; (iv) conforms to assumptions 1 and 2 (section2); (v) shows transitivity the predicate Cons.

All these may be considered as axioms of a possible logic. That the transitivity holds is shown below. We want to show that $Cons(P,Q)$ and $Cons(Q,R)$ implies $Cons(P,R)$ i.e, $Cons(P,Q)$ and $Cons(Q,R) \leq Cons(P,R)$. We use Lukasiewicz t-norm to compute 'and'. Let n be the fixed constant. So,what is needed is Max(0,$Cons(P,Q) + Cons(Q,R) - 1) \leq Cons(P,R)$. Clearly, $Cons(P,R) \geq 0$ ...(i). We shall now show $Cons(P,R) \geq Cons(P,Q) + Cons(Q,R) - 1$. Let $P \Rightarrow_x R$, $R \Rightarrow_y P$, $P \Rightarrow_a Q$, $Q \Rightarrow_b P$, $Q \Rightarrow_l R$, $R \Rightarrow_m Q$ So $x \geq (a+m-1)$ and $y \geq (b+l-1)$ [cf. Proposition 3(v)]...(ii). So, Cons(P,Q)+Cons(Q,R)-1 = $\frac{a+b+nab}{n+2} + \frac{l+m+nlm}{n+2} - 1 = \frac{(a+l-1)+(b+m-1)+n(ab+lm-1)}{n+2} \leq \frac{x+y+n(ab+lm-1)}{n+2}$ [using (ii)]...(iii). Here, xy $\geq (a+l-1)(b+m-1) =$ ab+lm+(m-1)(a-1)+(b-1)(l-1)-1 $\geq$ ab+lm-1. [as, m-1 ≤ 0 , a-1 ≤ 0 , so (m-1)(a-1) ≥ 0 , and b-1 ≤ 0 , l-1 ≤ 0 , (b-1)(l-1) ≥ 0] ...(iv) . So (iii) and (iv) imply $Cons(P,Q) + Cons(Q,R) - 1 \leq \frac{x+y+nxy}{n+2} = Cons(P,R)$ ... (v).

5 Concluding Remarks

This paper is only the beginning of a research on a logic of consistency of knowledges where knowledge is in the context of incomplete information understood basically as proposed by Pawlak. We foresee an interesting logic being developed and significant applications of the concept Cons and the the operator C.

References

1. Hajek, P.: Mathematics of fuzzy logic. Kluwer Academic Publisher, Boston (1998)
2. Klir, G.J., Yuan, B.: Fuzzy sets and fuzzy logic: Theory and Applications. Prentice-Hall, India (1997)
3. Nguyen, N.T, Malowiecki, M.: Consistency Measures for Conflict Profiles. Transactions on Rough Sets. 1, 169–186 (2004)
4. Novotný, M., Pawlak, Z.: Partial Dependency of Attributes. Bull. Polish Acad. of Sci. math. 36, 453–458 (1988)
5. Pawlak, Z.: Rough Sets. Internal Journal of Information and Computer Science 11, 341–356 (1982)
6. Pawak, Z.: On Rough Dependency of Attributes in Information System. Bull. Polish Acad. of Sci. Math. 33, 551–559 (1985)
7. Pawak, Z.: Rough sets- Theoritical Aspects of Reasoning About Data. Kluwer Academic Publisher, Boston (1991)
8. Sakai, H., Okuma, A.: Basic Algorithm and Tools for Rough Non-deterministic Information Analysis. Transactions on Rough Sets. 1, 209–226 (2004)

On Three Closely Related Rough Inclusion Functions

Anna Gomolińska*

University of Białystok, Department of Mathematics,
Akademicka 2, 15267 Białystok, Poland
anna.gom@math.uwb.edu.pl

Abstract. The aim of this article is to explore further the idea leading to the *standard rough inclusion function* (standard RIF for short). In fact, two more RIFs may be derived which are different from the standard RIF, yet definable by means of it. We examine properties of the three RIFs and, in particular, the relationships among them.

To the memory of Professor Zdzisław Pawlak

1 Introduction

Broadly speaking, rough inclusion functions (RIFs) are mappings which measure the degree of inclusion of sets of objects in sets of objects[1]. The formal notion of RIF was worked out within *rough mereology*, proposed by Polkowski and Skowron [1,2,3]. Rough mereology extends Leśniewski's mereology [4], a formal theory of being-part to the case of being-part-in-degree. The most famous RIF is the *standard* one, based on the frequency count in line with Łukasiewicz's idea [5][2]. Apart from the standard RIF, there are only several functions of such sort described in the literature (see, e.g., [3,6,7,8]).

Although the notion of RIF was dispensable when approximating concepts in the classical Pawlak rough-set model [9], it is of importance for more general rough-set models and many other issues. First of all, it is a basic component of Skowron–Stepaniuk's approximation spaces [10], where it is used to define rough approximations of concepts. Starting with a RIF, one can derive a family of *rough membership functions* what was already observed by Pawlak and Skowron in [11]. Also various mappings measuring similarity between concepts may be defined by means of RIFs [3,6,12,13]. In [8], a rough-set approach to knowledge reduction, based on the degree of inclusion, is proposed.

In this paper, we explore further the idea leading to the standard RIF, aiming at discovery of other RIFs. It is motivated by the fact that although the standard RIF is undoubtedly well-grounded, useful, and very popular, some of its

* Many thanks to the anonymous referees for interesting and useful comments which helped improve the paper.

[1] A set of objects is often called a *concept* in the rough-set framework.

[2] It is worth mentioning that the very idea underlies the well-known notion of confidence of a rule.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 142–151, 2007.

properties may seem to be too strong (e.g., Proposition 2a,b). Apart from that, it would be good to have alternative RIFs at our disposal as well. In result, we have obtained two RIFs more one of which is really new, whereas the remaining one was mentioned in [7]. We investigate properties of the three RIFs with emphasis on the mutual relationships among them. As regards the standard RIF, some of the properties are already known, other ones are new, at least up to the author's knowledge. As it turns out, the RIFs discovered are different from, yet definable in terms of the standard RIF. Also the latter RIF may be derived from the new ones.

The rest of the paper is organized as follows. Section 2 is entirely devoted to the standard RIF. In Sect. 3, the formal notion of rough inclusion, introduced by Polkowski and Skowron in [1], is presented. In Sect. 4, two alternatives of the standard RIF are derived and their properties are investigated. The last section contains final remarks.

2 The Standard Rough Inclusion Function

The idea underlying the notion of standard rough inclusion function[3] may be attributed to Jan Łukasiewicz, a famous Polish logician who, among other things, conducted research on probability of truth of propositional formulas [5].

Consider a structure M with a non-empty universe U and a propositional language L interpretable over M. For any formula α and $u \in U$, $u \models \alpha$ reads as 'α is satisfied by u' or 'u satisfies α'. The *extension* of α is the set $||\alpha|| \overset{\text{def}}{=} \{u \in U \mid u \models \alpha\}$. α is *satisfiable* in M if its extension is non-empty, and *unsatisfiable* otherwise. Morever, α is called *true* in M, $\models \alpha$, if $||\alpha|| = U$. Finally, α entails a formula β, written $\alpha \models \beta$, if and only if every object satisfying α satisfies β as well, i.e., $||\alpha|| \subseteq ||\beta||$. In classical logic, an implicative formula $\alpha \to \beta$ is true in M if and only if α entails β. Clearly, many interesting formulas are not true in this sense. Since implicative formulas with unsatisfiable predecessors are true, we limit our considerations to satisfiable α. Then, one can assess the degree of truth of $\alpha \to \beta$ by calculating the probability that an object satisfying α, satisfies β as well. Where U is finite, this probability may be approximated by the fraction of objects of $||\alpha||$ which also satisfy β. That is, the degree of truth of $\alpha \to \beta$ may be defined as $\#(||\alpha|| \cap ||\beta||)/\#||\alpha||$, where $\#||\alpha||$ means the cardinality of $||\alpha||$.

By a straithforward generalization, we arrive at the well-known notion of standard RIF, used already in [10]. It owes its popularity within the rough-set community to clarity of the underlying idea and to easiness of computation by means of this notion. Given a non-empty finite set of objects U and its power set $\wp U$, the standard RIF on U is a mapping $\kappa^{£} : \wp U \times \wp U \mapsto [0,1]$ such that for any concepts $X, Y \subseteq U$,

$$\kappa^{£}(X,Y) \overset{\text{def}}{=} \begin{cases} \frac{\#(X \cap Y)}{\#X} & \text{if } X \neq \emptyset \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

[3] And similarly for the notion of confidence of a rule.

To assess the degree of inclusion of a concept X in a concept Y by means of $\kappa^{\pounds}$, one needs to measure the relative overlap of X with Y. The larger the overlap of two sets, the higher is the degree of inclusion, viz., for any sets of objects X, Y, Z,

$$\#(X \cap Y) \leq \#(X \cap Z) \Rightarrow \kappa^{\pounds}(X, Y) \leq \kappa^{\pounds}(X, Z).$$

The success of the standard RIF also lies in its mathematical properties. Where $\mathcal{X}$ is a family of sets, we write $\mathrm{Pair}\mathcal{X}$ to say that elements of $\mathcal{X}$ are pairwise disjoint, i.e., $\forall X, Y \in \mathcal{X}.(X \neq Y \Rightarrow X \cap Y = \emptyset)$. It is assumed that conjunction and disjunction will take the precedence to implication and double implication.

Proposition 1. *For any $X, Y, Z \subseteq U$ and any non-empty families $\mathcal{X}, \mathcal{Y} \subseteq \wp U$, it holds:*

$$(a)\ \kappa^{\pounds}(X, Y) = 1 \Leftrightarrow X \subseteq Y$$
$$(b)\ Y \subseteq Z \Rightarrow \kappa^{\pounds}(X, Y) \leq \kappa^{\pounds}(X, Z)$$
$$(c)\ \kappa^{\pounds}(X, \bigcup \mathcal{Y}) \leq \sum_{Y \in \mathcal{Y}} \kappa^{\pounds}(X, Y)$$
$$(d)\ X \neq \emptyset \ \&\ \mathrm{Pair}\mathcal{Y} \Rightarrow \kappa^{\pounds}(X, \bigcup \mathcal{Y}) = \sum_{Y \in \mathcal{Y}} \kappa^{\pounds}(X, Y)$$
$$(e)\ \kappa^{\pounds}(\bigcup \mathcal{X}, Y) \leq \sum_{X \in \mathcal{X}} \kappa^{\pounds}(X, Y) \cdot \kappa^{\pounds}(\bigcup \mathcal{X}, X)$$
$$(f)\ \mathrm{Pair}\mathcal{X} \Rightarrow \kappa^{\pounds}(\bigcup \mathcal{X}, Y) = \sum_{X \in \mathcal{X}} \kappa^{\pounds}(X, Y) \cdot \kappa^{\pounds}(\bigcup \mathcal{X}, X)$$

Proof. We prove (e) only. Consider any concept Y and any non-empty family of concepts $\mathcal{X}$. First suppose that $\bigcup \mathcal{X} = \emptyset$, i.e., $\mathcal{X} = \{\emptyset\}$. The property clearly holds since $\kappa^{\pounds}(\bigcup \mathcal{X}, Y) = 1$ and $\kappa^{\pounds}(\bigcup \mathcal{X}, \emptyset) \cdot \kappa^{\pounds}(\emptyset, Y) = 1 \cdot 1 = 1$. Now, let $\bigcup \mathcal{X}$ be non-empty. In such a case, $\kappa^{\pounds}(\bigcup \mathcal{X}, Y) = \#(\bigcup \mathcal{X} \cap Y)/\#\bigcup \mathcal{X} = \#\bigcup\{X \cap Y \mid X \in \mathcal{X}\}/\#\bigcup \mathcal{X} \leq \sum\{\#(X \cap Y) \mid X \in \mathcal{X}\}/\#\bigcup \mathcal{X} = \sum\{\#(X \cap Y)/\#\bigcup \mathcal{X} \mid X \in \mathcal{X}\}$. Observe that if some element X of $\mathcal{X}$ is empty, then $\#(X \cap Y)/\#\bigcup \mathcal{X} = 0$ and, on the other hand, $\kappa^{\pounds}(X, Y) \cdot \kappa^{\pounds}(\bigcup \mathcal{X}, X) = 1 \cdot (\#X/\#\bigcup \mathcal{X}) = 1 \cdot 0 = 0$ as well. For every non-empty element X of $\mathcal{X}$, we have $\#(X \cap Y)/\#\bigcup \mathcal{X} = (\#(X \cap Y)/\#X) \cdot (\#X/\#\bigcup \mathcal{X}) = \kappa^{\pounds}(X, Y) \cdot \kappa^{\pounds}(\bigcup \mathcal{X}, X)$ as required. Summarizing, $\kappa^{\pounds}(\bigcup \mathcal{X}, Y) \leq \sum_{X \in \mathcal{X}} \kappa^{\pounds}(X, Y) \cdot \kappa^{\pounds}(\bigcup \mathcal{X}, X)$. □

Some comments may be useful here. (a) says that the standard RIF yields 1 if and only if the 1st argument is included in the 2nd one. According to (b), the degree of inclusion of a concept X in a concept Z is at least as high as the degree of inclusion of X in any subset of Z. It follows from (c) that for any covering of a concept, say Z, the sum of the degrees of inclusion of a concept X in the concepts constituting the covering is at least as high as the degree of inclusion of X in Z. The non-strict inequality in (c) may be strenghtened to = for non-empty X and coverings consisting of pairwise disjoint concepts as stated by (d). Due to (e), for any covering of a concept, say Z, the degree of inclusion of Z in a concept Y is not higher than a weighted sum of the degrees of inclusion of concepts constituting the covering in Y, where the weights are the degrees of inclusion of Z in the members of the covering of Z. Again, as said in (f), the inequality may be strenghtened to = if elements of the covering are pairwise disjoint.

The following conclusions may be drawn from the facts above.

Proposition 2. *For any $X, Y, Z, W \subseteq U$ where $X \neq \emptyset$, and a family $\mathcal{Y}$ of pairwise disjoint subsets of U such that $\bigcup \mathcal{Y} = U$, we have:*

$$(a)\ \sum_{Y \in \mathcal{Y}} \kappa^{\pounds}(X, Y) = 1$$

$$(b)\ \kappa^{\pounds}(X, Y) = 0 \Leftrightarrow X \cap Y = \emptyset$$

$$(c)\ \kappa^{\pounds}(X, \emptyset) = 0$$

$$(d)\ X \cap Y = \emptyset \Rightarrow \kappa^{\pounds}(X, Z - Y) = \kappa^{\pounds}(X, Z \cup Y) = \kappa^{\pounds}(X, Z)$$

$$(e)\ Z \cap W = \emptyset \Rightarrow \kappa^{\pounds}(Y \cup Z, W) \leq \kappa^{\pounds}(Y, W) \leq \kappa^{\pounds}(Y - Z, W)$$

$$(f)\ Z \subseteq W \Rightarrow \kappa^{\pounds}(Y - Z, W) \leq \kappa^{\pounds}(Y, W) \leq \kappa^{\pounds}(Y \cup Z, W)$$

Proof. We show (d) only. To this end, consider any concepts X, Y, where $X \neq \emptyset$ and $X \cap Y = \emptyset$. Immediately, (d1) $\kappa^{\pounds}(X, Y) = 0$ by (b). Hence, for any concept Z, $\kappa^{\pounds}(X, Z) = \kappa^{\pounds}(X, (Z \cap Y) \cup (Z - Y)) = \kappa^{\pounds}(X, Z \cap Y) + \kappa^{\pounds}(X, Z - Y) \leq \kappa^{\pounds}(X, Y) + \kappa^{\pounds}(X, Z - Y) = \kappa^{\pounds}(X, Z - Y)$ in virtue of Proposition 1b,d. In the sequel, $\kappa^{\pounds}(X, Z \cup Y) \leq \kappa^{\pounds}(X, Z) + \kappa^{\pounds}(X, Y) = \kappa^{\pounds}(X, Z)$ due to (d1) and Proposition 1c. The remaining inequalities are consequences of Proposition 1b. □

Let us note a few remarks. Property (a) states that the degrees of inclusion of a non-empty concept X in pairwise disjoint concepts which, taken together, cover the universe sum up to 1. In virtue of (b), the degree of inclusion of a non-empty concept in any concept equals to 0 just in case the both concepts are disjoint. (b) implies (c), where the latter says that the degree of inclusion of a non-empty concept in $\emptyset$ is equal to 0. Thanks to (d), removing (resp., adding) objects, not being members of a non-empty concept X, from (to) a concept Z does not influence the degree of inclusion of X in Z. As follows from (e), adding (resp., removing) objects, not belonging to a concept W, to (from) a concept Y does not increase (decrease) the degree of inclusion of Y in W. Finally, by (f), removing (resp., adding) members of a concept W from (to) a concept Y does not increase (decrease) the degree of inclusion of Y in W.

Example 1. Given $U = \{0, \ldots, 9\}$, $X = \{0, \ldots, 3\}$, $Y = \{0, \ldots, 3, 8\}$, and $Z = \{2, \ldots, 6\}$. Note that $X \cap Z = Y \cap Z = \{2, 3\}$. Thus, $\kappa^{\pounds}(X, Z) = 1/2$ and $\kappa^{\pounds}(Z, X) = 2/5$ which means that the standard RIF is not symmetric. Moreover, $\kappa^{\pounds}(Y, Z) = 2/5 < 1/2$. Thus, $X \subseteq Y$ may not imply $\kappa^{\pounds}(X, Z) \leq \kappa^{\pounds}(Y, Z)$, i.e., $\kappa^{\pounds}$ is not monotone in the 1st variable.

3 The Formal Notion of Rough Inclusion

The notion of standard RIF was generalized and formalized by Polkowski and Skowron within rough mereology, a theory of the notion of being-part-in-degree [1,2,3]. The starting point is a pair of formal theories, Leśniewski's mereology and ontology [4]. Mereology, a theory of the notion of being-part, is based on ontology, being a theory of names and playing the role of set theory[4]. In ontology, two basic semantical categories are distinguished: the category of names and the category of propositions. We use x, y, z, with subscripts if needed, as name variables. With every name x, there is associated a distributive class of individuals designated by the name, $|x|$. The empty name designates no entity at all. In Leśniewski's approach, only non-empty names are typically considered as the empty set is denied on philosophical grounds. The only primitive notion of ontology is the copula 'is', denoted by ε and characterized by the following axiom:

$$(L0)\ x\varepsilon y \leftrightarrow (\exists z.z\varepsilon x \wedge \forall y, z.(y\varepsilon x \wedge z\varepsilon x \rightarrow y\varepsilon z) \wedge \forall z.(z\varepsilon x \rightarrow z\varepsilon y))$$

$x\varepsilon y$ is read as 'x is y'. According to the standard interpretation, $x\varepsilon y$ is true if and only if x is an individual name, and the only entity designated by x is designated by y as well. In particular, $x\varepsilon x$ is true just in case x is an individual name.

Mereology is built upon ontology and introduces a name-forming functor pt, where $x\varepsilon\mathrm{pt}(y)$ reads as 'x is a *part* of y', characterized by the following axioms:

$$(L1)\ x\varepsilon\mathrm{pt}(y) \rightarrow x\varepsilon x \wedge y\varepsilon y\ \ (x, y \text{ have to be individual names})$$
$$(L2)\ x\varepsilon\mathrm{pt}(y) \wedge y\varepsilon\mathrm{pt}(z) \rightarrow x\varepsilon\mathrm{pt}(z)\ \text{(transitivity)}$$
$$(L3)\ \neg(x\varepsilon\mathrm{pt}(x))\ \text{(irreflexivity)}$$

The reflexive counterpart of being-part is the notion of *being-ingredient*, ing, given by

$$x\varepsilon\mathrm{ing}(y) \overset{\mathrm{def}}{\leftrightarrow} x\varepsilon\mathrm{pt}(y) \vee x = y \tag{2}$$

and such that:

$$(L1')\ x\varepsilon\mathrm{ing}(y) \rightarrow x\varepsilon x \wedge y\varepsilon y\ \ (x, y \text{ have to be individual names})$$
$$(L2')\ x\varepsilon\mathrm{ing}(y) \wedge y\varepsilon\mathrm{ing}(z) \rightarrow x\varepsilon\mathrm{ing}(z)\ \text{(transitivity)}$$
$$(L3')\ x\varepsilon\mathrm{ing}(x)\ \text{(reflexivity)}$$
$$(L4')\ x\varepsilon\mathrm{ing}(y) \wedge y\varepsilon\mathrm{ing}(x) \rightarrow x = y\ \text{(antisymmetry)}$$

One can also start with ing characterized by (L1')–(L4') and define pt by

$$x\varepsilon\mathrm{pt}(y) \overset{\mathrm{def}}{\leftrightarrow} x\varepsilon\mathrm{ing}(y) \wedge x \neq y. \tag{3}$$

[4] Leśniewski's mereology is also known as a theory of collective sets as opposite to ontology being a theory of distributive sets.

In Polkowski–Skowron's rough mereology which extends Leśniewski's mereology, a family of name-forming functors ing_t, formalizing the notion of being-ingredient-in-degree and originally denoted by μ_t, is introduced as follows, for any names x, y, z and $s, t \in [0, 1]$:

$(PS1)$ $\exists t.x\varepsilon\mathrm{ing}_t(y) \to x\varepsilon x \wedge y\varepsilon y$ (x, y have to be individual names)

$(PS2)$ $x\varepsilon\mathrm{ing}_1(y) \leftrightarrow x\varepsilon\mathrm{ing}(y)$ (ingredient in degree 1 is ingredient)

$(PS3)$ $x\varepsilon\mathrm{ing}_1(y) \to \forall z.(z\varepsilon\mathrm{ing}_t(x) \to z\varepsilon\mathrm{ing}_t(y))$ (weak transitivity)

$(PS4)$ $x = y \wedge x\varepsilon\mathrm{ing}_t(z) \to y\varepsilon\mathrm{ing}_t(z)$ (= is a congruence)

$(PS5)$ $x\varepsilon\mathrm{ing}_t(y) \wedge s \leq t \to x\varepsilon\mathrm{ing}_s(y)$ (ingredienthood in degree at least t)

Then, being a part in degree t may be defined as a special case of being an ingredient in degree t, viz.,

$$x\varepsilon\mathrm{pt}_t(y) \overset{\mathrm{def}}{\leftrightarrow} x\varepsilon\mathrm{ing}_t(y) \wedge x \neq y. \tag{4}$$

(PS1)–(PS5) specify minimal requirements to be fulfilled by rough inclusion functions, intended as functions measuring the degree of inclusion of concepts in concepts. Consider a structure M with a non-empty set of objects U as the universe. Individual names may designate sets of objects of U, and being an ingredient (resp., part) of a name may be interpreted as being a subset (proper subset) of the set of objects designated by this name. In our approach, where the empty set is allowed for convenience, a RIF upon U is defined as any mapping $\kappa : \wp U \times \wp U \mapsto [0, 1]$ satisfying rif_1 and rif_2 below:

$$\mathrm{rif}_1(\kappa) \overset{\mathrm{def}}{\Leftrightarrow} \forall X, Y.(\kappa(X, Y) = 1 \Leftrightarrow X \subseteq Y)$$
$$\mathrm{rif}_2(\kappa) \overset{\mathrm{def}}{\Leftrightarrow} \forall X, Y, Z.(Y \subseteq Z \Rightarrow \kappa(X, Y) \leq \kappa(X, Z))$$

Thus, according to rif_2, RIFs are monotone in the 2nd variable. On the other hand, as stipulated by rif_1, the greatest value 1 is achieved by a RIF only for such pairs of concepts that the 2nd element of a pair contains the 1st element[5]. Observe that $\mathrm{ing}_t(y)$ may be interpreted in M as the set of all such $X \subseteq U$ that $\kappa(X, Y) \geq t$, where $|y| = \{Y\}$ and κ is a RIF upon U.

Apart from $\mathrm{rif}_1, \mathrm{rif}_2$, RIFs may satisfy other postulates, e.g.:

$$\mathrm{rif}_3(\kappa) \overset{\mathrm{def}}{\Leftrightarrow} \forall X \neq \emptyset.\kappa(X, \emptyset) = 0$$
$$\mathrm{rif}_4(\kappa) \overset{\mathrm{def}}{\Leftrightarrow} \forall X, Y.(\kappa(X, Y) = 0 \Rightarrow X \cap Y = \emptyset)$$
$$\mathrm{rif}_{4*}(\kappa) \overset{\mathrm{def}}{\Leftrightarrow} \forall X \neq \emptyset.\forall Y.(X \cap Y = \emptyset \Rightarrow \kappa(X, Y) = 0)$$
$$\mathrm{rif}_5(\kappa) \overset{\mathrm{def}}{\Leftrightarrow} \forall X \neq \emptyset.\forall Y.(\kappa(X, Y) = 0 \Leftrightarrow X \cap Y = \emptyset)$$
$$\mathrm{rif}_6(\kappa) \overset{\mathrm{def}}{\Leftrightarrow} \forall X \neq \emptyset.\forall Y.\kappa(X, Y) + \kappa(X, U - Y) = 1$$

[5] In particular, the characteristic function of $\subseteq$, $f_\subseteq : \wp U \times \wp U \mapsto \{0, 1\}$, given by $f_\subseteq(X, Y) = 1 \overset{\mathrm{def}}{\Leftrightarrow} X \subseteq Y$ for any $X, Y \subseteq U$, is a RIF.

As follows from Propositions 1 and 2, the standard RIF satisfies all the conditions above. Moreover, it holds for any κ that $\mathrm{rif}_1(\kappa)$ and $\mathrm{rif}_6(\kappa)$ imply $\mathrm{rif}_5(\kappa)$; $\mathrm{rif}_5(\kappa)$ is equivalent to $\mathrm{rif}_4(\kappa)$ and $\mathrm{rif}_{4*}(\kappa)$; and $\mathrm{rif}_{4*}(\kappa)$ implies $\mathrm{rif}_3(\kappa)$.

4 In Search of RIFs Other Than the Standard One

According to rough mereology, rough inclusion is a generalization of the set-theoretical inclusion of sets. While keeping with this idea, we try to obtain RIFs different from the standard one. Observe that for any $X, Y \subseteq U$, where U is a non-empty finite set of objects as earlier, the following formulas are equivalent:

$$\begin{array}{rl} (i) & X \subseteq Y \\ (ii) & X \cap Y = X \\ (iii) & X \cup Y = Y \\ (iv) & (U - X) \cup Y = U \end{array} \tag{5}$$

The equivalence of the first two statements gave rise to the standard RIF. Now, we explore $(i) \Leftrightarrow (iii)$ and $(i) \Leftrightarrow (iv)$. In the case of (iii), '$\supseteq$' always holds true. Conversely, '$\subseteq$' always takes place in (iv). The remaining inclusions may or may not hold, so we may introduce degrees of inclusion. Thus, let us define mappings $\kappa_1, \kappa_2 : \wp U \times \wp U \mapsto [0, 1]$ such that for any concepts X, Y,

$$\kappa_1(X,Y) \overset{\text{def}}{=} \begin{cases} \frac{\#Y}{\#(X \cup Y)} & \text{if } X \cup Y \neq \emptyset \\ 1 & \text{otherwise}, \end{cases}$$

$$\kappa_2(X,Y) \overset{\text{def}}{=} \frac{\#((U - X) \cup Y)}{\#U}. \tag{6}$$

It is worth noting that κ_2 is mentioned in [7]. Now, we show that both κ_1, κ_2 are RIFs (i.e., they satisfy $\mathrm{rif}_1, \mathrm{rif}_2$) different from the standard RIF and from each other.

Proposition 3. *For* $i = 1, 2$, $\mathrm{rif}_1(\kappa_i)$ *and* $\mathrm{rif}_2(\kappa_i)$.

Proof. We only prove the property for $i = 1$. To this end, let $X, Y, Z \subseteq U$ be any concepts. In the case of rif_1, we only examine the non-trivial case, where $X, Y \neq \emptyset$. Then, $\kappa_1(X,Y) = 1$ if and only if $\#Y = \#(X \cup Y)$ if and only if $Y = X \cup Y$ if and only if $X \subseteq Y$. To show $\mathrm{rif}_2(\kappa_1)$, assume that (a1) $Y \subseteq Z$. If $X = Y = Z = \emptyset$, then $\kappa_1(X,Y) = \kappa_1(X,Z) = 1$ as required. Next, if $X = \emptyset$ and $Y \neq \emptyset$, then $\kappa_1(X,Y) = \#Y/\#Y = 1$, and similarly for Z. In this way, $X = \emptyset$ implies $\kappa_1(X,Y) = \kappa_1(X,Z) = 1$ as needed. Finally, if $X \neq \emptyset$, then $X \cup Y, X \cup Z \neq \emptyset$. Moreover, $Z = Y \cup (Z - Y)$ and $Y \cap (Z - Y) = \emptyset$ by (a1). As a consequence, (a2) $\#Z = \#Y + \#(Z - Y)$. Additionally, (a3) $\#(X \cup Z) \leq \#(X \cup Y) + \#(Z - Y)$ and (a4) $\#Y \leq \#(X \cup Y)$. Hence, $\kappa_1(X,Y) = \#Y/\#(X \cup Y) \leq (\#Y + \#(Z - Y))/(\#(X \cup Y) + \#(Z - Y)) \leq (\#Y + \#(Z - Y))/\#(X \cup Y \cup (Z - Y)) = \#Z/\#(X \cup Z) = \kappa_1(X,Z)$ by (a2)-(a4). □

Example 2. Consider $U = \{0,\ldots,9\}$ and its subsets $X = \{0,\ldots,4\}$, $Y = \{2,\ldots,6\}$. Notice that $X \cap Y = \{2,3,4\}$, $X \cup Y = \{0,\ldots,6\}$, and $(U-X)\cup Y = \{2,\ldots,9\}$. Hence, $\kappa^{\pounds}(X,Y) = 3/5$, $\kappa_1(X,Y) = 5/7$, and $\kappa_2(X,Y) = 4/5$, i.e., $\kappa^{\pounds}, \kappa_1, \kappa_2$ are different RIFs.

Proposition 4. *For any concepts X, Y, Z, we have:*

$$(a)\ X \neq \emptyset \Rightarrow (\kappa_1(X,Y) = 0 \Leftrightarrow Y = \emptyset)$$
$$(b)\ \kappa_2(X,Y) = 0 \Leftrightarrow X = U \ \&\ Y = \emptyset$$
$$(c)\ \mathrm{rif}_4(\kappa_1)\ \ \&\ \ \mathrm{rif}_4(\kappa_2)$$
$$(d)\ \kappa^{\pounds}(X,Y) \leq \kappa_1(X,Y) \leq \kappa_2(X,Y)$$
$$(e)\ \kappa_1(X,Y) = \kappa^{\pounds}(X \cup Y, Y)$$
$$(f)\ \kappa_2(X,Y) = \kappa^{\pounds}(U,(U-X)\cup Y) = \kappa^{\pounds}(U, U-X) + \kappa^{\pounds}(U, X\cap Y)$$
$$(g)\ \kappa^{\pounds}(X,Y) = \kappa^{\pounds}(X, X\cap Y) = \kappa_1(X, X\cap Y) = \kappa_1(X-Y, X\cap Y)$$

Proof. By way of illustration, we show (d). To this end, consider any concepts X, Y. If X is empty, then $(U-X)\cup Y = U$. Hence, by the definitions of the RIFs, $\kappa^{\pounds}(X,Y) = \kappa_1(X,Y) = \kappa_2(X,Y) = 1$. Now, suppose that $X \neq \emptyset$. Obviously, (d1) $\#(X\cap Y) \leq \#X$ and (d2) $\#Y \leq \#(X\cup Y)$. Since $X\cup Y = X\cup(Y-X)$ and $X\cap(Y-X) = \emptyset$, (d3) $\#(X\cup Y) = \#X + \#(Y-X)$. Similarly, it follows from $Y = (X\cap Y)\cup(Y-X)$ and $(X\cap Y)\cap(Y-X) = \emptyset$ that (d4) $\#Y = \#(X\cap Y) + \#(Y-X)$. Observe also that $(U-X)\cup Y = ((U-X)-Y)\cup Y = (U-(X\cup Y))\cup Y$ and $(U-(X\cup Y))\cap Y = \emptyset$. Hence, (d5) $\#((U-X)\cup Y) = \#(U-(X\cup Y)) + \#Y$. In the sequel, $\kappa^{\pounds}(X,Y) = \#(X\cap Y)/\#X \leq (\#(X\cap Y) + \#(Y-X))/(\#X + \#(Y-X)) = \#Y/\#(X\cup Y) = \kappa_1(X,Y) \leq (\#(U-(X\cup Y)) + \#Y)/(\#(U-(X\cup Y)) + \#(X\cup Y)) = \#((U-X)\cup Y)/\#U = \kappa_2(X,Y)$ by (d1)-(d5) and the definitions of the RIFs. □

Let us briefly comment on the properties. According to (a), if $X \neq \emptyset$, then emptiness of Y is not only sufficient (as claimed by rif_3) but also necessary condition for $\kappa_1(X,Y) = 0$. (b) says that κ_2 yields 0 solely for $(U,\emptyset)$. Due to (c), the both RIFs satisfy rif_4. (d) states that the degree of inclusion of a concept X in a concept Y given by κ_2 is at least as high as that one yielded by κ_1, and the latter is not lower than the degree obtained by means of the standard RIF. Properties (e) and (f) provide us with characterizations of κ_1 and κ_2 in terms of $\kappa^{\pounds}$, respectively. On the other hand, the standard RIF may be defined by means of κ_1 in virtue of (g).

With every mapping $f : \wp U \times \wp U \mapsto [0,1]$, we can associate a "complementary" mapping $\bar{f} : \wp U \times \wp U \mapsto [0,1]$ defined by

$$\bar{f}(X,Y) \overset{\mathrm{def}}{=} 1 - f(X,Y), \tag{7}$$

for any concepts X, Y. Observe that

$$\bar{\kappa}^{\pounds}(X,Y) = \kappa^{\pounds}(X, U-Y),$$

$$\bar{\kappa}_1(X,Y) = \begin{cases} \frac{\#(X-Y)}{\#(X\cup Y)} & \text{if } X \cup Y \neq \emptyset \\ 0 & \text{otherwise,} \end{cases}$$

$$\bar{\kappa}_2(X,Y) = \frac{\#(X-Y)}{\#U}. \tag{8}$$

Example 3. Let us note that unlike in the standard case, for $i = 1,2$, $\kappa_i(X, U - Y) \neq \bar{\kappa}_i(X,Y)$ in general. Indeed, for non-empty $X \cup (U - Y)$, $\kappa_1(X, U - Y) = \#(U - Y)/\#(X \cup (U - Y))$. Furthermore, $\kappa_2(X, U - Y) = \kappa_2(X \cap Y, \emptyset)$.

As regarding properties of $\bar{\kappa}$, we only show how the standard RIF can be expressed in terms of κ_i $(i = 1,2)$ and their complementary functions.

Proposition 5. *For any concepts X, Y where $X \neq \emptyset$,*

$$\kappa^{\pounds}(X,Y) = \frac{\bar{\kappa}_1(X, U-Y)}{\kappa_1(U-Y, X)} = \frac{\bar{\kappa}_2(X, U-Y)}{\kappa_2(U, X)}.$$

Proof. Consider any concepts X, Y and assume non-emptiness of X. Hence, $X \cup (U - Y) \neq \emptyset$ as well. Moreover, $\kappa_1(U - Y, X), \kappa_2(U, X) > 0$. Then, $\bar{\kappa}_1(X, U - Y) = \#(X - (U - Y))/\#(X \cup (U - Y)) = \#(X \cap Y)/\#(X \cup (U - Y)) = (\#(X \cap Y)/\#X) \cdot (\#X/\#(X \cup (U - Y))) = \kappa^{\pounds}(X,Y) \cdot \kappa_1(U - Y, X)$ by the definitions of $\kappa^{\pounds}$, κ_1, and $\bar{\kappa}_1$. Hence, $\kappa^{\pounds}(X,Y) = \bar{\kappa}_1(X, U - Y)/\kappa_1(U - Y, X)$ as required. Similarly, $\bar{\kappa}_2(X, U - Y) = \#(X - (U - Y))/\#U = \#(X \cap Y)/\#U = (\#(X \cap Y)/\#X) \cdot (\#X/\#U) = \kappa^{\pounds}(X,Y) \cdot \kappa_2(U,X)$ by the definitions of $\kappa^{\pounds}$, κ_2, and $\bar{\kappa}_2$. Immediately, $\kappa^{\pounds}(X,Y) = \bar{\kappa}_2(X, U - Y)/\kappa_2(U,X)$ what ends the proof. □

5 Final Remarks

In this article, an attempt was made to discover RIFs different from the standard one, yet having similar origin. First, we overviewed the notion of the standard RIF. In the next step, a general framework for discussion of RIFs and their properties was recalled. As a result, a minimal set of postulates specifying a RIF was derived. Also, several additional, optional conditions were proposed[6]. Then, we defined two RIFs which turned out to be different from the standard one. One of them was mentioned in [7], the remaining one seems to be completely new. We examined properties of the two RIFs with a special stress laid on the relationship to the standard RIF. Apart from that, we introduced functions in some sense complementary to RIFs what resulted in a new characterization of the standard RIF in terms of the new ones.

For the time being, our results have a theoretical value. It would be interesting to apply the two RIFs to some practical issues. Another task is to relate the results to those reported, e.g., in [8]. One more direction for the future research

[6] The list is subject to extension and modification.

is an extension of the notion of RIF to include such mappings as, e.g., $\nu : \wp U \times \wp U \mapsto [0,1]$ defined by

$$\nu(X,Y) \overset{\text{def}}{=} \kappa(\text{upp}X, \text{upp}Y), \tag{9}$$

where $X, Y \subseteq U$, κ is a RIF, and upp is an upper approximation mapping in the sense of Pawlak [9]. One can see that in an approximation space (U, ϱ, κ) based on a similarity relation ϱ, where the upper approximation of a concept X is defined by $\text{upp}X = \{u \mid \varrho^{\leftarrow}\{u\} \cap X \neq \emptyset\}$, rif_2 is satisfied for ν, yet only one half of rif_1 holds for it. Examples of interesting mappings which may serve the purpose of measurement of the inclusion of concepts in concepts and show the same feature can be found in [3,6].

References

1. Polkowski, L., Skowron, A.: Rough mereology. Lecture Notes in Artificial Intelligence 869, 85–94 (1994)
2. Polkowski, L., Skowron, A.: Rough mereology: A new paradigm for approximate reasoning. Int. J. Approximated Reasoning 15(4), 333–365 (1996)
3. Polkowski, L., Skowron, A.: Rough mereological calculi of granules: A rough set approach to computation. Int. J. Comput. Intelligence 17(3), 472–492 (2001)
4. Leśniewski, S.: Foundations of the General Set Theory 1 (in Polish). Volume 2 of Works of the Polish Scientific Circle. Moscow (1916). In: Surma, S. J. et al. (ed.) Stanisław Leśniewski Collected Works, Kluwer Acad. Publ. Dordrecht, pp. 128–173 (1992)
5. Łukasiewicz, J.: Die logischen Grundlagen der Wahrscheinlichkeitsrechnung. In: Borkowski, L. (ed.) Jan Łukasiewicz – Selected Works, North Holland, pp. 16–63. Polish Scientific Publ, Amsterdam (1970). First published in Kraków, 1913
6. Stepaniuk, J.: Knowledge discovery by application of rough set models. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.) Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems, pp. 137–233. Physica-Verlag, Heidelberg (2001)
7. Drwal, G., Mrózek, A.: System RClass – software implementation of a rough classifier. In: Kłopotek, M.A., Michalewicz, M., Raś, Z.W. (eds.) Proc. 7th Int. Symposium on Intelligent Information Systems (IIS'1998), Malbork, Poland, 1998, pp. 392–395 (1998)
8. Zhang, M., et al.: A rough set approach to knowledge reduction based on inclusion degree and evidence reasoning theory. Expert Systems 20(5), 298–304 (2003)
9. Pawlak, Z.: Rough Sets. In: Theoretical Aspects of Reasoning About Data, Kluwer, Dordrecht (1991)
10. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27(2–3), 245–253 (1996)
11. Pawlak, Z., Skowron, A.: Rough membership functions. In: Fedrizzi, M., Kacprzyk, J., Yager, R.R. (eds.) Advances in the Dempster–Shafer Theory of Evidence, pp. 251–271. John Wiley & Sons, New York (1994)
12. Gomolińska, A.: Possible rough ingredients of concepts in approximation spaces. Fundamenta Informaticae 72, 139–154 (2006)
13. Nguyen, H.S., Skowron, A., Stepaniuk, J.: Granular computing: A rough set approach. Int. J. Comput. Intelligence 17(3), 514–544 (2001)

Rough Set Theory from a Math-Assistant Perspective

Adam Grabowski* and Magdalena Jastrzębska

Institute of Mathematics, University of Białystok
ul. Akademicka 2, 15-267 Białystok, Poland
adam@math.uwb.edu.pl, magjas0@poczta.onet.pl

Abstract. In the paper, we draw a perspective of the computer-assisted theory exploration within rough set theory. We examine two well-known approaches to the topic, drawing some paradigms for a machine math-assistant to be feasible tool any researcher can use to verify his own results. Some features of a Mizar language chosen for the verification task are also presented.

1 Introduction

This paper is a survey of the development of rough set theory from a machine proof-assistant viewpoint, and a brief summary of basic results encoded in the computer-checked repository of mathematical knowledge formalized using the Mizar system. By *formalization* we mean the encoding of mathematics in a formal language sufficiently detailed for a computer program to verify the correctness. The greatest projects of this kind of a pre-computer age were Whitehead and Russell's "Principia Mathematica" and project Bourbaki. "Checking Landau's 'Grundlagen' in the Automath system" by Jutting (1977) was the first significant step of translating human efforts in a machine-checkable language.

The need of the computer verification of hardware and software emerged pretty recently. After the bug in the first Pentium processor was discovered in 1994, the Intel company established a special group of people doing research in the field of the hardware verification. But the issue of the uncertainty of results appears not only in the industry – also academia can face this problem, especially when publishing is taken into account. As the referees can be uncertain if a proof is really correct, the review procedure can take months, but the situation gets even more frustrating when we take into account the case of Thomas Hales, who has been waiting for five years to hear whether the mathematical community has accepted his 1998 proof of Kepler's conjecture that the most efficient way to pack equal-size spheres is to stack them in the usual pyramid. In 2003 a review panel of world experts appointed by the journal *Annals of Mathematics* finally declared that, whereas they had not found any irreparable error in the proof, they were still not sure that it was correct. The journal finally decided

* Partial support for this work provided by the EU FP6 IST grant TYPES (Types for Proofs and Programs) No. 510996.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 152–161, 2007.

to publish Hales's proof, but the disclaimer saying they were not sure that it was right, was added. Then Hales started the FlySpeck project to formalize his proof with the help of a computer and hence a new paradigm of using machine proof-checker – not as the experimental tool but to solve real-life problems – was confirmed.

The paper is organized as follows. The next section contains a brief summary of basic notions of RST formalized in the Mizar language, as well as some highlights of the system used for this purpose. Section 3 provides the discussion of various approaches to rough sets while in the fourth section we deal with the more general model for I-sets, i.e. interval sets, and then we discuss rough sets from a lattice-theoretical point of view. In Section 6 we sketch some advantages of machine support in the process of knowledge exploring. The paper ends with some conclusions and plans for future work.

2 A Primer of Rough Set Theory, Formal Approach

The previous century has brought many automated theorem proving projects (and so the work of Pawlak in this direction reflected contemporary trends) and also a few realizations of the idea of machine-checking proofs for their correctness. As a first, probably most notable, we can point out the aforementioned work of Jutting in the de Bruijn's system Automath. Usually, a researcher which is not well acquainted with automated theorem proving, gets know only about very large and successful formalization projects. The most impressive (and/or probably also best advertised) examples were the solution of the Robbins problem which was open for over sixty years, solved by automated equational theorem prover EQP/Otter, Four Color Theorem with the proof done in Coq, the Jordan Curve Theorem recently completed in HOL and later in Mizar.

Andrzej Trybulec, the designer of the Mizar system, in a private communication admitted that the person who influenced positively his early researches on the translation from a natural mathematical vernacular into a machine-understandable language (so, also the development of the Mizar language), was Zdzisław Pawlak. When they met in the seventies of the previous century and discussed a bit the problems emerging somewhere at the intersection of the human–computer spheres, Professor Pawlak suggested the application for a grant at IPI PAN (Polish acronym of the Institute of Computer Science of the Polish Academy of Sciences, although the name was yet slightly different). Trybulec and his group followed the advice of the designer of a first Polish digital computer and the unquestionable authority in the field of the young emerging discipline of the computer science (exploring topics of the automated reasoning and the mathematical model of a computer, both reflected in Trybulec's system), they did so, succeded and eventually got the financing. It was extremely important, because as yet it can be remembered, the access to a computing machine, necessary for experiments with automated reasoning, was rather complicated and highly cost-consuming those days. For sure, the language evolved from its predicative form which was popular some thirty years ago, to somewhat closer to its natural original. Also some physical bounds vanished – computer parameters are better and better, which allows us to face the problems inaccessible a couple decades ago. Some paradigms about human–computer interaction remained unchanged, though; we guess that many visionary ideas Zdzisław Pawlak had in mind, were influential for lots of people involved in the computer science, treated in the very general setting.

The Mizar language is a formal language close to the vernacular used in mathematical publications. An implemented Mizar verifier is available for checking correctness of Mizar texts according to Jaśkowski natural deduction. The perpetual development of the Mizar system (see [17]) has resulted in the Mizar Mathematical Library (MML) – a centrally maintained library of formalized mathematics based on Tarski-Grothendieck set theory which is a variant of ZFC.

The MML is organized as a cross-linked collection of the items called Mizar *articles*. As of the time of writing, there are 959 articles in the whole library, occupying 70 MB, containing 43149 theorems and 8185 definitions. It is commonly considered the biggest library of computer proof-checked mathematics (possessing e.g., recent proof of the Jordan Curve Theorem) and as such is also the subject of the research of data-miners (e.g., TPTP – Thousands of Problems for Theorem Provers). However not yet based on GNU license, the system is free, available for most popular platforms: MS Windows, Unixes, and MacOS on PowerPC. System requirements for installing both binaries and database are rather modest; about 200 MB of disk space to uncompress the full distribution.

2.1 Towards Formal Approximation Spaces

According to the classical paper of Pawlak [9], rough sets are based on the equivalence relations. Shortly thereafter, there were considered in the literature more general approaches, e.g. transitivity of the indiscernibility relation was dropped (see [7,11,10] for some paths of research, not only without transitivity).

Some of the natural properties are true only for the case of equivalence relations, which may make the theorems heterogeneous in some sense – some of them will require more complex assumptions under which they remain valid. So to keep his/her work more unified in style, the author could decide to formulate all of them in terms of approximation spaces. This approach, although transparent from the user's perspective, is hardly acceptable from the viewpoint of the knowledge reusability. We will write e.g. the upper approximation of the subset A of a universe U with respect to indiscernibility relation I classically as

$$\mathit{upp}_I(A) = \{x \in U : [x]_I \cap A \neq \emptyset\},$$

but using some hidden arguments (we take into account the space R with determined universe U and I being not necessarily an equivalence relation).

```
definition let X be Tolerance_Space, A be Subset of X;
  func UAp A -> Subset of A equals :: ROUGHS_1:def 5
    { x where x is Element of X :
      Class (the InternalRel of X, x) meets A };
end;
```

It should be noticed here that to start with rough sets we formalized the definition of approximation spaces based on tolerance relations whenever possible, we introduced membership functions with selected basic properties, and also provided the definition of rough sets. Some lines are devoted to various predicates of rough inclusion and rough equality. This primary development can be browsed under MML Identifier `ROUGHS_1` from the Mizar home page [1].

2.2 The State of the Art

A more or less formally axiomatized view for rough sets is not a novelty: Bryniarski [1] or Yao [12] are good representatives, not to enumerate yet classical [9] and [11]. But if we require the possibility of proof checking by the computer, the choice is not that wide although the idea of automatic correctness checking is also known. This approach presents relative uniformity – usually there is a unique definition because the Library Committee which takes care of the collection of articles does not allow for duplication of concepts and the library users report such repetitions. But also heterogeneity is not excluded completely – one can introduce constructions called redefinitions, which can result in having two approaches effectively benefitting from their equivalence. A good example is the definition of the rough equality of sets which is, on the one hand, the simultaneous equality of their upper and lower approximations, on the other hand – the conjunction of two rough inclusions.

The MML is roughly divided in three parts – concrete (based on pure set theory), abstract (where structures, including algebraic ones, as e.g. groups, lattices, vector and topological spaces are defined), and that devoted to the formalization of random access Turing machines, i.e. mathematical model of a computer, first decisions which had to be made were how to define approximations; because it was pretty clear approximation spaces should have been put in the abstract part of the library due to its strong algebraic flavour.

The correspondence between the very basic notions of the rough set theory chosen and their formal translated counterparts is given in Table 1. In the table the dot sign "`.`" stands for the application of a membership function, the brackets are used mainly for grouping multiple arguments of Mizar functors (i.e. a kind of language functions), formulas, etc. During the process of the automatic translation into the natural language (resulting also in the LaTeX source) Mizar functors are not typeset verbatim, but translated either in a way proposed by the author, or according to some simple transition rules applied automatically.

3 Two Views for Rough Sets

As widely known, the central notion of a rough set does not have its formal definition uniquely determined. We mean here two set-oriented views for rough sets. The first one is classical, due to Pawlak (P-sets). The other one (sometimes called I-sets in the literature for Iwiński [6]) is based on pairs of definable subsets. The more thorough discussion about such classification can be found in [12].

[1] `http://www.mizar.org/`

Table 1. The correspondence between natural language and Mizar objects

the notion	the Mizar counterpart
$[x]_R$	`Class(R,x)`
the upper approximation of A	`UAp A`
the lower approximation of A	`LAp A`
the boundary region of A	`BndAp A`
the membership function $\mu_X^I(x)$	`MemberFunc(X,I).x`
approximation space	`Approximation_Space`
tolerance approximation space	`Tolerance_Space`
rough upper inclusion	`c=^`
rough lower inclusion	`_c=`
rough inclusion combined	`_c=^`
I-rough set in the universe U	`rough Subset of U`
I-definable set in the universe U	`exact Subset of U`
P-rough set in the universe U	`RoughSet of U`

3.1 Classical Rough Sets

The notion of P-set is based on the original concept of equivalence relation which induces a partition on the field of the underlying relation. In the MML it has no clear representation, even if classes of abstraction are natural mathematical constructions. Classes of P-sets are identified via predicate of rough equality which reads as follows:

```
definition let A be Tolerance_Space, X, Y be Subset of A;
  pred X _=^ Y means
    LAp X = LAp Y & UAp X = UAp Y;
end;
```

In fact, the granularity of definitions is even better – we considered feasible to have distinct predicates for $X =_* Y$ and $X =^* Y$ (where X and Y have resp. their lower and upper approximations equal). Naturally, two sets are equal in the sense of $=^*_*$ iff they are equal in both senses – the lower and the upper equality. Alternatively, we claim that a subset of a tolerance approximation space is rough, if its boundary approximation, i.e. the set-theoretical difference between its upper and lower approximation is not equal to the empty set.

```
definition let A be Tolerance_Space, X be Subset of A;
  attr X is rough means
    BndAp X <> {};
end;
```

Otherwise, we claim that the subset is exact, that is it is a set in the ordinary sense (crisp). It is done via construction of antonyms for adjectives, which allow the user to divide all subsets formally into two disjoint classes.

3.2 Pairs of Definable Sets

I-sets can be considered a natural RST-counterpart of the interval sets – with respect to the same Pawlak approximation space both are uniquely determined by each other. In the Mizar formalism it can be described just as below:

```
definition let A be Tolerance_Space, X be Subset of A;
  mode RoughSet of X means  :: ROUGHS_1:def 8
    it = [LAp X, UAp X];
end;
```

Note however that this does not reduce to the ordinary set even if both approximations are equal. Because the I-model provides the better mathematical description, it was chosen by us to define a lattice of rough sets. Even if set-theoretical operators on the set of all I-sets do not rather have a well-defined semantics, this interpretation provides a mathematical model which is both elegant and can be a subject to further generalizations.

4 Interval Sets

Let us recall the notion of an interval set [12]:

$$[A_1, A_2] = \{A \in 2^U : A_1 \subseteq A \subseteq A_2\} \qquad (1)$$

where U is a finite set called the universe. Usually, the assumption of $A_1 \subseteq A_2$ is granted, but we define an interval set also in case when A_1 is not a subset of A_2. The set of all interval sets over a universe U, with operations $\sqcap$ and $\sqcup$ defined componentwise, forms a lattice. Moreover, it is a distributive and bounded lattice, where $[\emptyset, \emptyset]$ is its bottom and $[U, U]$ – its top. Firstly, an interval set is defined as a family of subsets with two parameters being its boundaries (we dropped an assumption of $X \subseteq Y$ since it can be proven otherwise the resulting interval is empty).[2]

```
definition let U be set, X, Y be Subset of U;
  func Inter (X,Y) -> Subset-Family of U equals
    { A where A is Subset of U : X c= A & A c= Y };
end;
```

An interval set of the form $[A, A]$ is equivalent to the set in an ordinary sense (but of course direct replacement is just erroneous). Furthermore, we gave the notion needed to characterize the carrier of the interval set algebra; its elements are all intervals.

```
definition let U;
  mode IntervalSet of U -> Subset-Family of U means
    ex A, B be Subset of U st it = Inter (A, B);
end;
```

[2] Due to the lack of space we usually drop proofs; they can be tracked in the full source.

The next part of the article is introducing operations on the interval sets:

```
definition let U be non empty set, A, B be non empty IntervalSet of U;
  func A _/\_ B -> IntervalSet of U equals
    INTERSECTION (A, B);
end;
```

(and similarly A _\/_ B, A __ B), where INTERSECTION is an ordinary Boolean operation taken componentwise. Equivalently, it was natural to give characterization of the aforementioned objects in terms of the operations on their bounds. Let us cite only the case of the difference of intervals.

```
theorem
  A _\_ B = Inter (A``1 \ B``2, A``2 \ B``1);
```

where A``1, A``2 denote the boundaries of interval set A.

Although the complementation operator is neither Boolean nor a pseudocomplement, it is definitely worth introducing (the symbols “[#]U” and “{}U”are introduced to add to the set U its proper type, i.e. a subset of itself).

```
definition let U be non empty set, A be non empty IntervalSet of U;
  func A ^ -> non empty IntervalSet of U equals :Def8:
    Inter ([#]U,[#]U) _\_ A;
end;
```

Obviously, an ordinary inclusion cannot be used as the ordering relation in the lattice of interval sets. Since the types of all objects are extended to the most general type set, the usual notation of set-theoretical inclusion (“c=”) could not be used.

```
definition let U be non empty set, A, B be non empty IntervalSet of U;
  pred A _c=_ B means
    A``1 c= B``1 & A``2 c= B``2;
end;
```

It is hardly the same relation, which can be illustrated by the fact that the identity $A \setminus B = \emptyset$ is true for arbitrary sets A, B such that $A \subseteq B$, but it is not the case of interval sets. Hence the following statement:

```
theorem
  ex A,B being non empty IntervalSet of U st
    A _c=_ B & A _\_ B <> Inter ({}U,{}U);
```

Of course, in the proof of this fact, the appropriate concrete example of two sets should have been constructed. After we have defined necessary binary operations on interval sets, the structure of the lattice of such sets (called InterLatt in our formalization) can be described – but let us omit the full citation here as we will provide it in the next section.

5 Lattices of Rough Sets

As many of the objects occuring in the MML base on the notion of a structure, the type "Lattice" is also the structure type (with the carrier and two binary operations). Underlying properties (commutativity, associativity, and absorption laws) are added to this radix type via six adjectives (attributes). The general lattice theory in the MML is formalized mainly according to Grätzer's *General Lattice Theory* and this development contains many standard results from this classical book. Also recent automatically obtained equational characterizations were translated into Mizar (as the Robbins problem about the alternative axiomatization of Boolean algebras, short single axioms for Boolean algebras based on the Sheffer stroke, ortholattice bases and so on).

Below we quote the definition of the lattice of rough sets. Its carrier consists of the set of all rough sets over an arbitrary but fixed tolerance approximation space X and the lattice operations are defined here elementwise.

```
definition let X be Tolerance_Space;
  func RSLattice X -> strict LattStr means
    the carrier of it = RoughSets X &
    for A, B being Element of RoughSets X,
        A', B' being RoughSet of X st A = A' & B = B' holds
     (the L_join of it).(A,B) = A' _\/_ B' &
     (the L_meet of it).(A,B) = A' _/\_ B';
end;
```

A similar definition of the interval set algebra `InterLatt` differs only in the case of carrier – we decided to have binary operations, although on various universes, encoded under the same symbols. Furthermore, it can be defined over an arbitrary non-empty set. We have proved formally that the lattice of rough sets is distributive and complete, it has also the lower and the upper bound. We decided for the binary operation approach because in this way the ordering is defined automatically[3] which makes this approach somewhat stronger.

Also, after we have defined

```
definition let X;
  func RoughIso X -> Homomorphism of RSLattice X,
                         InterLatt the carrier of X means
    for x being Element of RSLattice X holds
      it.x = [x`1, x`2];
  correctness;
end;
```

as the homomorphism (under `correctness` conditions we had to prove that `RoughIso` preserves suprema and infima, the existence and the uniqueness of such a mapping) between both structures (note that we forget in some sense about abstract structure connected with the lattices of rough sets, i.e. about the indiscernibility relation – we are interested only on the subsets of the carrier of X), usual properties can be proved only for the one of these objects.

[3] Recall that a poset to be a lattice should meet some additional requirements.

6 Learning Rough Sets

Rough sets definitely proved its feasibility and usefulness in various fields of the engineering (also biology or medicine, here especially systems as RSES and Rosetta are potentially useful), however they can be successfully used not only in the industry, but also in purely academic environment. For two consecutive years now (2005/2006 and 2006/2007), a group of students in University of Białystok (Poland) trained their ability in the field of reasoning on rough sets based on the Mizar proof-assistant. The main advantages of the application of this machine proof-checker seem to be the following:

Automatic verification – the results are checked without teacher's help, so very objective ("prove until computer will report no errors, I won't complain"); once the student knows the language, he/she can write syntactically correct texts (if not – computer parser points out the errors); once he/she learned the semantics – he/she tries to formulate the facts and prove them by him/herself;

Self-study enabled – the exercises are the best way for a student to getting knowledge ("experience, not only doctrine");

Human-friendly approach – if the justification is right, the computer also suggests the possible improvements of a proof, so students can benefit in various ways;

Logical correctness – while learning rough sets also classical predicate logic is taught as a side-effect; the logical correctness is crucial, so the student continuously has to remember about the rules; learning proving tactics (direct and indirect proof) and basic rules (exemplification, generalization);

Similarity to the natural language – even if readable for the machine, the language is not that artificial; so it is relatively not much time to get the syntax right;

Multi-purpose systems – rethinking and showing counterexamples is possible – e.g., develop concrete example of an approximation space one of the inclusions is not valid;

Real-life applications – verification of data – we can apply rough set methods to concrete systems; this is probably most difficult – the analysis of even small portions of data could be highly time-consuming;

Distance learning approach – the geographical diversity of learners is no problem any longer.

Note that for obvious reasons in a proof checker illustrative features of figures (e.g. proof suggestions based on diagrams) are not available.

7 Conclusions and Further Work

A kind of computer certification of mathematical proofs seems to become an important issue in the contemporary science. We believe that the use of the Mizar system can be attractive for mathematicians and the development of RST in it is really feasible, although of rather challenging character due to the broad nature of the discipline. We tried to show on selected examples of the development of the lattice of rough/interval sets that the formalization of this theory can be continued not paying the high price for raising duplicate notions from scratch, but via reusing the existing formal apparatus yet available in the MML.

As the future work, we may point out the formalization of [3] we started some time ago. We are interested in further development of the lattice-theoretical approach to the notion of RST (as in [8]) which are influential even in the broader algebraic sense [2] as well as in extending rough set model as [13]. Thanks to the structure of interval algebras we could obtain some direct correspondence with the described lattice of rough sets, so the results can easily be exchanged and proven only in one of both cases. We plan also to provide another axiomatic base for the MML, more RST-specific [1]. There is no doubt that logical foundations are harder to change (Mizar is based on the classical logic). On the other hand, we hope that rough sets techniques can be applied to improve the MML Query searching engine.

References

1. Bryniarski, E.: Formal conception of rough sets. Fundamenta Informaticae 27(2–3), 109–136 (1996)
2. Düntsch, I., Winter, M.: Construction of Boolean contact algebras. AI Communications 13, 235–246 (2004)
3. Gomolińska, A.: A comparative study of some generalized rough approximations. Fundamenta Informaticae 51(1–2), 103–119 (2002)
4. Grabowski, A.: On the computer-assisted reasoning about rough sets. In: Dunin-Kęplicz, B., et al. (ed.) Monitoring, Security, and Rescue Techniques in Multiagent Systems, Advances in Soft Computing, pp. 215–226. Springer, Heidelberg (2005)
5. Grabowski, A., Schwarzweller, Ch.: Rough Concept Analysis – theory development in the Mizar system. In: Asperti, A., Bancerek, G., Trybulec, A. (eds.) MKM 2004. LNCS, vol. 3119, pp. 130–144. Springer, Heidelberg (2004)
6. Iwiński, T.B.: Algebraic approach to rough sets. Bull. Pol. Acad. Sci. Math. 35, 673–683 (1987)
7. Järvinen, J.: Approximations and rough sets based on tolerances. In: Ziarko, W., Yao, Y. (eds.) RSCTC 2000. LNCS (LNAI), vol. 2005, pp. 182–189. Springer, Heidelberg (2001)
8. Järvinen, J.: Ordered set of rough sets. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 49–58. Springer, Heidelberg (2004)
9. Pawlak, Z.: Rough Sets. International Journal of Information and Computer Science 11, 341–356 (1982)
10. Pomykała, J.A.: About tolerance and similarity relations in information systems. In: Alpigini, J.J., Peters, J.F., Skowron, A., Zhong, N. (eds.) RSCTC 2002. LNCS (LNAI), vol. 2475, pp. 175–182. Springer, Heidelberg (2002)
11. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27(2–3), 245–253 (1996)
12. Yao, Y.Y.: Two views of the theory of rough sets in finite universes. International Journal of Approximation Reasoning 15(4), 291–317 (1996)
13. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences 46(1), 39–59 (1993)

Certain, Generalized Decision, and Membership Distribution Reducts Versus Functional Dependencies in Incomplete Systems[⋆]

Marzena Kryszkiewicz

Institute of Computer Science, Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
mkr@ii.pw.edu.pl

Abstract. An essential notion in the theory of Rough Sets is a reduct, which is a minimal set of conditional attributes that preserves a required classification feature, e.g. respective values of an original or modified decision attribute. Certain decision reducts, generalized decision reducts, and membership distribution reducts belong to basic types of Rough Sets reducts. In our paper, we prove that reducts of these types are sets of conditional attributes functionally determining respective modifications of a decision attribute both in complete and incomplete information systems. However, we also prove that, unlike in the case of complete systems, the reducts in incomplete systems are not guaranteed to be minimal sets of conditional attributes that functionally determine respective modifications of the decision attribute.

1 Introduction

Rough Sets theory defines reducts in a decision table as minimal sets of conditional attributes preserving the required classification feature [10]. The research devoted to reducts referred mostly to complete systems in which all attribute values were known. In this paper, we first revisit the results for certain decision, generalized decision, and membership distribution reducts, which belong to basic types of Rough Set reducts. Next, we examine properties of reducts of these types in incomplete systems in which values of attributes may be missing. As a result, we prove that reducts of these types are sets of conditional attributes functionally determining respective modifications of a decision attribute both in complete and incomplete information systems. However, we also prove that, unlike in the case of complete systems, the reducts in incomplete systems are not guaranteed to be minimal sets of conditional attributes that functionally determine respective modifications of the decision attribute.

The layout of the paper is as follows: In Section 2, we recall basic Rough Set notions and provide their properties. A notion of a functional dependency is recalled in Section 3. In Section 4, we systematically revisit the relationship

[⋆] Research has been supported by grant No 3 T11C 002 29 received from Polish Ministry of Education and Science.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 162–174, 2007.

between functional dependencies and generalized decision reducts, membership distribution reducts, and certain decision reducts in complete decision tables. The main part of our contribution is presented in Section 5, where we examine this relationship in incomplete decision tables. In Section 6, we conclude our results.

2 Basic Notions and Properties of Rough Sets

2.1 Information Systems

An *information system* (*IS*) is a pair $S = (O, AT)$, where O is a non-empty finite set of *objects* and AT is a non-empty finite set of *attributes*, such that $a : O \rightarrow V_a$ for any $a \in AT$, where V_a is called *domain* of the attribute a. Each subset of attributes $A \subseteq AT$ determines a binary *A-indiscernibility relation IND*(A),

$$IND(A) = \{(x, y) \in O \times O \mid \forall_{a \in A}\ a(x) = a(y)\}.$$

The relation $IND(A), A \subseteq AT$, is an equivalence relation and determines a partition of O, which will be denoted by π_A. Objects indiscernible with object x with regard to attribute set A in the system will be denoted by $I_A(x)$ and called *A-indiscernibility class*; that is, $I_A(x) = \{y \in O \mid (x, y) \in IND(A)\}$. Clearly, partition $\pi_A = \{I_A(x) \mid x \in O\}$.

Property 2.1.1 [10]. Let $A, B \subseteq AT$ and $x \in O$.
a) $A \subseteq B \Rightarrow I_B(x) \subseteq I_A(x)$
b) $I_{A \cup B}(x) = I_A(x) \cap I_B(x)$
c) $I_A(x) = \bigcap_{a \in A} I_a(x)$

Proposition 2.1.1. *Let $A \subseteq B \subseteq AT$ and $x \in O$. $I_A(x) = \bigcup_{y \in I_A(x)} I_B(y)$.*

Example 2.1.1. Table 1 describes a sample information system consisting of 10 objects and described by attributes $\{a, b, c, e, f, d\}$. Let $A = \{a, b\}$ and $B =$

Table 1. Sample *DT*

$x \in O$	a	b	c	e	f	d
1	1	0	0	1	1	1
2	1	1	1	1	2	1
3	0	1	1	0	3	1
4	0	1	1	0	3	2
5	0	1	1	2	2	2
6	1	1	0	2	2	2
7	1	1	0	2	2	3
8	1	1	0	2	2	3
9	1	1	0	3	2	3
10	1	0	0	3	2	3

Table 2. *DT* extended with d^{N}_{AT}, ∂_{AT}, μ^{AT}_d

$x \in O$	a	b	c	e	f	d	d^{N}_{AT}	∂_{AT}	$\mu^{AT}_d : < \mu^{AT}_1, \mu^{AT}_2, \mu^{AT}_3 >$
1	1	0	0	1	1	1	1	$\{1\}$	$< 1, 0, 0 >$
2	1	1	1	1	2	1	1	$\{1\}$	$< 1, 0, 0 >$
3	0	1	1	0	3	1	N	$\{1, 2\}$	$< 1/2, 1/2, 0 >$
4	0	1	1	0	3	2	N	$\{1, 2\}$	$< 1/2, 1/2, 0 >$
5	0	1	1	2	2	2	2	$\{2\}$	$< 0, 1, 0 >$
6	1	1	0	2	2	2	N	$\{2, 3\}$	$< 0, 1/3, 2/3 >$
7	1	1	0	2	2	3	N	$\{2, 3\}$	$< 0, 1/3, 2/3 >$
8	1	1	0	2	2	3	N	$\{2, 3\}$	$< 0, 1/3, 2/3 >$
9	1	1	0	3	2	3	3	$\{3\}$	$< 0, 0, 1 >$
10	1	0	0	3	2	3	3	$\{3\}$	$< 0, 0, 1 >$

$\{a, b, c, e, f\}$. $I_A(3) = \{3, 4, 5\}$, $I_B(3) = I_B(4) = \{3, 4\}$, $I_B(5) = \{5\}$. Hence, $I_A(3) = \{3, 4, 5\} = I_B(3) \cup I_B(4) \cup I_B(5)$ (see Proposition 2.1.1). □

Let $X \subseteq O$ and $A \subseteq AT$. AX is defined as an *A-lower approximation* of object set X, iff $\underline{A}X = \bigcup\{Y \in \pi_A \mid Y \subseteq X\}$ (or $\underline{A}X = \{x \in O \mid I_A(x) \subseteq X\}$). $\overline{A}X$ is defined as an *A-upper approximation* of X, iff $\underline{A}X = \bigcup\{Y \in \pi_A \mid Y \cap X \neq \emptyset\}$ (or $\overline{A}X = \{x \in O \mid I_A(x) \cap X \neq \emptyset\}$). $\underline{A}X$ is the set of objects that belong to X with certainty, while $\overline{A}X$ is the set of objects that possibly belong to X.

2.2 Decision Tables

A *decision table* is an information system $DT = (O, AT \cup \{d\})$, where $d \notin AT$ is a distinguished attribute called the *decision*, and the elements of AT are called *conditions*. A *decision class* is defined as the set of all objects with the same decision value. By X_{d_i} we will denote the decision class consisting of objects the decision value of which equals d_i, where $d_i \in V_d$. Clearly, for any object x in O, $I_d(x)$ is a decision class. DT is called *consistent* if for each $I_{AT}(x) \in \pi_{AT}$ there is $I_d(x) \in \pi_d$ such that $I_{AT}(x) \subseteq I_d(x)$. Otherwise, DT is called *inconsistent*.

Proposition 2.2.1. Let $A \subseteq AT$ and $x \in X \subseteq O$. $X \subseteq I_d(x)$ iff $\exists_{y \in O}\ X \subseteq I_d(y)$.

Proof. ($\Rightarrow$) Trivial.
($\Leftarrow$) Let y be an object in O such that $X \subseteq I_d(y)$ (*). Hence, $x \in X \subseteq I_d(y)$, so $x \in I_d(y)$. Thus, $d(x) = d(y)$ (**). By (*) and (**), $X \subseteq I_d(y) = I_d(x)$. □

An *A-positive region* (denoted by POS_A) in DT is defined as the union of the A-lower approximations of all decision classes, that is:

$$POS_A = \bigcup_{d_i \in V_d} \underline{A}X_{d_i}.$$

For $A = AT$, A-positive region is denoted briefly by POS.

Proposition 2.2.2. $POS_A = \{x \in O \mid I_A(x) \subseteq I_d(x)\}$.

Proof. $POS_A = \bigcup_{d_i \in V_d} \underline{A}X_{d_i} = \bigcup_{y \in O} \underline{A}I_d(y) = \bigcup_{y \in O} \{x \in O \mid I_A(x) \subseteq I_d(y)\} =$ /* by Proposition 2.2.1 */ $= \bigcup_{y \in O} \{x \in O \mid I_A(x) \subseteq I_d(x)\} = \{x \in O \mid I_A(x) \subseteq I_d(x)\}$. □

One can note that the positive region contains all objects in O about which we are certain that they belong to the decision classes determined by their decision values. An *A-negative region* (NEG_A) is defined as the set of all objects in O that do not belong to POS_A. In the sequel, NEG_{AT} will be denoted briefly by NEG. Clearly, DT is consistent iff $NEG = \emptyset$ (or $POS = O$).

For the sake of later use, we introduce a notion of an *A-derivable decision attribute* for an object $x \in O$, which we denote by $d_A^{\mathrm{N}}(x)$ and define as follows: $d_A^{\mathrm{N}}(x) = d(x)$ if $x \in POS_A$, and $d_A^{\mathrm{N}}(x) = \mathrm{N}$ otherwise. Clearly, all objects with value N of d_{AT}^{N} belong to NEG; all other objects belong to POS.

The notion of the negative region may be too vague in some applications. Looking at Table 1, one may note that objects 3 and 4, which are indiscernible with respect to $AT = \{a, b, c, e, f\}$, may belong to the decision classes X_{d_1} or X_{d_2}, but certainly do not belong the decision class X_{d_3}.

A notion of a generalized decision allows us to specify this knowledge. An A *generalized decision* for object x in DT (denoted by $\partial_A(x)$), $A \subseteq AT$, is defined as the set of all decision values of all objects indiscernible with x on A; i.e. [13]:

$$\partial_A(x) = \{d(y) \mid y \in I_A(x)\}.$$

Property 2.2.1. Let $x \in O$ and $A, B \subseteq AT$. If $A \subseteq B$, then $\partial_B(x) \subseteq \partial_A(x)$.

For $A = AT$, an A-*generalized decision* will be also called briefly a *generalized decision*. The generalized decision informs on decision classes to which an object may belong. One may additionally be interested in the degree in which the objects may belong to these classes. An A-*membership function*: $\mu^A_{d_i} : O \rightarrow [0, 1]$, $A \subseteq AT$, is defined as follows [15]:

$$\mu^A_{d_i}(x) = \frac{\mid I_A(x) \cap X_{d_i} \mid}{\mid I_A(x) \mid}.$$

An A-*membership distribution function*: $\mu^A_d : O \rightarrow [0, 1]^n$, $A \subseteq AT, n = \mid V_d \mid$, is defined as follows [15]:

$$\mu^A_d(x) = (\mu^A_{d_1}(x), \ldots, \mu^A_{d_n}(x)), \text{where } \{d_1, \ldots, d_n\} = V_d.$$

The values of the derivable decision attribute, generalized decision and membership distribution function for objects in DT from Table 1 are shown in Table 2.

2.3 Certain Decision, Generalized Decision, and Membership Distribution Reducts

A *reduct* is an essential notion in the Rough Set theory. In this paper, we will focus on three types of reducts, namely, on certain decision, generalized decision, and membership distribution reducts. Below, we recall their definitions:

A set of attributes $A \subseteq AT$ is a *certain decision reduct* of DT iff A is a minimal set such that

$$\forall_{x \in POS}\ I_A(x) \subseteq I_d(x).$$

$A \subseteq AT$ is a *generalized decision reduct* of DT iff A is a minimal set such that

$$\forall_{x \in O}\ \partial_A(x) = \partial_{AT}(x).$$

$A \subseteq AT$ is a μ-*decision reduct* (or *membership distribution reduct*) of DT iff A is a minimal set such that

$$\forall_{x \in O}\ \mu^A_d(x) = \mu^{AT}_d(x).$$

In general, for each certain decision reduct A, there is a superset of A which is a generalized decision reduct, and for each generalized decision reduct B, there is a superset of B which is a μ-decision reduct [6],[7]. In the Rough Set literature, one can also find definitions of other types of reducts. To the most important ones, we did not introduce, belong possible, approximate and μ-reducts. It has been proved in [6],[7] that the set of possible reducts as well as the set of approximate reducts equals the set of generalized decision reducts, and the set of μ-reducts of DT equals the set of μ-decision reducts. These and other types of reducts were also discussed e.g. in [1],[8-19].

3 Functional Dependencies

Let A and B be sets of attributes in an information system. $A \rightarrow B$ is defined a *functional dependency* (or A is defined to *determine B functionally*) if $\forall_{x \in O}\ I_A(x) \subseteq I_B(x)$. $A \rightarrow B$ is defined a *minimal functional dependency* if it is a functional dependency and $\forall_{C \subset A}\ C \rightarrow B$ is not a functional dependency.

Example 3.1. Let us consider the information system in Table 1. $\{ce\} \rightarrow \{a\}$ is a functional dependency, nevertheless, $\emptyset \rightarrow \{a\}$, $\{c\} \rightarrow \{a\}$ and $\{e\} \rightarrow \{a\}$ are not. Hence, $\{ce\} \rightarrow \{a\}$ is a minimal functional dependency. □

4 Reducts and Minimal Functional Dependencies

In this section, we prove that generalized decision, membership distribution, and certain decision reducts are minimal sets of conditional attributes in decision table DT which functionally determine the generalized decision ∂_{AT}, membership distribution μ_d^{AT}, and derivable decision attribute $d_{AT}^{\mathrm{N}}(x)$, respectively.

4.1 Generalized Decision Reducts and Minimal Functional Dependencies

Since generalized decision reducts are based on the notion of a generalized decision, we first examine the relationship between this notion and a functional dependency.

Lemma 4.1.1. Let $A \subseteq AT$. The following statements are equivalent:
a) $\forall_{x \in O}\ \partial_A(x) = \partial_{AT}(x)$
b) $\forall_{x \in O}\ \forall_{y \in I_A(x)} \partial_{AT}(y) = \partial_{AT}(x)$
c) $\forall_{x \in O}\ I_A(x) \subseteq I_{\partial_{AT}}(x)$
d) $A \rightarrow \{\partial_{AT}\}$ is a functional dependency

Proof. Ad a $\Rightarrow$ b) (by contradiction). Let $\forall_{z \in O}\ \partial_A(z) = \partial_{AT}(z)$ (*), $x \in O$, $y \in I_A(x)$ (**) and $\partial_{AT}(y) \neq \partial_{AT}(x)$. By (*), $\partial_A(x) = \partial_{AT}(x)$, $\partial_A(y) = \partial_{AT}(y)$, and by (**), $\partial_A(x) = \partial_A(y)$. Hence, $\partial_{AT}(x) = \partial_A(x) = \partial_A(y) = \partial_{AT}(y)$. Thus, we conclude, $\partial_{AT}(x) = \partial_{AT}(y)$, which contradicts the assumption. Ad a $\Leftarrow$ b) Let $x \in O$ and $\forall_{y \in I_A(x)}\ \partial_{AT}(y) = \partial_{AT}(x)$ (*). $\partial_A(x) = \bigcup_{y \in I_A(x)} \{d(y)\}$ $\subseteq$ /* $d(y) \in \partial_{AT}(y)$ for any object y */ $\bigcup_{y \in I_A(x)} \partial_{AT}(y)$ = /* by (*) */ = $\bigcup_{y \in I_A(x)}\ \partial_{AT}(x) = \partial_{AT}(x)$. Hence, $\partial_A(x) \subseteq \partial_{AT}(x)$ (**). On the other hand,

by Property 2.1.1, $\partial_{AT}(x) \subseteq \partial_A(x)$ (***). By (**) and (***), we conclude, $\partial_A(x) = \partial_{AT}(x)$.
Ad b $\Leftrightarrow$ c $\Leftrightarrow$ d) Trivial. □

Proposition 4.1.1. $AT \rightarrow \{\partial_{AT}\}$ is a functional dependency.

Proof. The formula $\forall_{x\in O}\ \partial_{AT}(x) = \partial_{AT}(x)$ is trivially true. Hence, and by Lemma 4.1.1a,d, $AT \rightarrow \{\partial_{AT}\}$ is a functional dependency. □

Theorem 4.1.1. Let $A \subseteq AT$. A is a generalized decision reduct of DT iff $A \rightarrow \{\partial_{AT}\}$ is a minimal functional dependency.

Proof. A is a generalized decision reduct of DT iff /* by definition of a generalized decision reduct /* $\forall_{x\in O}\ \partial_A(x) = \partial_{AT}(x)$ and there is no proper subset $C \subset A$ such that $\forall_{x\in O}\ \partial_C(x) = \partial_{AT}(x)$ iff /* by Lemma 4.1.1a,d /* $A \rightarrow \{\partial_{AT}\}$ is functional and there is no proper subset $C \subset A$ such that $C \rightarrow \{\partial_{AT}\}$ is functional iff $A \rightarrow \{\partial_{AT}\}$ is a minimal functional dependency. □

Theorem 4.1.1 corresponds to the result obtained in [13].

4.2 μ-Decision Reducts and Minimal Functional Dependencies

As μ-decision reducts are based on the notion of a membership distribution function, we first examine the relationship between this notion and a functional dependency.

Lemma 4.2.1. Let $A \subseteq AT$. The following statements are equivalent:
a) $\forall_{x\in O}\ \mu_d^A(x) = \mu_d^{AT}(x)$
b) $\forall_{x\in O}\ \forall_{y\in I_A(x)}\ \mu_d^{AT}(y) = \mu_d^{AT}(x)$
c) $\forall_{x\in O}\ I_A(x) \subseteq I_{\mu_d^{AT}}(x)$
d) $A \rightarrow \{\mu_d^{AT}\}$ is a functional dependency

Proof. Ad a $\Rightarrow$ b) (by contradiction). Let $\forall_{z\in O}\ \mu_d^A(z) = \mu_d^{AT}(x)$ (*), $x \in O$, $y \in I_A(x)$ (**) and $\mu_d^{AT}(y) \neq \mu_d^{AT}(x)$. By (*), $\mu_d^A(x) = \mu_d^{AT}(x)$, $\mu_d^A(y) = \mu_d^{AT}(y)$, and by (**), $\mu_d^A(x) = \mu_d^A(y)$. Hence, $\mu_d^{AT}(x) = \mu_d^A(x) = \mu_d^A(y) = \mu_d^{AT}(y)$. Thus, we conclude, $\mu_d^{AT}(x) = \mu_d^{AT}(y)$, which contradicts the assumption. Ad a $\Leftarrow$ b) Let $x \in O$ and $\forall_{y\in I_A(x)}\ \mu_d^{AT}(y) = \mu_d^{AT}(x)$ (or equivalently, $\mu_{d_i}^{AT}(y) = \mu_{d_i}^{AT}(x)$ for all $d_i \in V_d$) (*). Let d_i be an arbitrary decision value in V_d, $\mu_{d_i}^{AT}(x) = \varepsilon$, and $I_A(x) = I_1 \cup \ldots I_l$, where $I_1, \ldots, I_l$ are distinct (mutually exclusive) classes in π_{AT}. Clearly, for each class $I_j, j = 1..l$, there is an object $y \in I_A(x)$ such that $I_j = I_{AT}(y)$ and $\mid I_j \cap X_{d_i} \mid / \mid I_j \mid = \mid I_{AT}(y) \cap X_{d_i} \mid / \mid I_{AT}(y) \mid = \mu_{d_i}^{AT}(y) =$ /* by (*) */ $= \mu_{d_i}^{AT}(x)$. Hence, $\forall_{j=1..l} \mid I_j \cap X_{d_i} \mid / \mid I_j \mid = \mu_{d_i}^{AT}(x) = \varepsilon$, so $\forall_{j=1..l} \mid I_j \cap X_{d_i} \mid = \varepsilon \times \mid I_j \mid$ (**). Now, $\mu_{d_i}^A(x) = \mid I_A(x) \cap X_{d_i} \mid / \mid I_A(x) \mid$ $= \mid (\bigcup_{j=1..l} I_j) \bigcap X_{d_i} \mid / \mid \bigcup_{j=1..l} I_j \mid = (\sum_{j=1..l} \mid I_j \cap X_{d_i} \mid) / (\sum_{j=1..l} \mid I_j \mid) =$ $(\sum_{j=1..l} \varepsilon \times \mid I_j \mid) / (\sum_{j=1..l} \mid I_j \mid) = \varepsilon = \mu_{d_i}^{AT}(x)$. Hence, $\mu_{d_i}^A(x) = \mu_{d_i}^{AT}(x)$ (**).

As d_i was chosen arbitrarily, we may generalize (**) for all values d_i in V_d. In consequence, we conclude, $\mu_d^A(x) = \mu_d^{AT}(x)$.
Ad b $\Leftrightarrow$ c $\Leftrightarrow$ d) Trivial. □

Proposition 4.2.1. $AT \rightarrow \{\mu_d^{AT}\}$ is a functional dependency.

Proof. Analogical to the proof of Proposition 4.1.1; by Lemma 4.2.1a,d. □

Theorem 4.2.1. Let $A \subseteq AT$. A is a μ-decision reduct of DT iff $A \rightarrow \{\mu_d^{AT}\}$ is a minimal functional dependency.

Proof. Analogical to the proof of Theorem 4.1.1; follows from the definitions of a μ-decision reduct and minimal functional dependency, and Lemma 4.2.1a,d. □

Theorem 4.2.1 corresponds to the result reported in [16].

4.3 Certain Decision Reducts and Minimal Functional Dependencies

Certain decision reducts preserve the positive region. Let us thus start with investigating the consequences of (non-) belonging to POS.

Property 4.3.1. Let $x \in O$. The following statements are equivalent:
a) $x \in POS$
b) $I_{AT}(x) \subseteq I_d(x)$
c) $I_{AT}(x) \subseteq POS$

Proof. Ad (a ⇔ b) By Proposition 2.2.2.
Ad (a ⇒ c) Let $x \in POS$. Then by Proposition 2.2.2, $I_{AT}(x) \subseteq_d (x)$ (*). Since $\forall_{y \in I_{AT}(x)}\ I_{AT}(y) = I_{AT}(x)$, then (*) can be rewritten as $\forall_{y \in I_{AT}(x)}\ I_{AT}(y) \subseteq I_d(x)$. Hence, by Proposition 2.2.1, $\forall_{y \in I_{AT}(x)}\ I_{AT}(y) \subseteq I_d(y)$. Thus, by Proposition 2.2.2, $\forall_{y \in I_{AT}(x)}\ y \in POS$, so $I_{AT}(x) \subseteq POS$.
Ad (a ⇐ c) Trivial. □

Property 4.3.2. Let $x \in O$. The following statements are equivalent:
a) $x \notin POS$
b) $I_{AT}(x) \not\subseteq I_d(x)$
c) $I_{AT}(x) \subseteq O \backslash POS$

Proof. Ad (a ⇔ b) Follows from Property 4.3.1.
Ad (b ⇒ c) Let $I_{AT}(x) \not\subseteq I_d(x)$. Then, by Proposition 2.2.1, $\neg\exists_{y \in O}\ I_{AT}(x) \subseteq I_d(y)$. Hence, $\forall_{y \in I_{AT}(x)}\ I_{AT}(x) \not\subseteq I_d(y)$. Since $\forall_{y \in I_{AT}(x)}\ I_{AT}(y) = I_{AT}(x)$, then $\forall_{y \in I_{AT}(x)}\ I_{AT}(y) \not\subseteq I_d(y)$. Thus by Property 4.3.1, $\forall_{y \in I_{AT}(x)}\ y \in O \backslash POS$. Therefore, $I_{AT}(x) \subseteq O \backslash POS$.
Ad (b ⇐ c) Let $I_{AT}(x) \subseteq O \backslash POS$. Hence, $I_{AT}(x) \not\subseteq POS$. Then, by Property 4.3.1, $I_{AT}(x) \not\subseteq I_d(x)$. □

By Property 4.3.1, if object x belongs to POS, then AT-indiscernibility class of this object is contained in POS, and all objects in this class have the same decision value as x does. By Property 4.3.2, if x does not belong to POS, then AT-indiscernibility class of this object is contained in the negative region.

Lemma 4.3.1. Let $A \subseteq AT$ and $\forall_{y \in O}\ I_{AT}(y) \subseteq I_d(y) \Rightarrow I_A(y) \subseteq I_d(y)$. Then:
a) $\forall_{x \in O}\ I_{AT}(x) \subseteq I_d(x) \Rightarrow I_A(x) \subseteq POS$
b) $\forall_{x \in POS}\ I_A(x) \subseteq POS$
c) $\forall_{x \in O}\ I_{AT}(x) \not\subseteq I_d(x) \Rightarrow I_A(x) \subseteq O \backslash POS$
d) $\forall_{x \in O \backslash POS}\ I_A(x) \subseteq O \backslash POS$

Proof. Let $A \subseteq AT$ and $\forall_{y \in O}\ I_{AT}(y) \subseteq I_d(y) \Rightarrow I_A(y) \subseteq I_d(y)$ $(^*)$.
Ad a) Let x be an object such that $I_{AT}(x) \subseteq I_d(x)$. By Proposition 2.1.1 and $(^*)$ we conclude, $\bigcup_{y \in I_A(x)}\ I_{AT}(y) = I_A(x) \subseteq I_d(x)$. Hence and by Proposition 2.2.1, $\forall_{y \in I_A(x)}\ I_{AT}(y) \subseteq I_d(y)$. Thus, by Property 4.3.1, $\forall_{y \in I_A(x)}\ I_{AT}(y) \subseteq POS$. Having this result in mind and taking into account Proposition 2.1.1, we conclude $I_A(x) = \bigcup_{y \in I_A(x)}\ I_{AT}(y) \subseteq POS$.

Ad b) Follows from Lemma 4.3.1a and Property 4.3.1.
Ad c) (by contradiction). Let x be an object such that $I_{AT}(x) \not\subseteq I_d(x)$ $(^{**})$ and $I_A(x) \not\subseteq O \backslash POS$. By Proposition 2.1.1, we conclude: $\bigcup_{y \in I_A(x)}\ I_{AT}(y) \not\subseteq O \backslash POS$. Hence, $\exists_{y \in I_A(x)}\ y \in POS$. Thus, by Property 4.3.1, $\exists_{y \in I_A(x)}\ I_{AT}(y) \subseteq I_d(y)$. By $(^*)$ we conclude: $\exists_{y \in I_A(x)}\ I_A(y) \subseteq I_d(y)$. Since, $I_A(y) = I_A(x)$ for any $y \in I_A(x)$, then we may infer $\exists_{y \in I_A(x)}\ I_A(x) \subseteq I_d(y)$. Now, by Proposition 2.2.1, we may derive, $I_A(x) \subseteq I_d(x)$. Since $I_{AT}(x) \subseteq I_A(x)$ (by Property 2.1.1a), we conclude, $I_{AT}(x) \subseteq I_d(x)$. This contradicts the assumption $(^{**})$.
Ad d) Follows from Lemma 4.3.1c and Property 4.3.2. □

Lemma 4.3.2. Let $A \subseteq AT$. The following statements are equivalent:
a) $\forall_{x \in POS}\ I_A(x) \subseteq I_d(x)$
b) $\forall_{x \in O}\ I_A(x) \subseteq I_{d^{\mathrm{N}}_{AT}}(x)$
c) $A \rightarrow \{d^{\mathrm{N}}_{AT}\}$ is a functional dependency

Proof. Ad a $\Rightarrow$ b) Let $\forall_{x \in POS}\ I_A(x) \subseteq I_d(x)(^*)$. Hence, by Property 4.3.1, $\forall_{x \in O}\ I_{AT}(x) \subseteq I_d(x) \Rightarrow I_A(x) \subseteq I_d(x)$. Thus, by Lemma 4.3.1d, $\forall_{x \in O \backslash POS}\ I_A(x) \subseteq O \backslash POS$ $(^{**})$. Since $d^{\mathrm{N}}_{AT}(x) = \mathrm{N}$ for all and only $x \in O \backslash POS$, then $\forall_{x \in O \backslash POS}\ I_{d^{\mathrm{N}}_{AT}}(x) = O \backslash POS$. Hence, $(^{**})$ can be rewritten as $\forall_{x \in O \backslash POS}\ I_A(x) \subseteq I_{d^{\mathrm{N}}_{AT}}(x)$ $(^{***})$. In addition, since $d^{\mathrm{N}}_{AT}(x) = d(x)$ for $x \in POS$, then $(^*)$ can be rewritten as $\forall_{x \in POS}\ I_A(x) \subseteq I_{d^{\mathrm{N}}_{AT}}(x)(^{****})$. Thus, by $(^{***})$ and $(^{****})$, $\forall_{x \in O}\ I_A(x) \subseteq I_{d^{\mathrm{N}}_{AT}}(x)$.
Ad a $\Leftarrow$ b) Let $\forall_{x \in O}\ I_A(x) \subseteq I_{d^{\mathrm{N}}_{AT}}(x)$. Then, by definition of d^{N}_{AT}, $\forall_{x \in POS}\ I_A(x) \subseteq I_{d^{\mathrm{N}}_{AT}}(x) = I_{d_{AT}}(x)$.
Ad b $\Leftrightarrow$ c) Trivial. □

Having in mind properties of the positive region (Proposition 2.2.2), definition of a certain decision reduct and Lemma 4.3.2, we offer Proposition 4.3.1 and Theorem 4.3.1, in which we express the relationship between certain decision reducts and functional dependencies.

Proposition 4.3.1. $AT \rightarrow \{d^{\mathrm{N}}_{AT}\}$ is a functional dependency.
Proof. By Proposition 2.2.2, $\forall_{x \in POS}\ I_{AT}(x) \subseteq I_d(x)$. Hence and by Lemma 4.3.2a,c, $AT \rightarrow \{d^{\mathrm{N}}_{AT}\}$ is a functional dependency. □

Theorem 4.3.1. Let $A \subseteq AT$ A is a certain decision reduct of DT iff $A \rightarrow \{d^{\mathrm{N}}_{AT}\}$ is a minimal functional dependency.

Proof. Analogical to the proof of Theorem 4.1.1; follows from the definition of a certain decision reduct, definition of a minimal functional dependency and Lemma 4.3.2a,c. □

Theorem 4.3.1 corresponds to the result presented in [14].

5 Reducts and Functional Dependencies Under Incompleteness

It may happen that some of attribute values for an object are missing in an information system. The system in which values of all attributes for all objects from O are known is called *complete*, otherwise it is called *incomplete*. Further on, we will denote missing value by *. We will also assume that an object $x \in O$ possesses exactly one value for each attribute in AT, in reality. Thus, if the value of an attribute a is missing, then we conclude that the real value is one from the set $V_a \backslash \{^*\}$. Hence, an object with $a(x) = {}^*$ is likely to be $\{a\}$-indiscernible in reality with all other objects in O. The indiscernibility relation, nevertheless, would treat this object as indiscernible only with objects for which the value of attribute a is unknown, which seems incorrect. In [2-5], we have introduced and discussed a notion of a similarity relation in order to deal with the incompleteness. In this section, we examine the dependency between similarity-based certain decision, generalized decision, and μ-decision reducts and respective modification of the decision attribute.

5.1 Basic Notions Under Incompleteness

In Section 5, we consider an incomplete decision table $IDT = (O, AT \cup \{d\})$ that admits unknown values only for attributes in AT. A *similarity relation* wrt. $A \subseteq AT$ is denoted by $SIM(A)$, and is defined as follows:

$$SIM(A) = \{(x, y) \in O \times O \mid \forall_{a \in A}\ a(x) = a(y) \text{ or } a(x) = {}^* \text{ or } a(y) = {}^*\}.$$

The similarity relation is reflexive and symmetric, but may not be transitive. The set of objects similar with object x wrt. attribute set A in IDT is denoted by $S_A(x)$ and called *A-similarity class*; that is, $S_A(x) = \{y \in O \mid (x, y) \in SIM(A)\}$.

Example 5.1.1. Table 3 presents a sample incomplete decision table $IDT = (O, AT \cup \{d\})$, where $AT = \{a, b\}$. The similarity classes of objects 1 and 5 wrt. AT, $\{b\}$, and $\emptyset$ are as follows: $S_{AT}(1) = \{1\}, S_{\{b\}}(1) = \{1, 5\}$, $S_\emptyset(1) = \{1, 2, 3, 4, 5, 6, 7, 8\}$, $S_{AT}(5) = \{5, 6\}$, $S_{\{b\}}(5) = S_\emptyset(5) = \{1, 2, 3, 4, 5, 6, 7, 8\}$. □

Table 3. $IDT = (O, AT \cup \{d\})$, where $AT = \{a, b\}$, extended with modified decisions

$x \in O$	a	b	d	d^{N}_{AT}	$d^{\mathrm{N}}_{\{b\}}$	$d^{\mathrm{N}}_{\emptyset}$	∂_{AT}	$\partial_{\{b\}}$	$\partial_\emptyset$	μ_d^{AT}	$\mu_d^{\{b\}}$	$\mu_d^\emptyset$
1	1	1	1	1	N	N	$\{1\}$	$\{1,3\}$	$\{1,2,3\}$	$< 1,0,0 >$	$< 1/2,0,1/2 >$	$< 1/8,2/8,5/8 >$
2	2	3	2	N	N	N	$\{2,3\}$	$\{2,3\}$	$\{1,2,3\}$	$< 0,2/3,1/3 >$	$< 0,2/4,2/4 >$	$< 1/8,2/8,5/8 >$
3	2	3	2	N	N	N	$\{2,3\}$	$\{2,3\}$	$\{1,2,3\}$	$< 0,2/3,1/3 >$	$< 0,2/4,2/4 >$	$< 1/8,2/8,5/8 >$
4	2	3	3	N	N	N	$\{2,3\}$	$\{2,3\}$	$\{1,2,3\}$	$< 0,2/3,1/3 >$	$< 0,2/4,2/4 >$	$< 1/8,2/8,5/8 >$
5	3	*	3	3	N	N	$\{3\}$	$\{1,2,3\}$	$\{1,2,3\}$	$< 0,0,1 >$	$< 1/8,2/8,5/8 >$	$< 1/8,2/8,5/8 >$
6	3	4	3	3	3	N	$\{3\}$	$\{3\}$	$\{1,2,3\}$	$< 0,0,1 >$	$< 0,0,1 >$	$< 1/8,2/8,5/8 >$
7	4	5	3	3	3	N	$\{3\}$	$\{3\}$	$\{1,2,3\}$	$< 0,0,1 >$	$< 0,0,1 >$	$< 1/8,2/8,5/8 >$
8	5	6	3	3	3	N	$\{3\}$	$\{3\}$	$\{1,2,3\}$	$< 0,0,1 >$	$< 0,0,1 >$	$< 1/8,2/8,5/8 >$

Property 5.1.1. Let $A, B \subseteq AT$ and $x \in O$.
a) $I_A(x) \subseteq S_A(x)$
b) $\forall_{y \in I_A(x)}\ S_A(y) = S_A(x)$
c) $A \subseteq B \Rightarrow S_B(x) \subseteq S_A(x)$

In Table 4, we provide definitions of similarity-based Rough Sets notions, we use throughout this section. Table 3 illustrates d_A^{N}, ∂_A, and μ_d^A, where $A \subseteq AT$.

Let $A \subseteq AT$. A is a *certain decision reduct* of IDT iff A is a minimal set such that $\forall_{x \in POS}\ S_A(x) \subseteq I_d(x)$. A is a *generalized decision reduct* of IDT iff A is a minimal set such that $\forall_{x \in O}\ \partial_A(x) = \partial_{AT}(x)$. A is a *μ-decision reduct* of IDT iff A is a minimal set such that $\forall_{x \in O}\ \mu_d^A(x) = \mu_d^{AT}(x)$.

The definition of a (minimal) functional dependency in an incomplete system remains the same as in the case of a complete system (see Section 3).

Table 4. Similarity based Rough Sets notions

notion	definition	notion	definition
$\underline{A}X$	$\{x \in O \mid S_A(x) \subseteq X\}$;	$d_A^{\mathrm{N}}(x)$	$d(x)$ if $x \in POS_A$, and N otherwise;
$\overline{A}X$	$\{x \in O \mid S_A(x) \cap X \neq \emptyset\}$;	$\partial_A(x)$	$\{d(y) \mid y \in S_A(x)\}$;
POS_A	$\underline{A}X_{d_1} \cup \ldots \cup \underline{A}X_{d_n}$;	$\mu {d_i}^A(x)$	$\mid S_A(x) \cap X_{d_i} \mid / \mid S_A(x) \mid$;
POS	POS_{AT};	$\mu_d^A(x)$	$(\mu_{d_1}^A(x), \ldots, \mu_{d_n}^A(x))$.

5.2 Generalized Decision Reducts and Functional Dependencies Under Incompleteness

Lemma 5.2.1. Let $A \subseteq AT$ and $x \in O$. $\forall_{y \in I_A(x)}\ \partial_A(y) = \partial_A(x)$.

Proof. Let $y \in I_A(x)$. By definition, $\partial_A(y) = \{d(z) \mid z \in S_A(y)\} =$ /* by Property 5.1.1b */ $= \{d(z) \mid z \in S_A(x)\} = \partial_A(x)$. □

Proposition 5.2.1. Let $A \subseteq AT$. $A \rightarrow \{\partial_A\}$ is functional in IDT.

Proof. Follows immediately from Lemma 5.2.1. □

Proposition 5.2.2. Let $A \subseteq AT$. If A is a generalized decision reduct of IDT, then $A \rightarrow \{\partial_{AT}\}$ is a functional dependency in IDT.

Proof. Let A be a generalized decision reduct of IDT. Then $\forall_{x \in O}\ \partial_A(x) = \partial_{AT}(x)$ and, by Proposition 5.2.1, $A \rightarrow \{\partial_A\}$ is a functional dependency in IDT. Hence, $A \rightarrow \{\partial_{AT}\}$ is a functional dependency in IDT. □

According to Proposition 5.2.2, we observe in IDT from Table 3 that $AT \rightarrow \{\partial_{AT}\}$ and $\{b\} \rightarrow \{\partial_{\{b\}}\}$ are functional dependencies. In addition, we observe that $\{b\} \rightarrow \{\partial_{AT}\}$ is a minimal functional dependency in IDT. Nevertheless, there are objects in IDT for which the values of $\partial_{\{b\}}$ and ∂_{AT} differ; for example, $\partial_{AT}(1) \neq \partial_{\{b\}}(1)$. Thus, the minimal functional dependency $\{b\} \rightarrow \{\partial_{AT}\}$ does not imply that $\{b\}$ is a generalized decision reduct.

Theorem 5.2.1. Let $A \to AT$. The existence of a minimal functional dependency $A \to \{\partial_{AT}\}$ in IDT does not imply that A is a generalized decision reduct of IDT.

Corollary 5.2.1. Let $A \subseteq AT$. If A is a generalized decision reduct of IDT, then $A \to \{\partial_{AT}\}$ is a functional dependency, but not necessarily minimal.

Proof. By Proposition 5.2.2 and Theorem 5.2.1. □

5.3 μ-Decision Reducts and Functional Dependencies Under Incompleteness

Lemma 5.3.1. Let $A \subseteq AT$ and $x \in O$. $\forall_{y \in I_A(x)}\ \mu_d^A(y) = \mu_d^A(x)$.

Proof. Analogous to Proof of Lemma 5.2.1; follows from Property 5.1.1b. □

Proposition 5.3.1. Let $A \subseteq AT$. $A \to \{\mu_d^A\}$ is functional in IDT.

Proof. Follows immediately from Lemma 5.3.1. □

Proposition 5.3.2. Let $A \subseteq AT$. If A is a μ-decision reduct of IDT, then $A \to \{\mu_d^{AT}\}$ is a functional dependency in IDT.

Proof. Analogous to the proof of Proposition 5.2.2; follows from the definition of a μ-decision reduct and Proposition 5.3.1. □

Now, we note that $\{b\} \to \{\mu_d^{AT}\}$ is a minimal functional dependency in IDT from Table 3 and $\mu_d^{AT}(1) \neq \mu_d^{\{b\}}(1)$. Thus, the minimal functional dependency $\{b\} \to \{\mu_d^{AT}\}$ does not imply that $\{b\}$ is a μ-decision reduct. Thus, we conclude:

Theorem 5.3.1. Let $A \subseteq AT$. The existence of a minimal functional dependency $A \to \{\mu_d^{AT}\}$ in IDT does not imply that A is a μ-decision reduct of IDT.

Corollary 5.3.1. Let $A \subseteq AT$. If A is a μ-decision reduct of IDT, then $A \to \{\mu_d^{AT}\}$ is a functional dependency, but not necessarily minimal.

Proof. By Proposition 5.3.2 and Theorem 5.3.1. □

5.4 Certain Decision Reducts and Functional Dependencies Under Incompleteness

Lemma 5.4.1. $POS_A = \{x \in O \mid S_A(x) \subseteq I_d(x)\}$.

Proof. $POS_A = \bigcup_{d_i \in V_d} \underline{A}X_{d_i} = \bigcup_{y \in O} \underline{A}I_d(y) = \bigcup_{y \in O} \{x \in O \mid S_A(x) \subseteq I_d(y)\} =$ /* by Proposition 2.2.1 */ $= \bigcup_{y \in O} \{x \in O \mid S_A(x) \subseteq I_d(x)\} = \{x \in O \mid S_A(x) \subseteq I_d(x)\}$. □

Lemma 5.4.2. Let $A \subseteq AT$ and $x \in O$. $\forall_{y \in I_A(x)}\ d_A^{\mathrm{N}}(y) = d_A^{\mathrm{N}}(x)$.

Proof. We shall consider two cases: 1) $x \in POS_A$, and 2) $x \notin POS_A$.

Case 1: By definition, $d_A^{\mathrm{N}}(x) = d(x)$ (*). By Lemma 5.4.1, $S_A(x) \subseteq I_d(x)$. Hence, and by Property 5.1.1a,b, $\forall_{y \in I_A(x)}\ I_A(y) \subseteq S_A(y) \subseteq I_d(x)$, so, $\forall_{y \in I_A(x)}\ I_d(y) =$

$I_d(x)$ (**). Thus, $\forall_{y \in I_A(x)}$ $S_A(y) \subseteq I_d(y)$. Hence, and by Lemma 5.4.1, $\forall_{y \in I_A(x)}$ $y \in POS_A$ (***). By (*), (**) and (**), $\forall_{y \in I_A(x)}$ $d^{\mathrm{N}}_A(y) = d(y) = d(x) = d^{\mathrm{N}}_A(x)$.

Case 2: By definition, $d^{\mathrm{N}}_A(x) = \mathrm{N}$ (*), and by Lemma 5.4.1, $S_A(x) \not\subseteq I_d(x)$. Thus, by Proposition 2.2.1, $\neg(\exists_{z \in O}\ S_A(x) \subseteq I_d(z))$. Hence, and by Property 5.1.1b, $\forall_{y \in I_A(x)} \neg(\exists_{z \in O}\ S_A(y) \subseteq I_d(z))$. So, by Proposition 2.2.1, $\forall_{y \in I_A(x)}$ $S_A(y) \not\subseteq I_d(y)$. Therefore, $\forall_{y \in I_A(x)}$ $y \notin POS_A$. Hence, $\forall_{y \in I_A(x)}$ $d^{\mathrm{N}}_A(y) = \mathrm{N} =$ /* by (*) */ $= d^{\mathrm{N}}_A(x)$. □

Proposition 5.4.1. Let $A \subseteq AT$. $A \rightarrow \{d^{\mathrm{N}}_A\}$ is a functional dependency in IDT.

Proof. Follows immediately from Lemma 5.4.2. □

Proposition 5.4.2. Let $A \subseteq AT$. If A is a certain decision reduct of IDT, then $A \rightarrow \{d^{\mathrm{N}}_{AT}\}$ is a functional dependency in IDT.

Proof. Let A be a certain decision reduct. By the definitions of a certain decision reduct and d^{N}_{AT}, $\forall_{x \in POS}$ $S_A(x) \subseteq I_d(x)$ and $d^{\mathrm{N}}_{AT}(x) = d(x)$. Thus, by Property 5.1.1a, $\forall_{x \in POS}$ $I_A(x) \subseteq I_{d^{\mathrm{N}}_{AT}}(x)$ (*). By Lemma 5.4.1, $\forall_{x \notin POS}$ $S_{AT}(x) \not\subseteq I_d(x)$. Hence, and by Property 5.1.1c, $\forall_{x \notin POS}$ $S_A(x) \not\subseteq I_d(x)$. Thus, by Lemma 5.4.1, $\forall_{x \notin POS}$ $x \notin POS_A$. Therefore and by the definitions of d^{N}_{AT} and d^{N}_A, $\forall_{x \notin POS}$ $d^{\mathrm{N}}_{AT}(x) = \mathrm{N} = d^{\mathrm{N}}_A(x)$. Hence, and by Lemma 5.4.2, $\forall_{x \notin POS}$ $\forall_{y \in I_A(x)}$ $d^{\mathrm{N}}_A(y) = d^{\mathrm{N}}_A(x) = \mathrm{N} = d^{\mathrm{N}}_{AT}(x)$. Thus, $\forall_{x \notin POS}$ $I_A(x) \subseteq I_{d^{\mathrm{N}}_{AT}}(x)$ (**). By (*) and (**), $\forall_{x \in O}$ $I_A(x) \subseteq I_{d^{\mathrm{N}}_{AT}}(x)$. Hence, $A \rightarrow \{d^{\mathrm{N}}_{AT}\}$ is a functional dependency. □

Eventually, we note that $\{b\} \rightarrow \{d^{\mathrm{N}}_{AT}\}$ is a minimal functional dependency and $d^{\mathrm{N}}_{AT}(1) \neq d^{\mathrm{N}}_{\{b\}}(1)$. Hence, the minimal functional dependency $\{b\} \rightarrow \{d^{\mathrm{N}}_{AT}\}$ does not imply that $\{b\}$ is a certain decision reduct. Thus we conclude:

Theorem 5.4.1. Let $A \subseteq AT$. The existence of a minimal functional dependency $A \rightarrow \{d^{\mathrm{N}}_{AT}\}$ in IDT does not imply that A is a certain decision reduct of IDT.

Corollary 5.4.1. Let $A \subseteq AT$. If A is a certain decision reduct of IDT, then $A \rightarrow \{d^{\mathrm{N}}_{AT}\}$ is a functional dependency, but not necessarily minimal.

Proof. By Proposition 5.4.2 and Theorem 5.4.1. □

6 Conclusions

Certain decision reducts, generalized decision reducts, and membership distribution reducts are provable to be sets of conditional attributes that functionally determine respective modifications of a decision attribute both in complete and incomplete information systems. We have also proved, however, that, unlike in the case of complete systems, the reducts in incomplete systems are not guaranteed to be minimal sets of conditional attributes that functionally determine respective modifications of the decision attribute.

References

1. Järvinen, J.: Pawlak's Information Systems in Terms of Galois Connections and Functional Dependencies. Fundamenta Informaticae 75, 315–330 (2007)
2. Kryszkiewicz, M.: Rough Set Approach to Incomplete Information Systems. Journal of Information Sciences 112, 39–49 (1998)
3. Kryszkiewicz, M.: Properties of Incomplete Information Systems in the Framework of Rough Sets. In: Studies in Fuzziness and Soft Computing 18. Rough Sets in Knowledge Discovery 1, pp. 442–450. Physica Verlag, Heidelberg (1998)
4. Kryszkiewicz, M.: Rules in Incomplete Information Systems. Journal of Information Sciences 113, 271–292 (1999)
5. Kryszkiewicz, M.: Rough Set Approach to Rules Generation from Incomplete Information Systems. The Encyclopedia of Computer Science and Technology. Marcel Dekker, Inc. New York, vol. 44, pp. 319–346 (2001)
6. Kryszkiewicz, M.: Comparative Study of Alternative Types of Knowledge Reduction in Inconsistent Systems. Int'l Journal of Int. Systems 16(1), 105–120 (2001)
7. Kryszkiewicz, M.: Comparative Study of Alternative Types of Knowledge Reduction in Inconsistent Systems - Revised. ICS Research Report 13/2004, Warsaw (October 2004)
8. Lin, T.Y.: An Overview of Rough Set Theory from the Point View of Relational Databases. Bulletin of International Rough Set. Society 1(1), 30–34 (1998)
9. Nguyen, H.S.: Approximate Boolean Reasoning: Foundations and Applications in Data Mining. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, pp. 334–506. Springer, Heidelberg (2006)
10. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data, vol. 9. Kluwer Academic Publishers, Boston (1991)
11. Pawlak, Z., Skowron, A.: Rudiments of Rough Sets, Inf. Sci. 177(1) 3–27
12. Pawlak, Z., Skowron, A.: Rough Sets and Boolean Reasoning, Inf. Sci. 177(1) 41–73
13. Skowron, A.: Boolean Reasoning for Decision Rules Generation. ISMIS, 295–305 (1993)
14. Skowron, A., Rauszer, C.: The Discernibility Matrices and Functions in Information Systems. In: Intelligent Decision Support. Handbook of Applications and Advances of Rough Sets Theory, pp. 331–362. Kluwer, Dordrecht (1992)
15. Slezak, D.: Approximate Reducts in Decision Tables. IPMU 3, 1159–1164 (1996)
16. Slezak, D.: Searching for Frequential Reducts in Decision Tables with Uncertain Objects. RSCTC, 52–59 (1998)
17. Slezak, D.: Approximate Entropy Reducts. Fundam. Inform. 53(3-4), 365–390 (2002)
18. Slezak, D.: Association Reducts: Complexity and Heuristics. RSCTC, 157–64 (2006)
19. Slezak, D.: Association Reducts: Boolean Representation. RSKT, 305–312 (2006)

On Covering Attribute Sets by Reducts

Mikhail Moshkov[1], Andrzej Skowron[2], and Zbigniew Suraj[3]

[1] Institute of Computer Science, University of Silesia
Będzińska 39, 41-200 Sosnowiec, Poland
moshkov@us.edu.pl
[2] Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl
[3] Chair of Computer Science, University of Rzeszów
Rejtana 16A, 35-310 Rzeszów, Poland
zsuraj@univ.rzeszow.pl

Abstract. For any fixed natural k, there exists a polynomial in time algorithm which for a given decision table T and given k conditional attributes recognizes if there exist a decision reduct of T containing these k attributes.

Keywords: rough sets, decision tables, decision reducts.

1 Introduction

The set of all decision reducts of a decision table T [4] contains rich information about the table T. Unfortunately, there are no polynomial algorithms for construction of the set of all reducts.

In this paper, we show that there are polynomial (in time) algorithms for obtaining of indirect but useful information about this set.

We show that for any fixed natural k, there exists a polynomial (in time) algorithm $\mathcal{A}_k$ checking, for a given decision table T and given k conditional attributes, if there exist a reduct for T covering these k attributes.

The information obtained on the basis of algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ can be represented in a simple graphical form. One can construct a graph with the set of vertices equal to the set of attributes covered by at least one reduct, and the set of edges coincides with the set of pairs of attributes which do not belong to any reduct. The degree of an attribute in this graph (the number of edges incident to this attribute) characterizes attribute importance. The changes of this graph after adding of a new object to the decision table allow us to evaluate the degree of influence of this new object on a structure of the reduct set of decision table. In the paper, we construct such graphs for two real-life decision tables. Some properties of such graphs are studied in [2].

Note that there exist close analogies between results of this paper and results obtained in [1], where the following problem was considered: for a given positive Boolean function f and given subset of its variables it is required to recognize

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 175–180, 2007.

if there exists a prime implicant of dual Boolean function f^d containing these variables.

Another way for efficient extracting from a given decision table T of indirect information about the set of all reducts and its graphical representation was considered in [7]. It was shown that there exists a polynomial algorithm for constructing the so-called pairwise core graph for a given decision table T. The set of vertices of this graph is equal to the set of conditional attributes of T, and the set of edges coincides with the two element sets of attributes disjoint with the core of T (i.e., the intersection of all reducts of T) and having non-empty intersection with any reduct of T. This example is a step toward a realization of a program suggested in early 90s by Andrzej Skowron in his lectures at Warsaw University to study geometry of reducts for developing tools for investigating geometrical properties of reducts in the space of all reducts of a given information system. For example, the core of a given information system can be empty but in the reduct space can exist only few families of reducts with non-empty intersection.

2 On Covering of k Attribute Sets by Reducts

A *decision table* T is a finite table in which each column is labeled by a *conditional attribute*. Rows of the table T are interpreted as tuples of values of conditional attributes on some objects. Each row is labeled by a *decision* which is interpreted as the value of the *decision attribute*[1].

Let A be the set of conditional attributes (the set of names of conditional attributes) of T. We will say that a conditional attribute $a \in A$ *separates* two rows if these rows have different values at the intersection with the column labeled by a. We will say that two rows are *different* if at least one attribute $a \in A$ separates these rows. Denote by $P(T)$ the set of unordered pairs of different rows from T which are labeled by different decisions.

A subset R of the set A is called a *test* for T if for each pair of rows from $P(T)$ there exists an attribute from R which separates rows in this pair. A test R for T is called a *reduct* for T if each proper subset of R is not a test for T. In the sequel, we deal with decision reducts but we will omit the word "decision".

Let us fix a natural number k. We consider the following *covering problem for k attributes by a reduct*: for a given decision table T with the set of conditional attributes A, a subset B of the set A, and k pairwise different attributes $a_1, \ldots, a_k \in B$ it is required to recognize if there exist a reduct R for T such that $R \subseteq B$ and $a_1, \ldots, a_k \in R$, and if the answer is "yes" it is required to construct such a reduct. We describe a polynomial in time algorithm $\mathcal{A}_k$ for the covering problem.

[1] We consider uniformly both consistent and inconsistent decision tables. However, in the case of inconsistent decision table, one can use also the so called generalized decision instead of the original decision [4,5,6].

For $a \in A$, we denote by $P_T(a)$ the set of pairs of rows from $P(T)$ separated by a. For $a_1, \ldots, a_k \in A$ and $a_j \in \{a_1, \ldots, a_k\}$ let

$$P_T(a_j|a_1, \ldots, a_k) = P_T(a_j) \setminus \bigcup_{i \in \{1,\ldots,k\} \setminus \{j\}} P_T(a_i).$$

For $a_1, \ldots, a_k \in A$, let

$$\mathcal{P}_T(a_1, \ldots, a_k) = P_T(a_1|a_1, \ldots, a_k) \times \ldots \times P_T(a_k|a_1, \ldots, a_k).$$

Assuming that $(\pi_1, \ldots, \pi_k) \in \mathcal{P}_T(a_1, \ldots, a_k)$, we denote by

$$D_T(B, a_1, \ldots, a_k, \pi_1, \ldots, \pi_k)$$

the set of attributes a from $B \setminus \{a_1, \ldots, a_k\}$ such that a separates rows in at least one pair of rows from the set $\{\pi_1, \ldots, \pi_k\}$. Note that

$$D_T(B, a_1, \ldots, a_k, \pi_1, \ldots, \pi_k) = \bigcup_{j=1}^{k} D_T(B, a_j, \pi_j).$$

Using algorithm $\mathcal{A}_k$ first the set $\mathcal{P}_T(a_1, \ldots, a_k)$ is constructed. Next, for each tuple $(\pi_1, \ldots, \pi_k) \in \mathcal{P}_T(a_1, \ldots, a_k)$ the set

$$D_T(B, a_1, \ldots, a_k, \pi_1, \ldots, \pi_k)$$

is constructed and it is verified if the set $B \setminus D_T(B, a_1, \ldots, a_k, \pi_1, \ldots, \pi_k)$ is a test for T. It is clear that $|\mathcal{P}_T(a_1, \ldots, a_k)| \leq n^{2k}$, where n is the number of rows

Algorithm 1. Algorithm $\mathcal{A}_k$ for solving of the covering problem for k attributes by a reduct

```
Input:  Decision table $T$ with the set of conditional attributes $A$, $B \subseteq A$, and
        $a_1, \ldots, a_k \in B$.
Output: If there exists a reduct $R$ for $T$ such that $R \subseteq B$ and $a_1, \ldots, a_k \in R$,
        then the output is one of such reducts; otherwise, the output is "no".
construct the set $\mathcal{P}_T(a_1, \ldots, a_k)$;
for any tuple $(\pi_1, \ldots, \pi_k) \in \mathcal{P}_T(a_1, \ldots, a_k)$ do
    $R \longleftarrow B \setminus D_T(B, a_1, \ldots, a_k, \pi_1, \ldots, \pi_k)$
    if $R$ is a test for $T$ then
        while $R$ is not a reduct for $T$ do
            select $a \in R$ such that $R \setminus \{a\}$ is a test for $T$;
            $R := R \setminus \{a\}$
        end
        return $R$;
        stop
    end
end
return "no" (in particular, if $\mathcal{P}_T(a_1, \ldots, a_k) = \emptyset$, then the output is "no")
```

in T. Using this inequality and the fact that k is fixed natural number, one can prove that the algorithm $\mathcal{A}_k$ has polynomial time complexity. Unfortunately, algorithm $\mathcal{A}_k$ has relatively high time complexity.

The considered algorithm is based on the following proposition:

Proposition 1. *Let T be a decision table with the set of conditional attributes A, $B \subseteq A$, and $a_1, \ldots, a_k \in B$. Then the following statements hold:*

1. *A reduct R for T such that $R \subseteq B$ and $a_1, \ldots, a_k \in R$ exists if and only if there exists a tuple $(\pi_1, \ldots, \pi_k) \in \mathcal{P}_T(a_1, \ldots, a_k)$ such that*
$$B \setminus D_T(B, a_1, \ldots, a_k, \pi_1, \ldots, \pi_k)$$
is a test for T.
2. *If the set $S = B \setminus D_T(B, a_1, \ldots, a_k, \pi_1, \ldots, \pi_k)$ is a test for T then each reduct Q for T, obtained from S by removing from S of some attributes, has the following properties: $a_1, \ldots, a_k \in Q$ and $Q \subseteq B$.*

Proof. Let R be a reduct for T such that $a_1, \ldots, a_k \in R$ and $R \subseteq B$. It is clear that for each $a_j \in \{a_1, \ldots, a_k\}$ there exists a pair of rows π_j from $P(T)$ such that a_j is the only attribute a_j from the set R separating this pair. It is clear that $(\pi_1, \ldots, \pi_k) \in \mathcal{P}_T(a_1, \ldots, a_k)$ and $R \subseteq B \setminus D_T(B, a_1, \ldots, a_k, \pi_1, \ldots, \pi_k)$. Since R is a reduct for T, we conclude that $B \setminus D_T(B, a_1, \ldots, a_k, \pi_1, \ldots, \pi_k)$ is a test for T.

Let us assume that there exists a tuple $(\pi_1, \ldots, \pi_k) \in \mathcal{P}_T(a_1, \ldots, a_k)$ such that the set $S = B \setminus D_T(B, a_1, \ldots, a_k, \pi_1, \ldots, \pi_k)$ is a test for T. Let Q be a reduct for T obtained by removing some attributes from S. It is also clear that $Q \subseteq B$. Let $j \in \{1, \ldots, k\}$. Since a_j is the only attribute from the test S separating rows from π_j, we have $a_j \in Q$. Thus, $a_1, \ldots, a_k \in Q$. □

3 Graphical Representation of Information About the Set of Reducts

Let T be a decision table with the set of conditional attributes A. Let $B \subseteq A$. Using polynomial algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ one can construct a graph $G(T, B)$. The set of vertices of this graph coincides with the set of attributes $a \in B$ for each of which there exists a reduct R for T such that $R \subseteq B$ and $a \in R$. Two different vertices a_1 and a_2 of $G(T, B)$ are linked by an edge if and only if there is no a reduct R for T such that $R \subseteq B$ and $a_1, a_2 \in R$. Let us denote by $G(T)$ the graph $G(T, A)$.

Now, we present the results of two experiments with real-life decision tables from [3].

Example 1. Let us denote by T_L the decision table “Lymphography” [3] with 18 conditional attributes $a_1, \ldots, a_{18}$ and 148 rows. Each of the considered attributes is a vertex of the graph $G(T_L)$. The graph $G(T_L)$ is depicted in Fig. 1. In particular, one can observe from $G(T_L)$ that any reduct of T_L containing a_4 is disjoint with $\{a_2, a_3, a_5, a_7, a_8, a_9, a_{10}, a_{12}\}$.

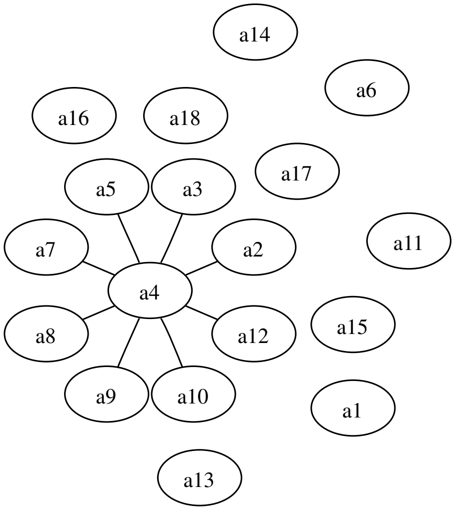


Fig. 1. Graph $G(T_L)$ for the decision table T_L ("Lymphography")

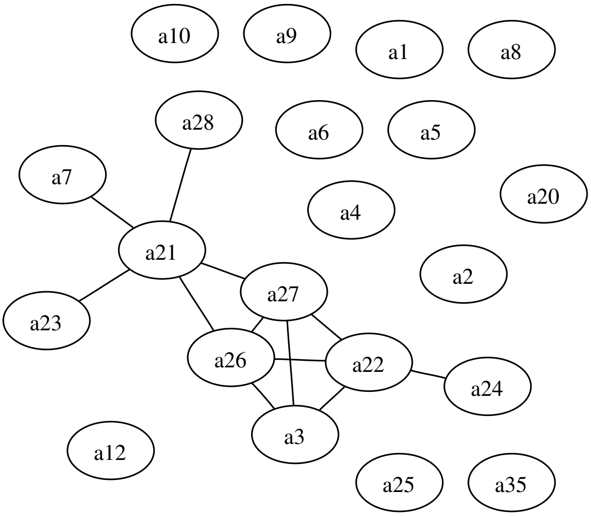


Fig. 2. Graph $G(T_S)$ for the decision table T_S ("Soybean-small")

Example 2. Let us denote by T_S the decision table "Soybean-small" [3] with 35 conditional attributes $a_1, \ldots, a_{35}$ and 47 rows. Only attributes $a_1, \ldots, a_{10}, a_{12}$ and $a_{20}, \ldots, a_{28}, a_{35}$ are vertices of the graph $G(T_S)$. The graph $G(T_S)$ is depicted in Fig. 2.

Some properties of graphs $G(T)$ are studied in [2]. In particular, it is shown that there exists a correlation between the degree of an attribute in $G(T)$ and the number of reducts of T which cover this attribute (last parameter is considered often as attribute importance).

4 Conclusions

In the paper, for each natural k a polynomial algorithm $\mathcal{A}_k$ is studied which for a given decision table and given k conditional attributes recognizes if there exist a decision reduct covering these k attributes. Results of two computer experiments with algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ are reported.

In our further study we would like to check if there exist efficient randomized algorithms for solution of the considered in the paper problem.

Acknowledgments

The authors are grateful to Marcin Piliszczuk for performing of experiments and for his suggestion allowing us to improve algorithms $\mathcal{A}_k$.

References

1. Boros, E., Gurvich, V., Hammer, P.L.: Dual subimplicants of positive Boolean functions. Optimization Methods and Software 10, 147–156 (1998)
2. Moshkov, M., Piliszczuk, M.: Graphical representation of information on the set of reducts. Joint Rough Sets Symposium JRS07, Toronto, Canada, May 14-16, 2007 (to appear)
3. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. University of California, Irvine, Department of Information and Computer Sciences (1998) http://www.ics.uci.edu/~mlearn/MLRepository.html
4. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
5. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007)
6. Pawlak, Z., Skowron, A.: Rough sets and Boolean reasoning. Information Sciences 177(1), 41–73 (2007)
7. Wróblewski, J.: Pairwise cores in information systems. In: Ślęzak, D., Wang, G., Szczuka, M., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 166–175. Springer, Heidelberg (2005)

Applying Rough Sets to Data Tables Containing Missing Values

Michinori Nakata[1] and Hiroshi Sakai[2]

[1] Faculty of Management and Information Science,
Josai International University
1 Gumyo, Togane, Chiba, 283-8555, Japan
nakatam@ieee.org

[2] Department of Mathematics and Computer Aided Sciences,
Faculty of Engineering, Kyushu Institute of Technology,
Tobata, Kitakyushu, 804-8550, Japan
sakai@mns.kyutech.ac.jp

Abstract. Rough sets are applied to data tables containing missing values. Discernibility and indiscernibility between a missing value and another value are considered simultaneously. A family of possible equivalence classes is obtained, in which each equivalence class has the possibility that it is an actual one. By using the family of possible equivalence classes, we can derive lower and upper approximations, even if the approximations are not obtained by previous methods. Furthermore, the lower and upper approximations coincide with those obtained from methods of possible worlds.

Keywords: Rough sets, Missing values, Possible equivalence classes, Lower and upper approximations.

1 Introduction

Rough sets proposed by Pawlak [22] play a significant role in the field of knowledge discovery and data mining. The framework of rough sets has the premise that data tables consisting of precise information are obtained. However, there ubiquitously exists imperfect information containing imprecision and uncertainty in the real world [21]. Under these circumstances, it has been investigated to apply rough sets to data tables containing imprecise values represented by a missing value, an or-set, a possibility distribution, a probability distribution, etc [3,5,6,11,12,14,15,16,17,18,23,24,25,26,28].

The methods are broadly separated into three ways. The first method is one based on possible worlds, which is called a method of possible worlds [20,23,24,25]. In the method, possible tables that consist of precise values are obtained from a data table. Each possible table is dealt with by the conventional method of rough sets and then the results from possible tables are aggregated. There is no doubt for correctness of the treatment, because the conventional method that is already established is applied to each possible table. The second method is to use assumptions on indiscernibility of missing values [3,6,10,11,12,28]. Under the

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 181–191, 2007.

assumptions, we can obtain a binary relation for indiscernibility between objects. To the binary relation the methods of rough sets are applied using a class of objects; for example, an indiscernible class, which is not an equivalence class. The third method directly deals with imprecise values, without any assumption for their indiscernibility with other values. In the method, imprecise values are handled probabilistically or possibilistically and the conventional method is extended probabilistically or possibilistically [14,15,16,17,18,28]. A degree for indiscernibility between any objects is calculated.

A missing value is handled by various ways [7]. When a missing value in an attribute is considered as every value in the domain of the attribute being a possible value, which is called ”do not care” conditions [5], every method mentioned above can deal with the missing value. Thus, these three methods should give the same results under the condition that every missing value means ”do not care.” Recently, Nakata and Sakai have established a third method that gives the same results as the first method, which is called a method of weighted equivalence classes [18,19]. So, we focus on whether or not the second method gives the same result as the first under ”do not care” conditions.

For the second method, the assumption under ”do not care” conditions is that a missing value and every value are indiscernible with each other, which is extensively studied by Kryszkiewicz [10,11]. However, a missing value has two possibilities. One is that it is equal to a value. The other is that it is not equal to the value. In other words, a missing value is regarded as indiscernible and discernible with another value. Our objective is to establish a second method giving the same results as the first method under considering not only indiscernibility but also discernibility of missing values, which is called a method of possible equivalence classes. Our approach corresponds to using the following correctness criterion [14,15,16]:

Correctness criterion

Suppose that operator rep creates set $rep(T)$ of possible tables derived from data table T containing missing values. Let q' be the conventional method applied to $rep(T)$, where q' corresponds to a second method q applied to data table T. The two results is the same; namely, $q(t) = q'(rep(T))$.

This criterion is schematized in Figure 1.

$$\begin{tikzcd}
T \arrow[r, "rep"] \arrow[dr, "q"] & rep(T) \arrow[d, "q'"] \\
 & q(t) = q'(rep(T))
\end{tikzcd}$$

Fig. 1. Correctness criterion of second method q

When this criterion is valid, second method q gives correct results at the level of possible values. This kind of criterion is commonly applied to query expressions in the field of databases handling imprecise information [1,2,9,29].

In Section 2, we briefly address the conventional method of applying rough sets to a data table consisting of precise values. In Section 3, methods of possible worlds are mentioned. This is the preparation for checking whether the method of possible equivalence classes creates the same results as the methods of possible worlds. In Section 4, the method of possible equivalence classes is applied to a data table containing missing values. Section 5 presents conclusions.

2 Rough Sets Under Precise Information

A data set is represented as a table, called an information table, where each row represents an object and each column represents an attribute. The information table is pair $\mathcal{A} = (U, AT)$, where U is a non-empty finite set of objects called the universe and AT is a non-empty finite set of attributes such that $\forall a \in AT\ : U \rightarrow V_a$. Set V_a is called the domain of attribute a. In information table T consisting of set AT of attributes, binary relation $IND(\Psi_A)$ for indiscernibility of objects in subset $\Psi \subseteq U$ on subset $A \subseteq AT$ of attributes is:

$$IND(\Psi_A) = \{(o, o') \in \Psi \times \Psi \mid \forall a \in A\ \ a(o) = a(o')\}. \tag{1}$$

This relation is called an indiscernibility relation. Obviously, $IND(\Psi_A)$ is an equivalence relation. From the indiscernibility relation, equivalence class $E(\Psi_A)_o$ $(= \{o' \mid (o, o') \in IND(\Psi_A)\})$ containing object o is obtained. This is also the set of objects that is indiscernible with object o, called the indiscernible class for object o. Finally, family $\Psi/IND(\Psi_A)$ $(= \{E(\Psi_A)_o \mid o \in \Psi\})$ of equivalence classes is derived from the indiscernibility relation.

Using equivalence classes, lower approximation $\underline{Apr}(\Phi_B, \Psi_A)$ and upper approximation $\overline{Apr}(\Phi_B, \Psi_A)$ of $\Phi/IND(\Phi_B)$ by $\Psi/IND(\Psi_A)$[1] are:

$$\underline{Apr}(\Phi_B, \Psi_A) = \{E(\Psi_A) \mid \exists E(\Phi_B)\ E(\Psi_A) \subseteq E(\Phi_B)\}, \tag{2}$$

$$\overline{Apr}(\Phi_B, \Psi_A) = \{E(\Psi_A) \mid \exists E(\Phi_B)\ E(\Psi_A) \cap E(\Phi_B) \neq \emptyset\}. \tag{3}$$

where $E(\Psi_A) \in \Psi/IND(\Psi_A)$ and $E(\Phi_B) \in \Phi/IND(\Phi_B)$ are equivalence classes for sets Ψ and Φ of objects on sets A and B of attributes, respectively.

3 Methods of Possible Worlds

In methods of possible worlds, the established ways addressed in the previous section are applied to each possible table, and then the results from the possible tables are aggregated. It is a possible table that every missing value is replaced

[1] U_A and U_B are used in place of Ψ_A and Φ_B when Ψ and Φ are equal to U, respectively.

by an element comprising the corresponding domain. When missing values are contained in information table T, set $rep(T)$ of possible tables is:

$$rep(T) = \{pt_1, \ldots, pt_n\}, \tag{4}$$

where each possible table pt_i has an equal possibility that it is the actual one, n is equal to $\Pi_{i=1,m} l_i$, the number of missing values is m, and each of them is a value of an attribute whose domain has $l_i (i = 1, m))$ elements.

All possible tables consist of precise values. Family $U/IND(U_A)_{pt_i}$ of equivalence classes is obtained from each possible table pt_i on set A of attributes. An equivalence class in $U/IND(U_A)_{pt_i}$ is a possible one, because it has the possibility that it is an actual equivalence class. The whole family $U/IND(U_A)$ of equivalence classes is the union of $U/IND(U_A)_{pt_i}$:

$$U/IND(U_A) = \cup_i U/IND(U_A)_{pt_i}. \tag{5}$$

To obtain lower and upper approximations, the conventional methods addressed in the previous section are applied to possible tables. Let $\underline{Apr}(U_B, U_A)_{pt_i}$ and $\overline{Apr}(U_B, U_A)_{pt_i}$ denote lower and upper approximations of $U/IND(U_B)_{pt_i}$ by $U/IND(U_A)_{pt_i}$ in possible table pt_i. Lower and Upper approximations $\underline{Apr}\ (U_B, U_A)$ and $\overline{Apr}(U_B, U_A)$ in information table T are the union of $\underline{Apr}(U_B, U_A)_{pt_i}$ and $\overline{Apr}(U_B, U_A)_{pt_i}$, respectively.

$$\underline{Apr}(U_B, U_A) = \cup_i \underline{Apr}(U_B, U_A)_{pt_i}, \overline{Apr}(U_B, U_A) = \cup_i \overline{Apr}(U_B, U_A)_{pt_i}. \tag{6}$$

When lower and upper approximations are expressed in terms of a set of objects,

$$\underline{apr}(U_B, U_A) = \{o \mid o \in E(A) \wedge E(A) \in \underline{Apr}(U_B, U_A)\}, \tag{7}$$

$$\overline{apr}(U_B, U_A) = \{o \mid o \in E(A) \wedge E(A) \in \overline{Apr}(U_B, U_A)\}. \tag{8}$$

Example 1
We suppose that information table T containing missing values is given as follows:

T

O	a_1	a_2	a_3
1	x	1	a
2	y	2	b
3	$*$	2	b
4	$*$	3	c

pt_1

O	a_1	a_2	a_3
1	x	1	a
2	y	2	b
3	x	2	b
4	x	3	c

pt_2

O	a_1	a_2	a_3
1	x	1	a
2	y	2	b
3	x	2	b
4	y	3	c

pt_3

O	a_1	a_2	a_3
1	x	1	a
2	y	2	b
3	y	2	b
4	x	3	c

pt_4

O	a_1	a_2	a_3
1	x	1	a
2	y	2	b
3	y	2	b
4	y	3	c

The mark O denotes the object identity. Suppose that domains V_{a_1}, V_{a_2}, and V_{a_3} of attributes a_1, a_2, and a_3 are $\{x, y\}$, $\{1, 2, 3\}$ and $\{a, b\}$, respectively. It is a possible table obtained from information table T that every missing value $*$ is replaced by an element comprising the corresponding domain. Four possible tables are derived. For possible table pt_1, families $U/IND(U_{a_1})_{pt_1}$ and $U/IND(U_{a_3})_{pt_1}$ of equivalence classes for attributes a_1 and a_2 are:

$U/IND(U_{a_1})_{pt_1} = \{\{o_2\}, \{o_1, o_3, o_4\}\}$. $U/IND(U_{a_3})_{pt_1} = \{\{o_1\}, \{o_4\}, \{o_2, o_3\}\}$.

Because of $\{o_2\} \subseteq \{o_2, o_3\}$, $\{o_1, o_3, o_4\} \not\subseteq \{o_1\}$, $\{o_1, o_3, o_4\} \not\subseteq \{o_4\}$ and $\{o_1, o_3, o_4\} \not\subseteq \{o_2, o_3\}$,

$$\underline{Aqr}(U_{a_3}, U_{a_1})_{pt_1} = \{o_2\}.$$

Because of $\{o_1, o_3, o_4\} \cap \{o_1\} \neq \emptyset$,

$$\overline{Aqr}(U_{a_3}, U_{a_1})_{pt_1} = \{\{o_2\}, \{o_1, o_3, o_4\}\}.$$

Similarly,

$$\underline{Aqr}(U_{a_3}, U_{a_1})_{pt_2} = \emptyset, \underline{Aqr}(U_{a_3}, U_{a_1})_{pt_3} = \{\{o_2, o_3\}\}, \underline{Aqr}(U_{a_3}, U_{a_1})_{pt_4} = \{\{o_1\}\}.$$

Finally,

$$\underline{Aqr}(U_{a_3}, U_{a_1}) = \cup_i \underline{Aqr}(U_{a_3}, U_{a_1})_{pt_i} = \{\{o_1\}, \{o_2\}, \{o_2, o_3\}\}.$$

For the upper approximation,

$$\begin{aligned} &\overline{Aqr}(U_{a_3}, U_{a_1})_{pt_2} = \{\{o_1, o_3\}, \{o_2, o_4\}\}, \overline{Aqr}(U_{a_3}, U_{a_1})_{pt_3} = \{\{o_1, o_4\}, \{o_2, o_3\}\}, \\ &\overline{Aqr}(U_{a_3}, U_{a_1})_{pt_4} = \{\{o_1\}, \{o_2, o_3, o_4\}\}, \\ \overline{Aqr}&(U_{a_3}, U_{a_1}) = \cup_i \overline{Aqr}(U_{a_3}, U_{a_1})_{pt_i} \\ &= \{\{o_1\}, \{o_2\}, \{o_1, o_3\}, \{o_1, o_4\}, \{o_2, o_3\}, \{o_2, o_4\}, \{o_1, o_3, o_4\}, \{o_2, o_3, o_4\}\}. \end{aligned}$$

For expressions by a set of objects,

$$\underline{aqr}(U_{a_3}, U_{a_1}) = \{o_1, o_2, o_3\}, \overline{aqr}(U_{a_3}, U_{a_1}) = \{o_1, o_2, o_3, o_4\},$$

4 Rough Sets Under Missing Values

When missing values are contained in information table T, Kryszkiewicz defines binary relation $IND(U_A)$ for indiscernibility between objects on set A of attributes as follows [10,11]:

$$IND(U_A) = \{(o, o') \in U \times U \mid \forall a \in A \ a(o) = a(o') \vee a(o) = * \vee a(o') = *\}. \quad (9)$$

This relation for indiscernibility is called a similarity relation. When an object has a missing value as an attribute value, the object may have the same properties as another object on the attribute. Then, the similarity relation treats two objects as similar. This corresponds to "do not care" conditions of missing values addressed by Grzymala-Busse [5,6], where missing values are replaced by all domain elements of the attribute [4]. From the above definition, we obtain that an object where all values on set A of attributes are missing is indiscernible with any object.

From the binary relation, the indiscernible class of object o is derived.

$$S(U_A)_o = \{o' \mid (o, o') \in IND(U_A)\}.$$

The indiscernible class is not an equivalence class. By using indiscernible classes obtained from $IND(U_A)$, Kryszkiewicz expresses lower and upper approximations:

$$\underline{apr}(\Phi, U_A) = \{o \in U \mid S(U_A)_o \subseteq \Phi\}, \tag{10}$$

$$\overline{apr}(\Phi, U_A) = \{o \in U \mid S(U_A)_o \cap \Phi \neq \emptyset\}, \tag{11}$$

where Φ is a set of objects.

The method has crucial drawbacks[2] as is shown in the following example.

Example 2
We suppose that information table T is obtained for the sake of clarifying the drawbacks and the essentials.

T

O	a_1	a_2	a_3
1	x	1	a
2	x	1	a
3	x	1	a
4	x	1	a
5	$*$	2	b

pt_1

O	a_1	a_2	a_3
1	x	1	a
2	x	1	a
3	x	1	a
4	x	1	a
5	x	2	b

pt_2

O	a_1	a_2	a_3
1	x	1	a
2	x	1	a
3	x	1	a
4	x	1	a
5	y	2	b

In information table T, $U = \{o_1, o_2, o_3, o_4, o_5\}$. We suppose that domains V_{a_1}, V_{a_2} and V_{a_3} of attributes a_1, a_2 and a_3 are $\{x, y\}$, $\{1, 2\}$ and $\{a, b\}$, respectively. For indiscernible classes of each objects on attribute a_1,

$$S(U_{a_1})_{o_1} = S(U_{a_1})_{o_2} = S(U_{a_1})_{o_3} = S(U_{a_1})_{o_4} = S(U_{a_1})_{o_5} = \{o_1, o_2, o_3, o_4, o_5\}.$$

We suppose that $\Phi = \{o_1, o_2, o_3, o_4\}$ for simplicity. We focus on lower approximation $\underline{apr}(\Phi, U_{a_1})$, because the upper approximation is trivial in this case. Using formula (10), because of $\{o_1, o_2, o_3, o_4, o_5\} \not\subseteq \{o_1, o_2, o_3, o_4\}$,

$$\underline{apr}(\Phi, U_{a_1}) = \emptyset$$

This shows that we do not obtain any information for the lower approximation. This is true for different expressions [6,8,13] proposed by several authors, where equivalence classes are not used. On the other hand, the method of possible

[2] Stefanowski and Tsoukiàs points out that the method of Kryszkiewicz using ”do not care” conditions creates quite poor results [27]. To handle the problem, other assumptions for indiscernibility of missing values are proposed [3,27].

worlds creates different results. We obtain two possible tables pt_1 and pt_2 from T, because missing value $*$ on attribute a_1 of object o_5 is replaced by x or y, which are a domain element of attribute a_1. Families of equivalence classes on attribute a_1 in possible tables pt_1 and pt_2 are:

$$U/IND(U_{a_1}) = \{o_1, o_2, o_3, o_4, o_5\},\ U/IND(U_{a_1}) = \{\{o_1, o_2, o_3, o_4\}, \{o_5\}\}.$$

From $\{o_1, o_2, o_3, o_4, o_5\} \not\subseteq \Phi$, pt_1 has $\underline{apr}(\Phi, U_{a_1}) = \emptyset$. From $\{o_1, o_2, o_3, o_4\} \subseteq \Phi$, pt_2 has $\underline{apr}(\Phi, U_{a_1}) = \{o_1, o_2, o_3, o_4\}$.

In the above example, possible table pt_1 corresponds to the case where object o_5 is indiscernible with the other objects whereas possible tables pt_2 does to the case where object o_5 is discernible with the other objects. The reason why the method of Kryszkiewicz creates the empty set for the lower approximation is due to that discernibility of a missing value with other values is not considered, although indiscernibility of a missing value with other values is considered. From this consideration, we take into account not only indiscernibility but also discernibility of a missing value with other values.

To handle indiscernibility and discernibility for missing values, we divide universe U into two sets $U_{a=*}$ and $U_{a\neq *}$. $U_{a=*}$ and $U_{a\neq *}$ consists of objects whose value of attribute $a \in A$ is missing value $*$ and is not, respectively. For set $U_{a\neq *}$, we obtain family $U_{a\neq *}/IND(U_{a\neq *})$ of equivalence classes on attribute a by using the conventional method addressed in Section 2. Family $Poss(U/IND(U_a))$ of possible equivalence classes on attribute a is:

$$Poss(U/IND(U_a)) = \{e \cup e' \mid e \in U_{a\neq *}/IND(U_{a\neq *}) \wedge e' \in PU_{a=*}\}, \quad (12)$$

where $PU_{a=*}$ is the power set of $U_{a=*}$. Family $Poss(U/IND(U_A))$ of possible equivalence classes on set A of attributes is:

$$Poss(U/IND(U_A)) = \{\cap_{a\in A} E(U_a) \mid E(U_a) \in Poss(U/IND(U_a))\}\backslash\{\emptyset\}. (13)$$

Element $E(U_A) \in Poss(U/IND(U_A))$ satisfies the following formula:

$$\wedge_{o\in E(U_A)\ and\ o'\in E(U_A)\ and\ a\in A} (a(o) = a(o') \vee a(o) = * \vee a(o') = *)$$
$$\wedge_{o\in E(U_A)\ and\ o'\notin E(U_A) and a\in A}(a(o) \neq a(o') \vee a(o) = * \vee a(o') = *), \quad (14)$$

where $o \neq o'$. The first and second terms deal with indiscernibility and discernibility, respectively.

Proposition 1
When $E(U_A)$ is an element of $Poss(U/IND(U_A))$ in an information table, there exists possible table pt_i where $U/IND(U_A)_{pt_i}$ contains $E(U_A)$.

Proposition 2
$Poss(U/IND(U_A))$ in an information table is equal to the union of the families of equivalence classes, where each family of equivalence classes is obtained from a possible table created from the information table.

Using families of possible equivalence classes, we obtain lower and upper approximations $\underline{Apr}(U_B, U_A)$ and $\overline{Apr}(U_B, U_A)$ of $Poss(U/IND(U_B))$ by $Poss(U/IND(U_A))$. For the lower approximation,

$$\underline{Apr}(U_B, U_A) = \{E(U_A) \mid E(U_A) \subseteq E(U_B) \wedge E(U_A) \in Poss(U/IND(U_A)) \wedge E(U_B) \in Poss(U/IND(U_B))\}. \quad (15)$$

Proposition 3
If $E(U_A)$ in an information table is an element of $\underline{Apr}(U_B, U_A)$, there exists possible table pt_i where $\underline{Apr}(U_B, U_A)_{pt_i}$ contains $E(U_A)$.

For the upper approximation,

$$\overline{Apr}(U_B, U_A) = \{E(U_A) \mid E(U_A) \cap E(U_B) \neq \emptyset \wedge E(U_A) \in Poss(U/IND(U_A)) \wedge E(U_B) \in Poss(U/IND(U_B))\}. \quad (16)$$

Proposition 4
If $E(U_A)$ in an information table is an element of $\overline{Apr}(U_B, U_A)$, there exists possible table pt_i where $\overline{Apr}(U_B, U_A)_{pt_i}$ contains $E(U_A)$.

For expressions in terms of a set of objects, the same expressions as in Section 3 are used.

Proposition 5
The lower and upper approximations that are obtained by the method of possible equivalence classes coincide with ones obtained by the method of possible worlds.

Example 3
For attribute a_1 in the information table of Example 1, $U_{a_1=*}$ and $U_{a_1\neq*}$ that consists of objects whose value of attribute a_1 is missing value $*$ and is not, respectively, are:

$$U_{a_1=*} = \{o_3, o_4\},\ U_{a_1\neq*} = \{o_1, o_2\}.$$

Power set $PU_{a_1=*}$ of $U_{a_1=*}$ is $\{\emptyset, \{o_3\}, \{o_4\}, \{o_3, o_4\}\}$. By using formula (12), the family of possible equivalence classes on attribute a_1 is:

$$Poss(U/IND(U_{a_1})) = \{\{o_1\}, \{o_2\}, \{o_1, o_3\}, \{o_1, o_4\}, \{o_2, o_3\}, \{o_2, o_4\}, \{o_1, o_3, o_4\}, \{o_2, o_3, o_4\}\}.$$

The family of equivalence classes on attribute a_3 is:

$$U/IND(U_{a_3}) = \{\{o_1\}, \{o_4\}, \{o_2, o_3\}\}.$$

Using the families of possible equivalence classes, we derive the lower and upper approximations of $U/IND(U_{a_3})$ by $U/IND(U_{a_1})$. Equivalence classes containing or equal to equivalence classes in $U/IND(U_{a_3})$ are $\{o_1\}$, $\{o_2\}$, and $\{o_2, o_3\}$

in $U/IND(U_{a_1})$. Equivalence classes having non-empty intersection with equivalence classes in $U/IND(U_{a_3})$ are $\{o_1\}$, $\{o_2\}$, $\{o_1, o_3\}$, $\{o_1, o_4\}$, $\{o_2, o_3\}$, $\{o_2, o_4\}$, $\{o_1, o_3, o_4\}$, $\{o_2, o_3, o_4\}$ in $U/IND(U_{a_1})$. Thus, for lower and upper approximations,

$$\begin{aligned}\underline{Apr}(U_{a_3}, U_{a_1}) &= \{\{o_1\}, \{o_2\}, \{o_2, o_3\}\},\\ \overline{Apr}(U_{a_3}, U_{a_1}) &= \{\{o_1\}, \{o_2\}, \{o_1, o_3\}, \{o_1, o_4\}, \{o_2, o_3\}, \{o_2, o_4\}, \{o_1, o_3, o_4\},\\ &\quad\ \{o_2, o_3, o_4\}\}.\end{aligned}$$

For expressions by a set of objects,

$$\underline{apr}(U_{a_3}, U_{a_1}) = \{o_1, o_2, o_3\}. \ \ \overline{apr}(U_{a_3}, U_{a_1}) = \{o_1, o_2, o_3, o_4\}.$$

Indeed, the lower and upper approximations coincide with ones obtained from the method of possible worlds in Example 1.

For computational complexity of lower and upper approximations expressed by formulae (15) and (16), the most crucial factor is the number of possible equivalence classes contained in $Poss(U/IND(U_B))$. This number is $O(m \times 2^n)$ where m is $\max_i m_i$ and n is $\min_i n_i$ when m_i and n_i are the number of equivalence classes in $U_{a_i \neq *}$ and the number of missing values on attribute $a_i \in B$. When any values of attributes contained in set B of decision attributes are not missing, this exponential factor does not appear. In this case, the method of possible equivalence classes would be practically available to large data sets.

5 Conclusions

We have proposed a method, where possible equivalence classes are used, to deal with missing values. The method takes into account not only indiscernibility but also discernibility of a missing value with another value. The lower and upper approximations by the method of possible equivalence classes coincide with ones by the method of possible worlds. In other words, this method satisfies the correctness criterion that is used in the field of incomplete databases. This is justification of the method of possible equivalence classes.

Acknowledgments. This research has been partially supported by the Grant-in-Aid for Scientific Research (C), Japan Society for the Promotion of Science, No. 18500214.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison Wesley, London (1995)
2. Grahne, G.: The Problem of Incomplete Information in Relational Databases. LNCS, vol. 554. Springer, Heidelberg (1991)

3. Greco, S., Matarazzo, B., Slowinski, R.: Handling Missing Values in Rough Set Analysis of Multi-attribute and Multi-criteria Decision Problem. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) New Directions in Rough Sets, Data Mining, and Granular-Soft Computing. LNCS (LNAI), vol. 1711, pp. 146–157. Springer, Heidelberg (1999)
4. Grzymala-Busse, J.W.: MLEM2: A New Algorithm for Rule Induction from Imperfect Data. In: Proceedings of the IPMU'2002, 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Annecy, France, 2002, pp. 243–250 (2002)
5. Grzymala-Busse, J.W.: Data with Missing Attribute Values: Generalization of Indiscernibility Relation and Rule Induction. Transactions on Rough Sets. 1, 78–95 (2004)
6. Grzymala-Busse, J.W.: Characteristic Relations for Incomplete Data: A Generalization of the Indiscernibility Relation. Transactions on Rough Sets. IV, 58–68 (2005)
7. Grzymala-Busse, J.W., Hu, M.: A Comparison of Several Approaches to Missing Attribute Values in Data Mining (2000)
8. Guan, Y.-Y., Wang, H.-K.: Set-valued Information Systems. Information Sciences 176, 2507–2525 (2006)
9. Imielinski, T., Lipski, W.: Incomplete Information in Relational Databases. Journal of the ACM 31(4), 761–791 (1984)
10. Kryszkiewicz, M.: Rough Set Approach to Incomplete Information Systems. Information Sciences 112, 39–49 (1998)
11. Kryszkiewicz, M.: Rules in Incomplete Information Systems. Information Sciences 113, 271–292 (1999)
12. Latkowski, R.: On Decomposition for Incomplete Data. Fundamenta Informaticae 54, 1–16 (2003)
13. Leung, Y., Li, D.: Maximum Consistent Techniques for Rule Acquisition in Incomplete Information Systems. Information Sciences 153, 85–106 (2003)
14. Nakata, N., Sakai, H.: Rough-set-based approaches to data containing incomplete information: possibility-based cases. In: Nakamatsu, K., Abe, J.M. (eds.) Advances in Logic Based Intelligent Systems, Frontiers in Artificial Intelligence and Applications, vol. 132, pp. 234–241. IOS Press, Amsterdam (2005)
15. Nakata, N., Sakai, H.: Checking Whether or Not Rough-Set-Based Methods to Incomplete Data Satisfy a Correctness Criterion. In: Torra, V., Narukawa, Y., Miyamoto, S. (eds.) MDAI 2005. LNCS (LNAI), vol. 3558, pp. 227–239. Springer, Heidelberg (2005)
16. Nakata, N., Sakai, H.: Rough Sets Handling Missing Values Probabilistically Interpreted. In: Ślęzak, D., Wang, G., Szczuka, M., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 325–334. Springer, Heidelberg (2005)
17. Nakata, N., Sakai, H.: Applying Rough Sets to Data Tables Containing Probabilistic Information. In: Proceedings of 4th Workshop on Rough Sets and Kansei Engineering, Tokyo, Japan, pp. 50–53 (2005)
18. Nakata, N., Sakai, H.: Applying Rough Sets to Data Tables Containing Imprecise Information Under Probabilistic Interpretation. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 213–223. Springer, Heidelberg (2006)
19. Nakata, N., Sakai, H.: Lower and Upper Approximations in Data Tables Containing Possibilistic Information. Transactions on Rough Sets. VII, 170–189 (2007)
20. Orłowska, E., Pawlak, Z.: Representation of Nondeterministic Information. Theoretical Computer Science 29, 313–324 (1984)

21. Parsons, S.: Current Approaches to Handling Imperfect Information in Data and Knowledge Bases. IEEE Transactions on Knowledge and Data. Engineering 8(3), 353–372 (1996)
22. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
23. Sakai, H.: Effective Procedures for Handling Possible Equivalence Relation in Non-deterministic Information Systems. Fundamenta Informaticae 48, 343–362 (2001)
24. Sakai, H., Nakata, M.: An Application of Discernibility Functions to Generating Minimal Rules in Non-deterministic Information Systems. Journal of Advanced Computational Intelligence and Intelligent Informatics 10, 695–702 (2006)
25. Sakai, H., Okuma, A.: Basic Algorithms and Tools for Rough Non-deterministic Information Systems. Transactions on Rough Sets. 1, 209–231 (2004)
26. Słowiński, R., Stefanowski, J.: Rough Classification in Incomplete Information Systems. Mathematical and Computer Modelling 12(10/11), 1347–1357 (1989)
27. Stefanowski, J., Tsoukiàs, A.: On the Extension of Rough Sets under Incomplete Information. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) New Directions in Rough Sets, Data Mining, and Granular-Soft Computing. LNCS (LNAI), vol. 1711, pp. 73–81. Springer, Heidelberg (1999)
28. Stefanowski, J., Tsoukiàs, A.: Incomplete Information Tables and Rough Classification. Computational Intelligence 17(3), 545–566 (2001)
29. Zimányi, E., Pirotte, A.: Imperfect Information in Relational Databases. In: Motro, A., Smets, P. (eds.) Uncertainty Management in Information Systems: From Needs to Solutions, pp. 35–87. Kluwer Academic Publishers, Boston (1997)

Category-Based Rough Induction

Marcin Wolski

Department of Logic and Methodology of Science,
Maria Curie-Skłodowska University, Poland
marcin.wolski@umcs.lublin.pl

Abstract. The present paper is concerned with Rough Set Theory (RST) and Similarity Coverage Model (SCM) of category-based induction. It redefines basic concepts of RST in the light of SCM, and explains how RST may be seen as an elegant formal model of inductive reasoning. Furthermore, following SCM, we enrich RST by the concept of an ontology defined as a subset of the family of all definable sets. The paper also presents a model of inductive reasoning which is driven by recent works on RST and nearness-type structures. We show how approximation spaces can be characterised in terms of non-Archimedean nearness spaces.

1 Introduction

Rough Set Theory [7,8] – RST for short – is a mathematical theory of how to deal with uncertainty which stems from the limited amount of available information. Indeed, it provides formal tools to approximate new concepts which are undefinable in terms of gathered knowledge. The present paper aims to show that RST provides also an elegant formal representation of *category-based induction*. This type of inductive reasoning is based on the structure of (folk biological) taxonomies which support predictions and inferences by which people generalise from a particular category (e.g. bobcats) to another one (e.g. all mammals). Actually, there are a number of formal models of category-based reasoning [1,5,6,11]. We consider one such model, namely *Similarity Coverage Model* (SCM) [1,6]. Basically, SCM deals with similar issues as RST: how – despite lack of knowledge – to reason about a given category on the basis of better known categories. Thus, there should be no surprise that, although each theory has a different emphasis and formal methods, both theories may enrich one another. We shall show that RST provides a very well-established formal model for category-based reasoning, while SCM gives some new interpretations of mathematical structures underlying RST.

The basic concept of RST is an *approximations space* defined as a pair (X, E), where X is a set of objects and $E \subseteq X \times X$ is an equivalence relation, called an *indiscernability*, which represents the collected pieces of knowledge about X. This knowledge allows us to specify concepts (i.e. subsets of X), called *definable concepts*; the other concepts may only be approximated. Generally speaking, RST is concerned with such approximations. Hereafter several attempts were made to generalise the concept of an approximation space; in the present paper we shall focus on a *generalised approximation space* introduced by A. Skowron and J. Stepaniuk [10]. Basically, a generalised approximation space is a triple (X, R, v), where R is an uncertainty function $R : X \to 2^X$ (a

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 192–201, 2007.

binary reflexive relation) giving objects y similarly described to x, i.e. $y \in R(x)$, and v is a rough inclusion function telling us to which extent a given set is included in another one. In the light of SCM both types of approximation spaces encode different types of knowledge. An approximation space (X, E), as was stated by Z. Pawlak, represents our knowledge about the universe of objects X. The special role is played by definable concepts of X with respect to E. However, not all definable concepts will be meaningful but only some of them. The set of these distinguished definable concepts will be called an *ontology*. On the other hand, a generalised approximation space (X, R, v) (compatible with the approximation space (X, E)) will support inductive inferences based on the ontology. Thus we do not regard a generalised approximation space, at it has been so far, as a generalisation of an approximation space in the sense of Pawlak, but we view it as an inference engine allowing one to make inductive generalisations from one to another category induced by the approximation space.

The pair of equivalence relations $\{E, R\}$ can be generalised to a non-Archimedean structure which gives rise to a non-Archimedean nearness space (X, ν) – a mathematical model for our investigations. Nearness-type structures and RST have been recently studied in, e.g., [9,12]. Here, starting from considerations about SCM, we add a new characterisation of Pawlak's approximation spaces in terms of non-Archimedean spaces.

2 Similarity Coverage Model

Ethnobiology or folk biology is a branch of cognitive science which studies the ways in which people categorise the local fauna and flora, and project their knowledge about a certain category to another ones [1,5,6,11]. For example, given that *bobcats secrete uric acid crystals* and *cows secrete uric acid crystals*, subjects, on the basis that all mammals may have this property, infer that *foxes secrete uric acid crystals*. There are different accounts of such category-based induction; in the paper we focus on the very influential *Similarity Coverage Model* (SCM) introduced by Osherson et al [6].

According to SCM, the subject calculates the *similarity* of the premise categories (i.e. bobcats, cows) to the conclusion category (i.e. foxes). Then the subject calculates the average similarity (*coverage*) of the premise categories to the superordinate category including both the premise and conclusion categories (i.e. mammals). Let us consider the following example:

$$\begin{array}{c} \textit{Horses have an ileal vein,} \\ \textit{Donkeys have an ileal vein.} \\ \hline \textit{Gophers have an ileal vein.} \end{array}$$

This argument is weaker than:

$$\begin{array}{c} \textit{Horses have an ileal vein,} \\ \textit{Gophers have an ileal vein.} \\ \hline \textit{Cows have an ileal vein.} \end{array}$$

Of course, the similarity of horses to cows is much higher than the similarity of horses or donkeys to gophers. Thus the "strongness" of inductive inferences depends on the maximal similarity of the conclusion category to some of the premise categories. Now let us shed some light on the coverage principle:

$$\begin{array}{c}\textit{Horses have an ileal vein,}\\ \textit{Cows have an ileal vein.}\\ \hline \textit{All mammals have an ileal vein.}\end{array}$$

According to SCM this argument is weaker than the following one:

$$\begin{array}{c}\textit{Horses have an ileal vein,}\\ \textit{Gophers have an ileal vein.}\\ \hline \textit{All mammals have an ileal vein.}\end{array}$$

The reason is that the average similarity of horses to other mammals is almost the same as that of cows. Thus the second premise does not bring us nothing in terms of coverage (both horses and cows share the same set of similar animals). By contrast, gophers are similar to other mammals than horses and thus this premise makes the coverage higher. Observe also that the following inductive inference

$$\begin{array}{c}\textit{Horses have an ileal vein,}\\ \hline \textit{All mammals have an ileal vein.}\end{array}$$

is stronger, than

$$\begin{array}{c}\textit{Bats have an ileal vein,}\\ \hline \textit{All mammals have an ileal vein.}\end{array}$$

The range of mammals similar to cows is much wider than the rage of mammals similar to bats. Thus, cows are more *typical* examples of mammals than bats or gophers.

Now, let us summarise the above examples in a more formal way. Firstly, there is given a set of categories we reason about C. This set is provided with a binary "kind of" relation K, which is acyclic and thus irreflexive and asymmetric.

Definition 1. *An acyclic relation K is called* taxonomic *over C iff K is transitive and for any $a, b, c \in C$ such that aKb and aKc, either $b = c$ or bKc or cKb.*

For example, *collie* is a kind of *dog* and *dog* is a kind of *mammal*. Items $x \in C$, such that there is no t satisfying yKx constitute basic categories. An example of non-taxonomic relation is as follows: *chair* is a kind of *furniture* but not a kind of *vehicle*; furthermore, *wheelchair* is a kind of *chair* and a kind of *vehicle*. Now, *wheelchairKchair* and *wheelchairKvehicle*, but neither *chair* = *vehicle* nor *chairKvehicle* nor *vehicleKchair*.

Subjects reasoning about (C, K) are additionally provided with similarity relation R defined on basic categories. Actually, this is R what allows one to reason about categories under incomplete knowledge: given that elements of A have a property p, a subject may infer that elements of B also satisfy p when ARB and A is a "substantial" subset of B. Thus, (C, K) represents gathered information, while R is an inductive "engine" making inferences about unknown features of objects belonging to C. In the next section we shall represent SCM in terms of structures taken from RST.

3 Rough Set Theory

In this section we recall notions of an approximation space and a generalised approximation space. They are fundamental concepts of RST [10,7,8] which, traditionally viewed, represent given knowledge about some universe of objects X. Here we show how the basic notions related to rough sets may be applied to inductive reasoning based on a categorisation of objects.

Definition 2. *A pair (X, E), where X is a nonempty set (of objects) and $E \subseteq X \times X$ is an equivalence relation, is called an* approximation space.

A subset $A \subseteq X$ is called *definable* if $A = \bigcup \mathcal{B}$ for some $\mathcal{B} \subseteq X/E$, where X/E is the family of equivalence classes of E. The chief idea underlying RST is to approximate an undefinable set A by means of two definable sets:

Definition 3. *Let (X, E) be an approximation space and $[x]_E$ the equivalence class containing $x \in X$. With each $A \subseteq X$, we can associate its E-lower and E-upper* approximations, *$\underline{A}$ and $\overline{A}$, respectively, defined as follows:*

$$\underline{A} = \{x \in X : [x]_E \subseteq A\},$$

$$\overline{A} = \{a \in X : [x]_E \cap A \neq \emptyset\}.$$

The lower approximation $\underline{A}$ consists of points which necessarily belong to A while the upper approximation $\overline{A}$ consists of points which possibly belong to A. Actually, $\underline{A}$ is the greatest definable set included in A, while $\overline{A}$ is the smallest definable superset of A. Thus RST employs all definable subsets of X.

Definition 4. *A* generalised approximation space *is a triple(X, R, v), where*

- *X is a set of objects,*
- *$R : X \to \mathcal{P}(X)$ is an* uncertainty function*, where $\mathcal{P}(X)$ is the powerset of X,*
- *$v : \mathcal{P}(X) \times \mathcal{P}(X) \to [0, 1]$ is a* rough inclusion function.

The uncertainty function R defines for every object $x \in X$ a set of objects similar to x. The standard inclusion function v_s defined on a finite universe X is given by:

$$v_s(A, B) = \begin{cases} \frac{card(A \cap B)}{card(A)} & \text{if } A \neq \emptyset \\ 1 & \text{if } A = \emptyset \end{cases}$$

Typically, generalised approximation spaces are viewed as a generalisation of approximation spaces in the sens of Pawlak's definition. Here we would like to change this perspective and interpret these structures from the standpoint of SCM. All categories are viewed extensionally, i.e. as subsets of some universe X. Thus, each category is a sum of some basic categories. For a given approximation space (X, E) basic categories are given by the equivalence classes $[x]_E$ of E. Furthermore, we can define different *ontologies* on (X, E). The richest ontology is given by the set of all definable subsets of (X, E) denoted by $\mathcal{D}$. It is worth emphasising that actually $\mathcal{D}$ is the ontology implicitly assumed in RST. Thus, an approximation space may be represented by $(X, E, \mathcal{D})$.

From the perspective of SCM this ontology is generally too rich; therefore we shall call any $\mathcal{O} \subseteq \mathcal{D}$ an *ontology* over $(X, E, \mathcal{D})$. The triple $(X, E, \mathcal{O})$ will be called *$\mathcal{O}$-approximation space*. For every ontology $\mathcal{O}$ we define its "kind of" relation K by: AKB iff $A \neq B$ and $A \subseteq B$. The ontology is called taxonomic iff K is a taxonomic relation over $\mathcal{O}$. Obviously, $\mathcal{D}$ is an example of non-taxonomic ontology.

The main difference between RST and SCM is that is RST we often reason about unknown concepts, i.e. some undefinable subsets of X, while in SCM we are always concerned with known concepts forming our ontology; yet this knowledge is often incomplete. That is why in RST the "kind of relation" K comes second after indiscernability relation E. Although E is often regarded as similarity, in terms of induction over categories (i.e. definable sets) it brings us nothing: given that x has a property p, we can reason that y also satisfies p when xEy, but then they necessary share the same category $[x]_E = [y]_E$. To reason we need another structure, which allows one to reason about categories.

Let us consider a generalised approximation space (X, R, v). Now, we show how it models an inductive reasoning in the sense of SCM. Firstly, the space shares the same set of objects X as its corresponding $\mathcal{O}$-approximation space $(X, E, \mathcal{O})$. Additionally, we assume that R, representing similarity among objects, is an extension of E in the sense that R is an equivalence relation and $R(x) \in \mathcal{D}$, for all $x \in X$. The last element of generalised approximation space, namely v, allows us to model the coverage rule of SCM. The average similarity (coverage) of A to B, where $A, B \in \mathcal{O}$, is defined by $asim(A, B) = v(B, A)$. Now we can model a process of induction over an $\mathcal{O}$-approximation space $(X, E, \mathcal{O})$. Let us consider the following scenario: a child has knowledge about animals represented by E. This relation allows she to distinguish the following basic categories (i.e. equivalence classes of E): *lions*, *tigers*, *elephants*, *zebras*, *skunks*, *jaguars*, *foxes*, *giraffes*, and *weasels*. The child has only two "supreme" categories: *cats* consisting of *lions*, *tigers*, and *jaguars*; and *non-cats*. Thus,

$$\mathcal{O} = \{lions, tigers, elephants, zebras, skunks, jaguars, foxes, giraffes,$$
$$weasels, cats, non - cats\}$$

Additionally, she perceives that – according to R – *lions* are similar to *tigers*, *jaguars* and *foxes*; and *skunks* goes with *weasels*. As uncertainty function we take v_s. Now, given that *tigers* are dangerous and *skunks* stink, she may infer that:

$$\frac{\textit{Tigers are dangerous.}}{\textit{Cats are dangerous.}}$$

or

$$\frac{\textit{Skunks stink.}}{\textit{Weasels stink.}}$$

Moreover, the following argument

$$\frac{\textit{Tigers are dangerous.}}{\textit{All animals in Zoo are dangerous.}}$$

is stronger than

$$\frac{\textit{Skunks stink.}}{\textit{Non-cats stink.}}$$

since $asim(Tigers, Zoo) = 4/9$ is greater than $asim(Skunks, Non - cats) = 2/6$. Of course, the first argument above is much stronger than the last two; it is easy to calculate that $asim(Tigers, Cats) = 1$. By the same reason, arguments based on *elephants* or *zebras* are very weak.

As far, we have described a very simple device for inductive reasoning based on RST. An approximation space in the sense of Pawlak's definition provides gathered knowledge about some universe X. It also induces basic categories and an ontology $\mathcal{O}$. A parameterised approximation space provides means to reason about these categories. In the next section we shall try to find a better representation for such type of induction.

4 Rough Induction

Recently a number of attempts have been made to connect RST with nearness type structures, e.g. [9,12]. Firstly we start with a few simple observations concerning approximation spaces and category-based induction. Then we shall build a mathematical structure which generalises these observations and prove how it is related to nearness spaces.

The knowledge about a given universe is encoded by an equivalence relation E. Furthermore, to reason about concepts we need a similarity relation R which is compatible with E. Since both E and R are equivalence relations we may regard them as partitions of X: $\mathcal{P}_E$ and $\mathcal{P}_R$, respectively.

Definition 5. *Let $\mathcal{A}, \mathcal{B} \subseteq 2^X$; then a* refinement relation $\preceq$ *is defined by:*

$$\mathcal{A} \preceq \mathcal{B} \overset{def}{\Leftrightarrow} \forall A \in \mathcal{A}\, \exists B \in \mathcal{B} (A \subseteq B).$$

Let us observe that the pair $\beta = \{\mathcal{P}_E, \mathcal{P}_R\}$ fullfills the following condition:

$$\mathcal{P}_E \preceq \mathcal{P}_R$$

The simple mathematical structure which generalise this observation is called a *non-Archimedean structure* [2].

Definition 6. *A non-Archimedean structure μ on a set X is a set of partitions of X satisfying:*
(i) if $\mathcal{A} \preceq \mathcal{B}$ and $\mathcal{A} \in \mu$, then $\mathcal{B} \in \mu$.
The couple (X, μ) is called a non-Archimedean space.

Please observe that we use here a more general concept of non-Archimedean structure [2] instead of a more popular notion of non-Archimedean uniformity.

Definition 7. $stack_{\preceq}\mu = \{\mathcal{B} \subseteq 2^X : \exists \mathcal{A} \in \mu (\mathcal{A} \preceq \mathcal{B})\}$.

It is easy to observe that:

$$stack_{\preceq}\beta = stack_{\preceq}\{\mathcal{P}_E\}.$$

The stack operation allows us to connect E with nearness type structures.

Definition 8. *Let X be a set and ν be a non-empty set of coverings of X such that:*
(i) if $\mathcal{A}$ refines $\mathcal{B}$ and $\mathcal{A} \in \nu$, then $\mathcal{B} \in \nu$.
Then (X, ν) is called a pre-nearness space

Thus $stack_{\preceq}\{\mathcal{P}_E\}$ is a pre-nearness space. Let (X, ν) be a pre-nearness space and let

$$\mathcal{E}_\nu = \{\mathcal{P} \in \nu : \mathcal{P} \text{ is a partition of } X\}$$

When $stack_{\preceq}\mathcal{E}_\nu = \nu$, then (X, ν) is called a *non-Archimedean pre-nearness space* and $\mathcal{E}_\nu$ is its *base*. Thus a non-Archimedean structure μ on X is a base of the non-Archimedean pre-nearness space $(X, stack_{\preceq}\mu)$. Now we answer the following question: given a non-Archimedean structure μ induced by an equivalence relation E, what is $(X, stack_{\preceq}\mu)$?

Definition 9. *Let X, ν be a pre-nearness space such that:*
(i) if $\mathcal{A} \in \nu$ and $\mathcal{B} \in \nu$, then $\{A \cap B : A \in \mathcal{A}$ and $B \in \mathcal{B}\} \in \nu$.
Then (X, ν) is called a merotopic space.

Definition 10. *A merotopic space (X, ν) which satisfies*
(i) if $\mathcal{A} \in \nu$, then $\{Int_\nu(A) : A \in \mathcal{A}\} \in \nu$, where $Int_\nu(A) = \{x \in X : \{A, X \setminus \{x\}\} \in \nu\}$, is called a nearness space.

Proposition 1. *Let μ a non-Archimedean structure on X induced by an equivalence relation E. If μ satisfies the condition (i) of Definition 9 then $(X, stack_{\preceq}\mu)$ is a non-Archimedean nearness space.*

Proof. In order not to overload the paper with definitions, we shall give just a hint how to prove this theorem. Every partition star-refines itself and therefore $(X, stack_{\preceq}\mu)$ is a uniform pre-nearness space. Furthermore, any uniform pre-nearness space satisfies (i) of Definition 10. See also [3].

Proposition 2. *There is one-to-one correspondence between finite approximation spaces (X, E) and finite non-Archimedean nearness spaces.*

Proof. For the same reason as above we also give a sketch of proof. Every finite non-Archimedean nearness space (X, ν) is induced by a partition $\mathcal{P}$ and, as a result, by some equivalence relation E. Since $\mathcal{P}$ is a minimal open basis for the topology induced by Int_ν it follows that (X, ν) is a topological nearness space. But there is one-to-one correspondence between equivalence relations and topological nearness spaces. See also [12].

Proposition 3. *Let (X, E) be an approximation space, $\mathcal{P}_E$ the partition corresponding to E, and (X, μ) the non-Archimedean space induced by $\mathcal{P}_E$, i.e. μ is a set of partitions refined by $\mathcal{P}_E$; then $\bigcup \mu = \mathcal{D}$.*

Proof. Firstly, by the refinement condition every $B \in \mathcal{B} \in \mu$ is a sum of basic categories, i.e. equivalence classes of E. Thus, $B \in \mathcal{D}$. Secondly, each definable set $A \in \mathcal{D}$ induces a partition of X, namely $\{A\} \cup \mathcal{D} \setminus \{[x]_E : x \in A\}$ which is refined by $\mathcal{P}_E$. Thus, this partition belongs to μ.

Thus, a non-Archimedean structure μ gives us a representation for $\mathcal{D}$ – the richest ontology of (X, E). Therefore, one can represent an $\mathcal{O}$-approximation space $(X, E, \mathcal{D})$ as (X, E, μ). However, we would like to restrict the ontology $\mathcal{D}$ to a smaller structure. Moreover, a non-Archimedean nearness space $(X, stack_{\preceq}\mu)$ seems to be more interesting than (X, μ).

Definition 11. *Let (X, E) be an approximation space and (X, ν) its non-Archimedean nearness space. By an* ontology $\mathcal{O}_\nu$ *over (X, ν) we mean a finite chain of partitions $\mathcal{A}_1 \preceq \mathcal{A}_2 \preceq \mathcal{A}_k$, where $\mathcal{A}_i \neq \mathcal{A}_j$ for $i \neq j$, and $\mathcal{A}_i \in \nu$ for every i.*

It is easy to observe that:

Proposition 4. *An ontology $\mathcal{O}_\nu$ over a non-Archimedean nearness space (X, ν) is taxonomic iff every $\mathcal{A}_i \in \mathcal{E}_\nu$.*

Proof. The obvious "kind of" relation K is as follows: AKB iff $A \subseteq B$ and there exist $\mathcal{B}_i, \mathcal{B}_j \in \mathcal{O}$ such that $A \in \mathcal{B}_i$, $B \in \mathcal{B}_j$ and $i < j$. Now, K is transitive and no A can stand in relation K to two distinct elements B, C of $\mathcal{B}_j$, since every $\mathcal{B}_j$ is a partition of X. Thus, K is taxonomic. On the other hand, when $\mathcal{O}_\nu$ is taxonomic then for each cover $\mathcal{A}_i \in \mathcal{O}_\nu$ we have that for every $A, B \in \mathcal{A}_i$, $A \cap B = \emptyset$. Thus, every $\mathcal{A}_i$ is a partition and belongs to $\mathcal{E}_\nu$.

Thus, for finite spaces X we can regard (X, E) as a non-Archimedean nearness space $(X, stack_{\preceq}\{\mathcal{P}_E\})$. Now we would like to enrich this space by an ontology $\mathcal{O}$ to obtain a nearness-type $\mathcal{O}$-approximation space. A structure $(X, \nu, \mathcal{O}_\nu)$, where (X, ν) is a non-Archimedean nearness space and $\mathcal{O}_\nu$ an ontology over ν, will be called an $\mathcal{O}_\nu$-approximation space. Please observe that the non-Archimedean structure induced by $\mathcal{P}_E$, which represents the family $\mathcal{D}$ of all definable sets, is not an ontology. Generally we are interestend in taxonomic ontologies.

Now, let us consider how an $\mathcal{O}_\nu$-approximation space $(X, \nu, \mathcal{O}_\nu)$, where $\mathcal{O}_\nu$ is taxonomic, may be applied to SCM. It is worth emphasising that here we measure similarity among categories by means of the ontology $\mathcal{O}_\nu$ alone. Suppose we are given a standard taxonomy in terms of genus, family, suborder, subclass and class. Suppose also that a skunk and a bear share a biological property. Michigan students are reported to conclude that it is more likely that all mammals have this property than if it were shared by a skunk and an opossum [1]. Students actually made a false assumption that skunks are taxonomically further from bears than from opossum. To model their reasoning, let $\mathcal{O}_\mu$ denote an ontology representing this taxonomy. By $\mathcal{B}_i$ we shall denote the first i such that $skunks, bears \in B$ for some $B \in \mathcal{B}_i \in \mathcal{O}_\mu$. Similarly, $\mathcal{B}_j$ we shall denote the first j such that $skunks, opossums \in B$ for some $B \in \mathcal{B}_j \in \mathcal{O}_\mu$. The students actually assumed that $i \leq j$. It shows that the higher diversity of premises strengthens the argument. Let us recall that recently A. Gomolińska has studied extensions of approximation spaces by an additional relation understood as a relation of dissimilarity of objects [4]. Here, dissimilarity is just the inverse of similarity.

Now, let us consider a similarity relation R which is assumed to be an equivalence relation compatible with E; thus $\mathcal{P}_R$ belongs to the non-Archimedean space induced by E. One might also wish the similarily relation to encode to what an extent two objects are similar. It all can be done by means of another taxonomic ontology. To be more precise, let (X, E) be an approximation space, (X, ν) its non-Archimedean nearness space, $\mathcal{O}_\nu$ and $\mathcal{O}'_\nu$ taxonomic ontologies over (X, ν). The former ontology encodes our knowledge and thus $\mathcal{P}_E = \mathcal{A}_1$. The latter ontology represents similarity among objects: two objects x, y are similar in a degree i, when there is $A \in \mathcal{A}_i \in \mathcal{O}'_\nu$ such that $x, y \in A$. It is worth emphasising that the smaller i the more similar objects. Thus, similarity encoded in $\mathcal{O}'_\nu$ shows how *near* the objects are. Summing up, we need a non-Archimedean nearness space (X, ν), and two taxonomic ontologies $\mathcal{O}_\nu, \mathcal{O}'_\nu$, where $\mathcal{O}_\nu$ is generated by $\mathcal{P}_E$ such that $stack_{\preceq}\{\mathcal{P}_E\} = \nu$, and a rough inclusion function v. Of course, these two ontologies may by easily generalised to a set of ontologies $\mathcal{O}$.

Consider the following scenario. Children often believe that whales and bats are mammals, despite the lack of knowledge about anatomical facts that make these identifications possible. Thus, ontologies often represent given apriori scientific knowledge. On the other hand children also have aposteriori knowledge about the surrounding world. Therefore, let $\mathcal{O}'_\mu$ represent their real-life experience. For example, one could observe that bats are very similar to mouses (e.g. $\{bats, mouses, rats\} \in B \in \mathcal{B}_1$), while *wales* goes alone. On this basis the following argument

$$\frac{\textit{Bats suffer from A.}}{\textit{Mammals suffer from A.}}$$

is stronger than

$$\frac{\textit{Whales suffer from A.}}{\textit{Mammals suffer from A.}}$$

One could also observe that, e.g., $\{bats, mouses, rats, cats, dogs\} \in B \in \mathcal{B}_2$. Then

$$\frac{\textit{Bats suffer from A.}}{\textit{Dogs suffer from A.}}$$

is weaker than

$$\frac{\textit{Bats suffer from A.}}{\textit{Rats suffer from A.}}$$

The above examples show that inductive reasoning may be seen as a kind of interplay between apriori and aposteriori knowledge. Suppose, that students try to reason over their false taxonomy. Since $\mathcal{O}'_\nu$ must be in accordance with a false taxonomy $\mathcal{O}_\nu$, students eventually fail to find $\mathcal{O}'_\nu$ giving sufficiently good predictions. Then they will be forced to abandon the false ontology $\mathcal{O}_\nu$. Thus, to some extent we can model also the process of learning.

To sum up, the most general structure coming from the above examples is a non-Archimedean nearness space (X, ν) equipped with a set $\mathcal{O}$ of ontologies over ν and a rough inclusion function v, that is $(X, \nu, \mathcal{O}, v)$.

5 Final Remarks

The article presents account of preliminary results concerning Rough Set Theory (RST) and Similarity Coverage Model of category-based induction (SCM). We have showed how RST can model this type of inductive reasoning employing both approximation spaces: approximation spaces in the sense of Pawlak's definition and generalised approximation spaces. The former spaces provide knowledge about a given universe of objects; especially it allows to define the set of basic categories and an ontology. The former space provides an "inductive engine" allowing us to make inferences – despite of incomplete knowledge – about some categories on the basis of better known ones. Furthermore, following recent papers about nearness type structures and RST, we have shown how non-Archimedean structures and non-Archimedean nearness spaces allow us to represent approximation spaces and to model the SCM-based inductive reasoning.

References

1. Atran, S.: Classifying nature across cultures. In: Osherson, D., Smith, E. (eds.) An Invitation to Cognitive Science. Thinking, pp. 131–174. MIT Press, Cambridge, Massachusetts (1995)
2. Deses, D., Lowen-Colebunders, E.: On completeness in a non-Archimedean setting, via firm reflections. Bulletin of the Belgian Mathematical Society, Special volume: p-adic numbers in number theory, analytic geometry and functional analysis, pp. 49–61(2002)
3. Deses, D.: Completeness and zero-dimensionality arising from the duality between closures and lattices. PhD. Thesis, Free University of Brussels (2003) http://homepages.vub.ac.be/~diddesen/phdthesis.pdf
4. Gomolińska, A.: Approximation spaces based on similarity and dissimilarity. In: Lindemann, G., Schlingloff, H., Burkhard, H., Czaja, L., Penczek, W., Salwicki, A., Skowron, A., Suraj, Z. (eds.) Concurrency, Specification and Programming CS&P', Informatik-Berichte 206, Berlin 2006, pp. 446–457 (2006)
5. Heit, E.: Properties of inductive reasoning. Psychonomic Bulletin & Review 7, 569–592 (2000)
6. Osherson, D.N., Smith, E.E., Wilkie, O., Lopez, A., Shafir, E.: Category-based induction. Psychological Review 97(2), 185–200 (1990)
7. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
8. Pawlak, Z. (ed.): Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Boston (1991)
9. Peters, J., Skowron, A., Stepaniuk, J.: Nearness in approximation spaces. In: Lindemann, G., Schlingloff, H., Burkhard, H., Czaja, L., Penczek, W., Salwicki, A., Skowron, A., Suraj, Z. (eds.): Concurrency, Specification and Programming CS&P', Informatik-Berichte 206, Berlin 2006, pp. 434–445 (2006)
10. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
11. Sloman, S.A.: Feature-based induction. Cognitive Psychology 25, 231–280 (1993)
12. Wolski, M.: Similarity as Nearness: Information Quanta, Approximation Spaces and Nearness Structures. In: Lindemann, G., Schlingloff, H., Burkhard, H., Czaja, L., Penczek, W., Salwicki, A., Skowron, A., Suraj, Z. (eds.): Concurrency, Specification and Programming CS&P'2006, Informatik-Berichte 206, Berlin 2006, pp. 423–433 (2006)

Finding the Reduct Subject to Preference Order of Attributes

Xiaofeng Zhang, Yongsheng Zhao, and Hailin Zou

School of Computer Science and Technology
Ludong University, Yantai 264025, P.R. China
iamzxf@126.com, jsjzhao@sohu.com, zhl_8655@sina.com

Abstract. In machine learning and knowledge discovery, rough set theory is a useful tool to be employed as a preprocessing step for dimension reduction. However, for a given system, there may be more than one reduct to be selected. Different reducts will lead to discovered knowledge, which may be concise, precise, general, understandable and practically useful in different levels. It is a crucial issue to select the most suitable features or properties of the objects in a dataset in the machine learning process. In this paper, some external information is added to information system and may be simply regarded as user preference on attributes. Consequently, it will guide the procedure of retrieving reducts, which will give birth to the reduct subject to preference order of attributes.

Keywords: Reduct, rough set theory, preference order of attributes.

1 Description of Problem

Information system is the main object in data mining and knowledge discovery. It consists of two major parameters of complexity leading to intractable behavior: the number of attributes in an application domain, namely dimensionality, and the number of examples in a dataset. The latter is typically applied only to the training stage of the system and, depending on intended use, may be acceptable. However, data dimensionality is an obstacle for both the training and runtime phases of a learning system. Many systems exhibit non-polynomial complexity with respect to dimensionality, which imposes a ceiling on the applicability. The curse of dimensionality limits the applicability of learning systems to a great degree.

Rough set theory, proposed by Pawlak Z. [3], is a formal methodology that can be employed in data reduction as a preprocessing step. A fundamental notion supporting this is the concept of reduct, which has been studied extensively by many researchers. A reduct is a subset of attributes which are jointly sufficient and individually necessary for preserving the same information under consideration as provided by the entire set of attributes. However, for a given information system, there may be more than one reduct. The use of different reducts will lead to different discovered knowledge. Typically, discovered knowledge should be concise, precise, general, easy to understand and practically useful, which can

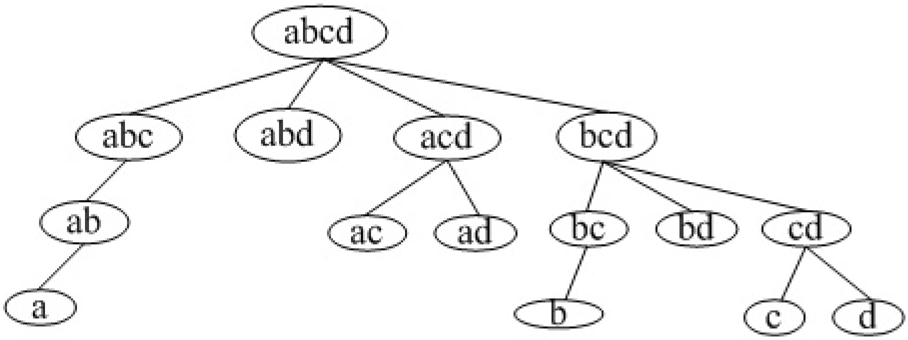
M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 202–210, 2007.

be measured according to external information. In this paper, we will consider such external information simply as user preference, which may be weights of attributes, ranking of attributes, and etc. Especially, if the preference is formally a chain, then the reduct subject to preference order of attributes will be unique.

2 Related Research

This section will introduce several proposals associated with "optimal reduct". One is "optimal reduct" in [2], which is in fact the reduct containing the least number of attributes, also is the shortest one. This algorithm makes use of heuristic information in discernibility matrix–attribute frequency to retrieve the shortest reduct, yet they cannot make sure that the reduct they get is affirmatively the shortest one in any case, but in most cases.

In [6], we propose the concept "optimal reduct under dictionary order". We assume that all attributes are ordered lexically, and therefore, all reducts are ordered lexically accordingly. Since dictionary order is formally a chain, the optimal reduct is unique. In this paper, by constructing a special data structure, called dictionary tree, we design a new algorithm to retrieve this reduct under dictionary order. As an example, a typical dictionary tree containing 4 attributes is shown in Figure 1.



Notes: The attribute set serial when accessing the dictionary tree by mid-root mode is $a, ab, abc, abcd, abd, \ldots, d$, which is ordered under dictionary order.

Fig. 1. A Dictionary Tree Containing 4 attributes

When the tree is accessed in mid-root traversing, the attribute set serial is $a, ab, abc, abcd, abd, \ldots, d$, which is in the lexical order. Moreover, we prove the correctness of the algorithm to construct the dictionary tree and algorithm to retrieve optimal reduct. However, when the algorithm is applied in real environment, it is hard to be carried out.

Yao presented a formal model of machine learning by considering user preference in [5]. This model combined internal information and external information seamlessly and could be extended to user preference of attribute sets. In that paper, Yao discussed many useful properties of user preference and presented two linear preference order, called *left-to-right* lexical order and *right-to left* one. In addition, two general algorithms are designed to retrieve corresponding optimal reduct under the two linear order from the deletion and addition strategy. However, the two algorithms were implemented based on two concepts–super reduct and partial reduct. How to judge whether a feature set is a super reduct or partial one is still a crucial issue while no reduct is available.

3 Basic Concepts and Theories

3.1 Information System

Definition 1. *An information table is a quadruple:*

$$\mathrm{IT} = (U, A, \{V_a|a \in A\}, \{I_a|a \in A\}) \tag{1}$$

Where

U is a finite nonempty set of objects,
A is a finite nonempty set of attributes,
V_a is a nonempty set of values for $a \in A$,
$I_a : A \rightarrow V_a$ is an information function.

For simplicity, we only consider information tables characterized by two finite sets: U and A, of which are objects and attributes, formally as (U, A). In general, an information table contains all available information and knowledge about objects under consideration, which are only perceived or measured by using attributes in A.

Given an information system (U, A), for two objects $x, y \in U$ and one attribute $a \in A$, if the value of x on a is equal to that of y on the same attribute, denoted as $x =_a y$, then we say that the two objects are indiscernible on a. For $B \subseteq A$, if for all attributes $b \in B$, $x =_b y$ holds, denoted as $x =_B y$, we call that the two objects are indiscernible on B.

In (U, A), the family of equivalence class with respect to A, denoted as $\mathrm{IND}(A)$, is defined as following.

$$\mathrm{IND}(A) = \{[x]_A|x \in U\} \tag{2}$$

where $[x]_A = \{y|y \in U, \text{ and } , x =_A y\}$ is the equivalence class containing x constructed by A.

$\mathrm{IND}(A)$ is the set of equivalence class, and can be seen as the classification of the given universe. For example, given $U = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\}$, if $\mathrm{IND}(A) = \{\{u_1, u_2, u_6, u_8\}, \{u_3, u_4, u_5, u_7\}\}$, then we can say that u_1, u_2, u_6 and

u_8 can be seen as the same class by available knowledge in (U, A) and u_3, u_4, u_5 and u_7 can be seen as another different class.

Provided an information system (U, A), $P, Q \subseteq A$, an functional dependency $P \rightarrow Q$ holds if equation (3) holds.

$$\forall u, v \in U, u =_P v \Rightarrow u =_Q v \tag{3}$$

Generally, an functional dependency $P \rightarrow Q$ has the following properties 1-3 [1]:

Property 1. $P \rightarrow Q \Rightarrow P \rightarrow q$, for all $q \in Q$; $(R_{inclusion})$

Property 2. $P \rightarrow Q \Rightarrow (P \cup V) \rightarrow Q$; $(R_{augment})$

Property 3. $P \rightarrow V \wedge V \rightarrow Q \Rightarrow P \rightarrow Q$; (R_{trans})

Definition 2. *In the given information system (U, A), an attribute $a \in A$ is* dispensable*, if the following equation holds.*

$$\mathrm{IND}(A) = \mathrm{IND}(A - \{a\}) \tag{4}$$

Lemma 1. *$a \in A$ is dispensable in (U, A) if and only if $A - \{a\} \rightarrow a$ holds.*

Definition 3. *Given an information system (U, A), $P \subseteq A$ is* dependent*, if any attribute $p \in P$ is not dispensable. Formally, P is dependent if and only if*

$$\forall p \in P, \mathrm{IND}(P) \neq \mathrm{IND}(P - \{p\}) \tag{5}$$

3.2 User Preference

In machine learning algorithms, it is implicitly assumed that all attributes are of the same importance from a user's point of view. Consequently, attributes are based solely on their characteristics revealed in an information system. This results in a simple model, which is easy to analyze. At the same time, without considering the semantic information of attributes, the model is perhaps unrealistic. A more applicable model can be built by considering attributes as non-equally important. This type of external information is normally provided by users in addition to the information system, and is referred to as user judgement or user preference [5].

Given an information system (U, A), for any $p, q \in A$, if p is preferred to q by user, we will simply denote it as $p \succ q$.

Also, how to acquire user preference is a crucial issue. In this paper, for clarity, we simply assume that a user can express preference on the entire attribute set precisely and completely, and this enable us to investigate the real issues without the interference of unnecessary constraints. For simplicity, we assume that any two attributes are preferred in user preference, that is to say, all attributes are assumed to be ordered in a linear order. Formally, $\forall p, q \in C$, either $p \succ q$ holds, or $q \succ p$ holds. Based on user preference on attribute, we can define user preference on the set of attributes as follows.

Definition 4. *Given two feature set* $P = \{p_1, p_2, \ldots, p_m\}$ *and* $Q = \{q_1, q_2, \ldots, q_n\}$ *such that* $p_1 \succ p_2 \succ \ldots \succ p_m$ *and* $q_1 \succ q_2 \succ \ldots \succ q_n$*, where* $\succ$ *is user preference on attributes. Let* $t = min\{m, n\}$*. We say that* P *precedes* Q*, written* $P \succ Q$ *if and only if either of the following two conditions holds:*

(1) *there exist a* $1 \leq i \leq t$ *such that* $p_j = q_j$ *for* $1 \leq j \leq i$ *and* $p_i \succ q_i$

(2) $a_i = b_i$ *for* $0 \leq i \leq t$ *and* $m < n$*.*

User preference defined above is in fact *left-to-right* lexical order in [5] and there are many applications in practice, such as dictionary order, and etc. If all attributes in (U, A) are linearly ordered in user preference, all reducts of (U, A) must be ordered in user preference and the reduct which is preferred to any other reduct is called *optimal reduct under user preference.* Particularly, if user preference is a linear order, the optimal reduct under assumed preference must be unique.

4 Optimal Reduct Under Preference

In this section, first we present two algorithms associated with optimal reduct, yet both of them have disadvantages. The feature set retrieved by the first algorithm is surely to be one reduct, but not the optimal one. The second algorithm will retain the better attributes, but cannot give birth to one reduct. After the two algorithms, we present the algorithm to retrieve optimal reduct in information system (U, A) and prove its correctness.

4.1 Algorithm to Retrieve Comparatively Optimal Reduct

First we will give an algorithm to retrieve comparatively optimal reduct. The reason why we call a comparatively optimal reduct is that the feature set we retrieve is a reduct but we cannot ensure it is the optimal one under preference. The algorithm is illustrated in *Algorithm 1*.

```
Data: an information system $(U, A)$, while $A = \{a_1, a_2, \ldots, a_m\}$ satisfying the
      predefined preference such that $a_1 \succ a_2 \succ \ldots \succ a_m$.
Result: an reduct $red$ of $(U, A)$.
$red = A$;
$i$=m;
while $i \leq 1$ do
    if $red - \{a_i\} \rightarrow a_i$ then
        $red = red - \{a_i\}$;
    end
    else
        $i = i - 1$;
    end
end
```

Algorithm 1. Algorithm $COReductRetrIS(U, A)$

The strategy that Algorithm 1 adopts is deletion strategy, that is to say, the procedure of implementing the algorithm is to delete dispensable attributes one by one. However, the reduct retrieved is not sure to be the optimal one under user preference defined in this paper; this will be illustrated in the following example.

Example 1. Given an information table (U, A) in Table 1 as follows.

Table 1. An Information Table

U	a	b	c
1	1	2	0
2	1	2	0
3	1	1	1
4	0	0	0
5	0	0	0
6	0	3	1

In the given information system, there are two reducts: $\{a, c\}$ and $\{b\}$. According to Algorithm 1, we will get $\{b\}$ as the final output, but it is not the optimal reduct under user preference defined in this paper. However, it is optimal reduct under *right-to-left* lexical order defined in [5], which will be discussed in the expanded version of this paper.

4.2 Algorithm to Retrieve Optimal Feature

Since the reduct retrieved by the algorithm in the former section is not the optimal one, we will extend the algorithm so as to retrieve the optimal feature set, which may not certainly be the reduct. Based on the measure function $\alpha(P)$ on P which is increasing monotonously, the algorithm is illustrated in Algorithm 2 as follows, which can be seen as the revised version of algorithm in [7].

Algorithm 2 attempts to retain the important attributes, however, the final feature set is not sure to be one reduct.

4.3 Algorithm to Retrieve Optimal Reduct

After two attempted algorithms, we will design the algorithm to retrieve the optimal reduct for the given information system (U, A) as follows.

Now let us prove the correctness of the algorithm.

Proof. In order to prove the correctness of the designed algorithm, there are three problems to be explained:

(1) $\text{IND}(redr) = \text{IND}(A)$;

(2) $redr$ is dependent;

(3) There is no other reduct $redr' \subseteq A$ such that $redr' \succ redr$.

Now we will prove the above three sub-problems one by one.

```
Data: An decision information table $(U, A)$,while $A = a_0, a_1, \ldots, a_n$.
Result: A feature set $GloBalFea$.
$GlobalFea = \Phi$;
L1:
for $(i = 1\ to\ n\ do\ )$ do
    $TempGlobalFea = \{S_q | \alpha(S_q) = max\{\alpha(S_p), p = 1..n\}\}$
    if $(TempGlobalFea = GlobalFea)$ then
        Return $GlobalFea$;
        End Algorithm;
        else
            $GlobalFea = TempGlobalFea$
            for $(j = 1..n)$ do
                $S_j = Sj \bigcup GlobalFea$
            end
            Goto L1;
        end
    end
end
return $GlobalFea$;
```

Algorithm 2. Algorithm for optimal features

```
Data: an information system $(U, A)$, while $A = \{a_1, a_2, \ldots, a_m\}$ satisfying the
      predefined preference such that $a_1 \succ a_2 \succ \ldots \succ a_m$.
Result: an reduct $red \subseteq A$.
$i$=1;
while $i \leq m$ do
    $redr = \{a_i\}$;
    $j = i + 1$;
    while $j \leq m$ do
        if $redr \nrightarrow a_j$ then
            if $\forall a_p \in redr,\ such\ that\ (red - \{a_p\}) \cup \{a_j\} \nrightarrow a_p$ then
                $redr = redr \cup \{a_j\}$;
                if IND($redr$) = IND($A$) then
                    return $redr$, algorithm end.
                end
            end
        end
        $j = j + 1$;
    end
    $i = i + 1$;
end
```

Algorithm 3. Algorithm for optimal reduct

(1) According to the step which can end the algorithm in the algorithm, the first formula is apparent;

(2) In the algorithm, the final reduct *redr* is produced by adding attributes one by one. Therefore, we can prove that it is dependent in the following two steps.

a.) In the initial step, since $redr = \{a_i\}$, and there is only one attribute, obviously it is dependent;

b.) Suppose that in the q^{th} step, $redr = \{a_{k_1}, a_{k_2}, \ldots, a_{k_q}\}$ satisfies the dependent property, that is to say, any attribute a_p in $redr$ satisfies $\mathrm{IND}(redr) \neq \mathrm{IND}(redr - \{a_p\})$. According to the algorithm, if one attribute a_j is added to $redr$, it must satisfy the following two properties:

• $redr \not\rightarrow a_j$, which ensures that a_j is not dispensable at the $(q+1)^{th}$ step based on the result of ***Lemma 1***;

• $\forall a_u \in redr, (redr - \{a_u\}) \cup a_j \not\rightarrow a_u$, which ensures that all attributes in current attribute set $redr$ are not dispensable in the new attribute set $redr \cup \{a_j\}$ retrieved in $(q+1)^{th}$ step based on the result of ***Lemma 1***.

Therefore, the attribute set $redr \cup a_j$ retrieved in $(q+1)^{th}$ step is dependent.

(3) Suppose the reduct retrieved by applying the designed algorithm is $redr = \{a_{k_1}, a_{k_2}, \ldots, a_{k_u}\}$. If it is not the optimal one, there must be another reduct $redr' = \{a_{l_1}, a_{l_2}, \ldots, a_{l_v}\}$ such that $redr \succ redr'$.

According to Definition 4, $redr \succ redr'$ holds must satisfy either of the following two conditions, from which we will prove the sub-theorem.

a.) $l_v \leq k_u$ and $\forall q \leq l_v, a_{k_q} = a_{l_q}$.

According to the designed algorithm, this case will not occur in the designed algorithm. For if $redr' = \{a_{l_1}, a_{l_2}, \ldots, a_{l_v}\}$ and $redr = \{a_{k_1}, a_{k_2}, \ldots, a_{k_u}\}$ are two reducts, according to description in this case, $\forall q \leq l_v, a_{k_q} = a_{l_q}$ and $l_v < k_u$, which is to say, $redr' \subset redr$, which is contradict to the definition of reduct.

b.) $\exists w,$ such that $\forall q \leq w, a_{k_q} = a_{l_q},$ and $a_{l_{q+1}} \succ a_{k_{q+1}}$.

Supposing that in some step of the implement of the algorithm, we retrieve the attribute set is $redr = \{a_{k_1}, a_{k_2}, \ldots, a_{k_q}\}$, if we can explain that $redr'$ will be the reduct we retrieved in the algorithm, then we can prove the sub-question.

First we can get the conclusion that current attribute set $redr$ is not a reduct, otherwise the algorithm will stop. Then according to the hypothesis $a_{l_{q+1}} \succ a_{k_{q+1}}$, the two attributes must in the same order in given preference. In the procedure of the algorithm, any attribute a_j between a_{l_q} and $a_{l_{q+1}}$ will not be added to $redr$, either $redr \rightarrow a_j$ holds, or there exists better attribute $a_q \in redr$, such that $(redr - \{a_q\}) \cup \{a_j\} \rightarrow a_q$ holds.

Since attributes are added one by one in the order of given preference, $a_{l_{q+1}}$ will be faced ahead of $a_{k_{q+1}}$. Furthermore, for current attribute set $redr$ is not the reduct and there exists a reduct $\{a_{l_1}, a_{l_2}, \ldots, a_{l_v}\}$ containing $a_{l_{q+1}}$ according to the hypothesis, therefore, the following equation $\mathrm{IND}(redr) \neq \mathrm{IND}(redr \cup \{a_{l_{q+1}}\})$ holds. From this *Step* , $a_{l_{q+1}}$ will not be added in $redr$ if and only if $\nexists P \subseteq \{a_{l_{q+2}}, \ldots, a_m\}$, such that $\mathrm{IND}(redr \cup P \cup \{a_{l_{q+1}}\}) \neq \mathrm{IND}(A)$, of course this is impossible, for there exists a reduct $redr'$.

From the above three aspects, we can illustrate that the reduct retrieved in the algorithm is the one subject to preference order of attributes.

4.4 Complexity Analysis

Suppose $|A| = m$ and $|U| = n$. The worst case is that the reduct is the last attribute. In such case, when $i = 1$, we will add all other attributes one by one. We will judge all functional dependencies in 2-attribute set, 3-attributes, ..., m-attribute set. When judging in 2-attribute set, we will test whether two functional dependencies only containing one attribute in the right hold, the complexity of each is $2 * n^2$, so finding all functional dependencies in 2-attribute set is $(2n)^2$. Also, finding all functional dependencies in 3-attribute set, 4-attribute set, ..., m-attribute set are of $(3n)^2, (4n)^2, \ldots, (mn)^2$ complexity. Therefore, the complexity to judge whether the first attribute is included in the reduct is $(2n)^2 + (3n)^2 + \ldots + (mn)^2$.

In the same mode, judging whether the second attribute is included in the reduct is of the following complexity: $(2n)^2+(3n)^2+\ldots+((m-1)n)^2,\ldots$, judging the $(m-1)^{\text{th}}$ attribute is of $(2n)^2$ complexity.

Therefore, the total complexity of this algorithm is

$$(2n)^2 + \ldots + (mn)^2 + (2n)^2 + \ldots + ((m-1)n)^2 + \ldots + (2n)^2 = O(n^2m^4)$$

5 Conclusions

This paper investigates how to retrieve the reduct subject to preference order of attributes in the information system. However, how good is the reduct subject to preference order of attributes for both knowledge representation and prediction? Algorithms should take into account not only the length and the preference order of attributes, but also the intended application of the obtained reduct.

References

1. Dan, Simovici, A.: Relational Database System. Academic Press, San Diego (2002)
2. Hu,K., Diao,L., Lu,Y., Shi,C.: Sampling for approximate reduct in very large databases. URL: //citeseer.ist.psu.edu/587308.html
3. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
4. Yao, Y., Zhao, Y., Wang, J.: On reduct construction algorithms. In: Wang, G.-Y., Peters, J.F., Skowron, A., Yao, Y. (eds.) RSKT 2006. LNCS (LNAI), vol. 4062, pp. 297–304. Springer, Heidelberg (2006)
5. Yao,Y., Zhao,Y., Wang,J., Han,S.: A model of Machine Learning Baed on User Preference of Attributes (to be published)
6. Zhang, X., Zhang, F., Li, M., Wang, N.: Research of Optimal reduct under preference. Computer Engineering and Design 26, 2103–2106 (2005)
7. Zhao, Y., Zhang, X., Jia, S., Zhang, F.: Applying PSO in finding useful features. In: Wang, G.-Y., Peters, J.F., Skowron, A., Yao, Y. (eds.) RSKT 2006. LNCS (LNAI), vol. 4062, pp. 580–585. Springer, Heidelberg (2006)

Transformation of Fuzzy Takagi-Sugeno Models into Piecewise Affine Models

Martin Herceg*, Michal Kvasnica, and Miroslav Fikar

Institute of Information Engineering, Automation and Mathematics,
Slovak University of Technology,
81237 Bratislava, Slovakia
{martin.herceg,michal.kvasnica,miroslav.fikar}@stuba.sk

Abstract. Fuzzy modeling of dynamical systems can be viewed as an interpolation of a collection of linear models where the interpolation coefficients depend on set membership functions. The fuzzy interference applies only when the membership functions intersect otherwise only one model is valid. The approach presented in this paper models the intersections with an uncertainty measure reducing the overall fuzzy model to Piecewise Affine (PWA) description, over-approximating the original fuzzy model. Once such an approximation is calculated, existing algorithms can be applied which yield controllers guaranteeing closed-loop stability. Since the PWA model over-approximates a given fuzzy model, if such a controller is calculated, it guarantees stability of the original fuzzy model as well.

Keywords: Takagi-Sugeno models, Piecewise Affine models, Model Predictive Control.

1 Introduction

It is well known that fuzzy modelling can approximate any process with prescribed accuracy and therefore it can be classified as an universal approximation [10]. Although the particular issues of fuzzy modeling are not addressed in this paper some valuable references can be found for example in [1]. Instead, we focus on application of fuzzy models as prediction patterns in Model Predictive Control (MPC). MPC is an optimization-based control policy widely adopted by the industry due to its ability to provide optimal performance together with constraint satisfaction [14]. In the MPC framework, the prior knowledge of the process behavior, represented by the prediction model, is used to design a sequence of control inputs such that certain performance criterion is optimized. Contrary to classical proportional-integral-derivative (PID) controllers, the decisions are done with respect to process properties and constraints.

Depending on the model used, slightly different approaches were developed. Supposedly the first approach was made by impulse response models, pioneered

[*] Corresponding author.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 211–220, 2007.

by [5,15], denoted as Dynamic Matrix Control (DMC). Revolutionary contribution was brought by [4] where the step responses serve for predictions, often abbreviated as Generalized Predictive Control (GPC). The use of state-space models is associated to the original term MPC as well as the use of Piecewise Affine (PWA) models. The growing need for tighter controlling demands motivated use of nonlinear models. This novel approach is nowadays referred as Nonlinear Model Predictive Control (NMPC) [2,11]. Recently, many predictive strategies which employ fuzzy models emerged and this class of control problems is referred to as Fuzzy Model Predictive Control (FMPC).

An excellent comparative study [6] provides a deeper view into four recently developed predictive strategies. Each strategy uses the GPC approach but the control action is calculated differently. Either a linear combination of all locally designed controllers is considered or a global controller based on linear time-varying models (LTV) is used. A hierarchical structure of multiple Takagi-Sugeno models, proposed by [8,13], deploys the GPC approach where the controller is obtained by weighted aggregation over governing local rules. All of the aforementioned approaches, however, do not address the issues of closed-loop stability. Stability concerns have been partially resolved in [12,19,9] using Linear Matrix Inequalities (LMI). These techniques are, however, overly conservative, since they assume that all possible local dynamical models are all active at the same time.

Motivated by the lack of rigorous results in the field of synthesis of stabilizing MPC controllers based on fuzzy Takagi-Sugeno (TS) models [18] we propose a different way of assuring closed-loop stability and feasibility. Considering the recent advances in the field of hybrid systems [3], we propose to convert a given Takagi-Sugeno fuzzy model into a PWA model, for which efficient control strategies ensuring closed-loop stability and infinite-time feasibility exist [16,7]. Unlike TS fuzzy models, the PWA description requires that the regions, over which individual dynamical modes are defined, to be non-overlapping. Therefore we propose to over-approximate the overlaps naturally present in TS models by means of an unknown, but bounded additive uncertainty. The main contribution of the paper, described in Section 3, is represented by a constructive procedure to derive a PWA model from a Takagi-Sugeno fuzzy model. Beside controller synthesis, the resulting PWA model can be used, for instance, to perform reachability analysis tasks and hence to verify safety and liveness properties of the underlying systems. Since the PWA model over-approximates the original Takagi-Sugeno model, all results hold for this original formulation as well.

2 Takagi-Sugeno Fuzzy Model Representation

The class of Takagi-Sugeno (TS) models can be generally described by fuzzy "IF ... THEN" rules where the fuzzy sets stay on the antendecent side while the consequence is given by a linear dynamics. Generally, the ith TS rule can be expressed as

$$\text{IF } x_{1,k} \text{ is } \mu_{i1} \text{ and } \ldots\ x_{n,k} \text{ is } \mu_{in} \quad \text{THEN } x_{k+1} = A_i x_k + B_i u_k \tag{1}$$

where $x_k \in \mathbb{R}^{n\times 1}$ is the state vector, $u_k \in \mathbb{R}^{m\times 1}$ denotes the vector of manipulated variables, μ_{ij} are input fuzzy sets for $i = 1, \ldots, r$ rules. $A_i \in \mathbb{R}^{n\times n}$, $B_i \in \mathbb{R}^{n\times m}$ are matrices representing the system dynamics. The process dynamics is assumed to be discretized with k denoting one sampling instant. The aggregated system output is modelled using the max-product inference, i.e.

$$x_{k+1} = \frac{\sum_{i=1}^{r} w_i(x_k)(A_i x_k + B_i u_k)}{\sum_{i=1}^{r} w_i(x_k)} \tag{2}$$

with

$$w_i(x_k) = \prod_{j=1}^{n} \mu_{ij}(x_{j,k}) \tag{3}$$

where the membership function μ_{ij} measures the activation of the fuzzy set j in the rule i. Using the notation

$$\alpha_i(x_k) = \frac{w_i(x_k)}{\sum_{i=1}^{r} w_i(x_k)},\ \alpha_i(x_k) > 0,\ \sum_{i=1}^{r} \alpha_i(x_k) = 1 \tag{4}$$

the overall system model can be described as

$$x_{k+1} = \sum_{i=1}^{r} \alpha_i(x_k)(A_i x_k + B_i u_k). \tag{5}$$

A simple TS model is illustrated in Fig. 1 using three rules with linear fuzzy membership functions. It can be seen that each dynamics contributes to the overall model with its corresponding membership function and moreover, if the state belongs to a region where more than one dynamics become active, then the weighted contribution of overlapping nodes is considered.

3 The Transformation Procedure

In this section the main result of the paper will be presented. Consider the TS model (1) with linear fuzzy membership functions. The fuzzy input sets μ_{ij} can be decomposed in the following manner:

$$\text{IF } x_k \in \mathcal{P}_i \quad \text{THEN } x_{k+1} = A_i x_k + B_i u_k \tag{6}$$

where the region $\mathcal{P}_i$ in which the corresponding rule is active can be described by a polyhedral set

$$\mathcal{P}_i := \{H_i x_k \leq K_i\}. \tag{7}$$

The aim is to transform the TS model into a Piecewise Affine model of the following form:

$$\begin{aligned} x_{k+1} &= f_{PWA}(x_k, u_k, w_k) \\ &= A_i x_k + B_i u_k + f_i + w_k \qquad \text{if} \quad x_k \in \mathcal{D}_i \end{aligned} \tag{8}$$

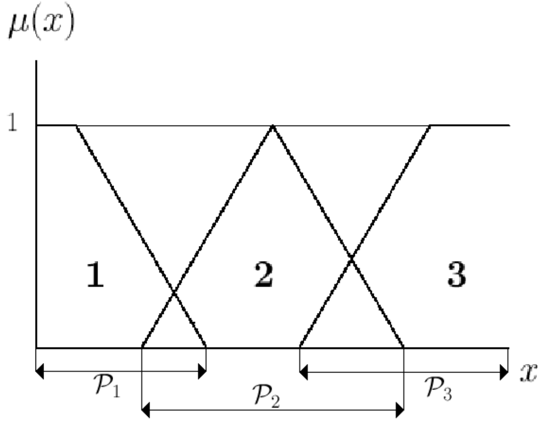


Fig. 1. Illustration of linear membership functions for three rules in the Takagi-Sugeno modelling approach

with $A_i \in \mathbb{R}^{n\times n}$, $B_i \in \mathbb{R}^{m\times n}$, and $f_i \in \mathbb{R}^{n\times 1}$. Here, $\{\mathcal{D}_i\}_{i=1}^{n_d} \in \mathbb{R}^n$ denotes a polyhedral partition satisfying $\mathcal{D} = \bigcup_{i=1}^{n_d} \mathcal{D}_i$. The measured state is denoted by x_k, manipulated inputs correspond to u_k, and w_k denotes an unknown additive disturbance. The system states x, control inputs u as well as the disturbance w of the system (8) are subject to the constraints

$$x_k \in \mathcal{X} \subseteq \mathbb{R}^n, u_k \in \mathcal{U} \subseteq \mathbb{R}^m, w_k \in \mathcal{W} \subseteq \mathbb{R}^n, \quad \forall k \in \{0, \ldots, N\} \tag{9}$$

where $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{W}$ are polyhedral sets containing the origin in their respective interiors.

To obtain the strictly separated regions $\mathcal{D}_i$, the overlaps in the membership functions of the TS model have to be removed first. This can be done in a straightforward manner by defining new regions for each intersection of the neighboring fuzzy sets, i.e.

$$\mathcal{D}_j = \mathcal{P}_i \cap \mathcal{P}_{i+1} \quad j = 1, \ldots, n_i \tag{10}$$

which is also a polyhedral set. If the set $\mathcal{D}_j$ is a subset of the next set (e.g. when more than 2 fuzzy sets intersect) then the statement

$$\mathcal{D}_j \subset \mathcal{D}_{j+1} \Rightarrow \mathcal{D}_j = \emptyset \tag{11}$$

implies that the redundant sets will be removed. Fig. 2 depicts the decomposition of the fuzzy sets to a crisp sets by introducing additional regions $\mathcal{D}_2$ and $\mathcal{D}_4$, respectively. Because the regions $\mathcal{P}_i$ are represented by convex polytopes as in (7), the overall calculation of intersections can be performed using standard

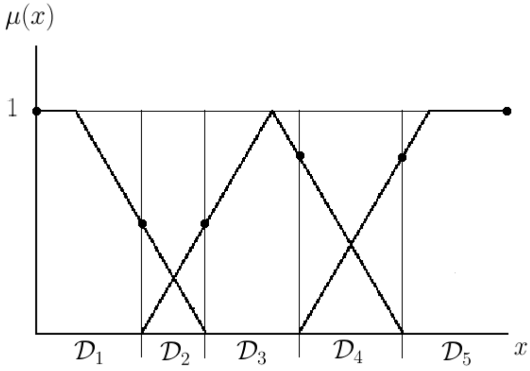


Fig. 2. Intersections of fuzzy sets are replaced by new regions with crisp boundaries

algebraic manipulation techniques. The remaining regions can be obtained by a set-difference operation

$$\mathcal{D}_j = \bigcup_i \mathcal{P}_i \setminus \bigcup_{i=1}^{n_i} \mathcal{D}_i := \left\{ x_k \in \mathbb{R}^n \mid x_k \in \bigcup_i \mathcal{P}_i, x_k \notin \bigcup_{i=1}^{n_i} \mathcal{D}_i \right\} \quad j = n_i, \ldots, n_d \tag{12}$$

Secondly, it is important to determine the mean PWA description for each region with bounded additive uncertainty, i.e. to express the transition from (2) to (8). To do so, the worst case perturbations of the mean model have to be considered. Obviously, these values will be located at the boundary of each region, as indicated by black dots in Fig. 2. Thus, the mean model for region $\mathcal{D}_j$ is given by arithmetic mean of neighboring models corresponding to the boundary of a given region, i.e.

$$\hat{A}_j = \frac{1}{n_n} \sum_i A_i, \quad \hat{B}_j = \frac{1}{n_n} \sum_i B_i \tag{13}$$

with $i \in \mathcal{I}_j$ where $\mathcal{I}_j$ stands for the index set of dynamics active in the region $\mathcal{D}_j$ and n_n denotes the number of overlapping models.

The next step is to determine the affine term f_j and the maximal allowed uncertainty w_j in each region $\mathcal{D}_j$. For this purpose the maximum allowed reachable set of the uncertain system is explored. Let $\mathcal{A}_j, \mathcal{B}_j$ denote the families of possible realizations of matrices $\hat{A}_j, \hat{B}_j$. An over-approximation of the maximum reachable set for the region $\mathcal{D}_j$ is given by

$$\mathcal{T}_j := \{x_{k+1} \mid \underline{x}_{k+1} \leq x_{k+1} \leq \overline{x}_{k+1}\} \tag{14}$$

where the update x_{k+1} of the state is driven by the TS model (5) and $\underline{x}_{k+1}$ and $\overline{x}_{k+1}$ denote, respectively, the lower and upper limits of all possible realizations of x_{k+1}. The key idea is to use an approximation of the form

$$
\begin{aligned}
x_{k+1} &\cong \hat{x}_{k+1} \\
\sum_{i=1}^{r} \alpha_i(x)(A_i x_k + B_i u_k) &\cong \hat{A}_j x_k + \hat{B}_j u_k \quad \text{if} \quad x_k \in \mathcal{D}_j \subset \mathcal{X} \\
&\qquad \text{s.t.}\ u_k \in \mathcal{U}, \quad \hat{A}_j \in \mathcal{A}_j, \quad \hat{B}_j \in \mathcal{B}_j \\
&\qquad\qquad j = 1, \ldots, n_d
\end{aligned}
\tag{15}
$$

and to transform the model (15) into a PWA system with bounded additive disturbances (8). Note that the PWA model (8) actually over-approximates the behavior of the original problem (5) because even if the linearization for the particular regions $\mathcal{D}_j$ is determined, the conservatism appears in the unknown signal w where the maximum allowed disturbance is considered. Obviously, the transformation will be applied to regions where multiple membership functions overlap. In the remaining regions only a single dynamical model will be active.

Obtaining the maximum reachable set $\mathcal{T}_j$ for the sector $\mathcal{D}_j$ via solving (15) can be viewed as a collection of polytope operations. Define the partial reachable set for the model i in the region $\mathcal{D}_j$ by

$$
\mathcal{Q}_{ji} = \{x_{k+1} \mid x_{k+1} = A_i x_k + B_i u_k,\ x_k \in \mathcal{D}_j,\ u_k \in \mathcal{U}\}\,. \tag{16}
$$

Consequently, the maximum reachable set for the region $\mathcal{D}_j$ can be found as the bounding box of the union of the partial sets, i.e.

$$
\mathcal{T}_j = \text{Bbox}\left(\bigcup_{i=1}^{n_n} \mathcal{Q}_{ji}\right) \tag{17}
$$

where the operator Bbox is defined as follows:

Definition 1. *[17] A bounding box Bbox(P) of a set P is the smallest hyper-rectangle which contains the set P. If P is defined as a (possibly) non-convex union of convex polytopes P_i, i.e. $P = \bigcup_i P_i$, then the bounding box can be computed by solving $2n$ linear programs per each element of the set P. Here, n denotes the dimension of P.*

The maximum estimated reachable set $\hat{\mathcal{T}}_j$ can be computed similarly as a bounding box of the reachable sets for the mean model (8):

$$
\hat{\mathcal{T}}_j = \text{Bbox}\left(\hat{\mathcal{Q}}_j\right) \tag{18}
$$

with

$$
\hat{\mathcal{Q}}_j = \left\{x_{k+1} \mid x_{k+1} = \hat{A}_j x_k + \hat{B}_j u_k,\ x_k \in \mathcal{D}_j,\ u_k \in \mathcal{U}\right\}. \tag{19}
$$

The affine terms f_j of (8) can now be computed as a difference between the analytic centers of the reachable sets for the "true" and for the "approximated" system:

$$
f_j = \text{ce}(\mathcal{T}_j) - \text{ce}(\hat{\mathcal{T}}_j) \tag{20}
$$

where the operator *ce* is given by

$$\mathrm{ce}(\mathcal{T}) = \overline{x} - \frac{\overline{x} - \underline{x}}{2}. \tag{21}$$

Graphically are these sets depicted in Fig. 3a. It can be seen in Fig. 3b that the transformation procedure shifts these sets to one common analytic center. The allowable disturbance is then selected as the maximum distance over the edges of the sets in the sector $\mathcal{D}_j$, i.e.

$$w_j = \begin{cases} \max \left(\mathcal{T}_j - \left(\hat{\mathcal{T}}_j + f_j\right)\right) & \text{if} \quad \mathcal{T}_j \geq \hat{\mathcal{T}}_j + f_j \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

In other words, if the approximated reachable set $\hat{\mathcal{T}}_j$, shifted by the offset f_j, is smaller than the original reachable set $\mathcal{T}_j$, then the difference is modeled by an unknown, but bounded disturbance w_j, whose element-wise bounds are given by (22). By applying the same procedure to each sector $\mathcal{D}_j$ the original fuzzy model (2) can be converted to a PWA description (8). In the next section the pattern will be demonstrated on an illustrative example.

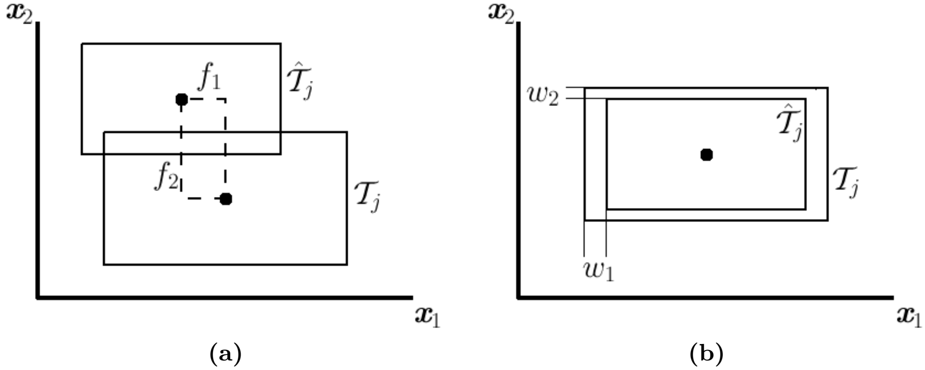


Fig. 3. The transformation procedure shifts the reachable sets to one common centre

4 Example

Consider a TS model (1) described by two linear dynamics

$$A_1 = \begin{pmatrix} 0.3216 & 0.0114 \\ 0.0864 & -0.8143 \end{pmatrix}, B_1 = \begin{pmatrix} -0.5867 \\ 0.5451 \end{pmatrix}$$
$$A_2 = \begin{pmatrix} 0.5331 & -0.7570 \\ -0.0404 & -0.2694 \end{pmatrix}, B_2 = \begin{pmatrix} -0.2836 \\ 0.6453 \end{pmatrix}$$

associated with the following membership functions

$$\mu_1(x_1) = \begin{cases} \mu_1(x_1) = 0 & \text{if} \quad |x_1| \geq 1.5 \\ \mu_1(x_1) = 1 - \frac{2}{3}|x_1| & \text{otherwise} \end{cases}$$

$$\mu_2(x_1) = \begin{cases} \mu_2(x_1) = 0 & \text{if} \quad |x_1| \leq 1.0 \\ \mu_2(x_1) = -\frac{1}{2} + \frac{1}{2}|x_1| & \text{otherwise} \end{cases}$$

The functions are depicted in Fig. 4. Constraints imposed for this example are the closed intervals

$$u \in [-5,\ 5], \quad x \in [-3,\ 3] \times [-2,\ 2]. \tag{23}$$

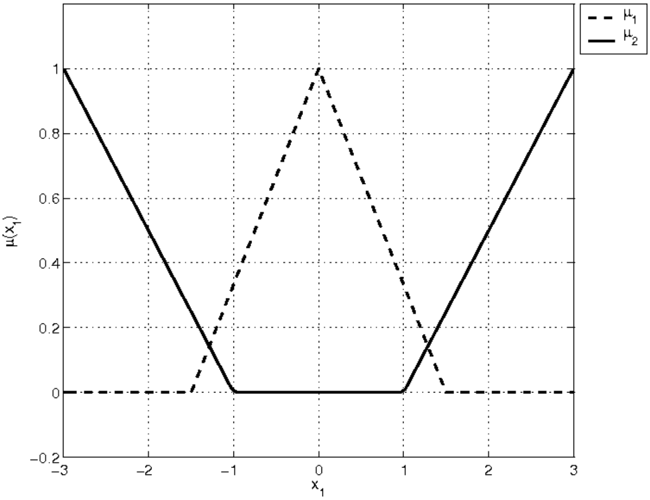


Fig. 4. Membership functions

To convert a given TS model into the PWA form (8), the feasible region (23) is first decomposed into 5 intervals given by following polytopes:

$$\mathcal{D}_1 := \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix} x \leq \begin{pmatrix} 3 \\ -1.5 \end{pmatrix}, \mathcal{D}_2 := \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix} x \leq \begin{pmatrix} 1.5 \\ -1 \end{pmatrix}, \tag{24}$$

$$\mathcal{D}_3 := \quad \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix} x \leq \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \mathcal{D}_4 := \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix} x \leq \begin{pmatrix} -1 \\ 1.5 \end{pmatrix},$$

$$\mathcal{D}_5 := \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix} x \leq \begin{pmatrix} -1.5 \\ 3 \end{pmatrix}.$$

The polytopes (24) have been selected following the procedure illustrated in Fig. 2. The PWA model takes the affine form

$$x_{k+1} = \begin{cases} A_2 x_k + B_2 u_k + w & \text{if} \quad x \in \mathcal{D}_1 \\ (0.5A_2 + 0.5A_1)x_k + (0.5B_2 + 0.5B_1)u_k + f_2 + w & \text{if} \quad x \in \mathcal{D}_2 \\ A_1 x_k + B_1 u_k + w & \text{if} \quad x \in \mathcal{D}_3 \\ (0.5A_1 + 0.5A_2)x_k + (0.5B_1 + 0.5B_2)u_k + f_4 + w & \text{if} \quad x \in \mathcal{D}_4 \\ A_2 x_k + B_2 u_k + w & \text{if} \quad x \in \mathcal{D}_5 \end{cases} \tag{25}$$

Important to notice is that modes 2 and 4 (which are active in sectors $\mathcal{D}_2$ and $\mathcal{D}_4$) are averaged due to overlapping membership functions. Using reachability analysis and computing $\mathcal{T}_j$ as per (17) we got

$$\mathcal{T}_2 = \begin{bmatrix} -3.7317, & 2.6347 \\ -4.4838, & 4.2677 \end{bmatrix}, \quad \mathcal{T}_4 = \begin{bmatrix} -2.6347, & 3.7317 \\ -4.2677, & 4.4838 \end{bmatrix}. \tag{26}$$

The maximum approximated reachable sets $\hat{\mathcal{T}}_j$ can be computed using (18) and are given by following axis-aligned intervals:

$$\hat{\mathcal{T}}_2 = \begin{bmatrix} -3.5624, 2.4940 \\ -4.0942, 4.0367 \end{bmatrix}, \quad \hat{\mathcal{T}}_4 = \begin{bmatrix} -2.4949, 3.5624 \\ -4.0367, 4.0942 \end{bmatrix}. \tag{27}$$

The affine terms f_j in (25), and the range for the maximum allowable disturbance w have been computed according to (20) and (22), respectively, as

$$f_2 = \begin{pmatrix} -0.0142 \\ -0.0792 \end{pmatrix}, \; f_4 = \begin{pmatrix} 0.0142 \\ 0.0792 \end{pmatrix}, \; \begin{pmatrix} -0.1551 \\ -0.3102 \end{pmatrix} \leq w \leq \begin{pmatrix} 0.1551 \\ 0.3102 \end{pmatrix}. \tag{28}$$

The final PWA model of the form (8) is then composed of (25) and (28), where the regions $\mathcal{D}_i$ over which each dynamics is active is given by (24).

5 Conclusion

In this paper a methodology of transforming fuzzy Takagi-Sugeno models into a Piecewise Affine representation has been presented. The approximation is based on deinterlacing the regions in which several membership functions overlap and subsequently approximating the effect of such overlaps by an unknown, but bounded disturbance. Computation of the bounds of the unknown disturbance is performed using reachability analysis. The resulting PWA model can then be used as a prediction model to derive MPC feedback laws with stability and feasibility guarantees. Since the PWA representation over-approximates the behavior of a given fuzzy Takagi-Sugeno model, the stability guarantees naturally extend to this class of models as well.

Acknowledgment. The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under grants No. 1/3081/06 and 1/4055/07.

References

1. Vasičkaninová, A., Bakošová, M.: Fuzzy modelling and identification of the chemical technological processes. In: Krejčí, S. (ed.): Proc. 7. Int. Scientific-Tehnical Conf. Process Control 2006, June 13-16 2006 (2006)
2. Allgöwer, F., Zheng, A. (ed.): Nonlinear Model Predictive Control. Birkhäuser (2000)
3. Bemporad, A., Morari, M.: Control of systems integrating logic, dynamics, and constraints. Automatica 35(3), 407–427 (1999)
4. Clarke, D.W., Mohtadi, C., Tuffs, P.S.: Generalized predictive control – Parts I-II. Automatica, 23(2) (1987)
5. Cutler, C.R., Ramaker, B.L: Dynamic matrix control – a computer control algorithm. In: Proceedings, Joint American Control Conference, San Francisco, California, USA (1980)
6. Espinosa, J.J., Hadjili, M., Wertz, V., Vandewalle, J.: Predictive Control Using Fuzzy Models-Comparative Study. In: Proc. of the European Control Conference, Karlsruhe, Germany, August, September 1999 (1999)
7. Grieder, P., Kvasnica, M., Baotic, M., Morari, M.: Stabilizing low complexity feedback control of constrained piecewise affine systems. Automatica 41(10), 1683–1694 (2005)
8. He, M., Cai, W.-J., Li, S.-Y.: Multiple fuzzy model-based temperature predictive control for HVAC systems. Information Sciences 169(1–2), 155–174 (2005)
9. Khaber, F., Zehar, K., Hamzaoui, A.: State Feedback Controller Design via Takagi-Sugeno Fuzzy Model: LMI Approach. International Journal of Computational Intelligence, 2(3) (2005)
10. Kosko, B.: Fuzzy systems as universal approximators. In: Proceedings FUZZ'IEEE'92, San Diego, California, USA, pp. 1153–1162 (1992)
11. Kouvaritakis, B., Cannon, M. (eds.): Nonlinear predictive control: theory and practice. IEE Control Engineering series (2001)
12. Li, J., Wang, H.O., Bushnell, L., Hong, Y.: A Fuzzy Logic Approach to Optimal Controlof Nonlinear Systems. International Journal of Fuzzy Systems 2(3), 153–163 (2000)
13. Li, N., Li, S.-Y., Xi, Y.-G.: Multi-model predictive control based on the Takagi-Sugeno fuzzy models: a case study. Information Sciences 165(3–4), 247–263 (2004)
14. Maciejowski, J.M.: Predictive Control with Constraints. Prentice-Hall, Englewood Cliffs (2002)
15. Prett, D.M., Garcia, C.E.: Fundamental Process Control. Butterworths, Boston (1988)
16. Raković, S.V., Grieder, P., Kvasnica, M., Mayne, D.Q., Morari, M.: Computation of Invariant Sets for Piecewise Affine Discrete Time Systems subject to Bounded Disturbances. In: Proceeding of the 43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas, December 2004, pp. 1418–1423 (2004)
17. Suard, R., Löfberg, J., Grieder, P., Kvasnica, M., Morari, M.: Efficient Computation of Controller Partitions in Multi-Parametric Programming. In: IEEE Conference on Decision and Control, Bahamas, December 2004 (2004)
18. Takagi, T., Sugeno, M.: Fuzzy identications of fuzzy systems and its applications to modelling and control. IEEE Trans. Systems Man and Cybernetics 15, 116–132 (1985)
19. Yoneyama, J.: Robust stability and stabilization for uncertain Takagi-Sugeno fuzzy time-delay systems. Fuzzy Sets. and Systems 158(2), 115–134 (2007)

Set Operations for L-Fuzzy Sets

Jouni Järvinen

Turku Centre for Computer Science (TUCS)
FI-20014 University of Turku, Finland
Jouni.Jarvinen@utu.fi

Abstract. In this paper, we introduce the operations of union, intersection, and complement for preorder-based fuzzy sets. The given operations are even capable of dealing with fuzzy sets that have membership degrees coming from different preordered sets. This enables us to handle the difficult situation in which one has different people giving judgements and they all like to use their own language and expressions.

1 Fuzzy Sets and L-Fuzzy Sets

Fuzzy sets were introduced in 1965 by Zadeh [9]. For a given universe of discourse U, a *fuzzy set* A on U is determined by a membership function $\mu_A\colon U \to [0,1]$ associating with each element $x \in U$ a real number $\mu_A(x)$ which represents the grade of membership of x in A.

Zadeh also introduced the set operations of union, intersection, and complementation for fuzzy sets. These operations are important because if one looks at the logical aspect of these operations, they represent 'or', 'and', and 'not'. The *union* of two fuzzy sets A and B is a fuzzy set whose membership function is $\mu_{A\cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$. Further, the *intersection* of the fuzzy sets A and B is a fuzzy set with the membership function $\mu_{A\cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$. The *complement* of the fuzzy set A is defined by $\mu_{A'}(x) = 1 - \mu_A(x)$. The above operations are often referred to as the *standard fuzzy set operations*, but in the literature one can find numerous different ways to define the set operations; see [7], for example.

The fundamental problem with fuzzy sets is that our perceptions have to be quantized to the unit interval. In this paper, our aim is to get rid of this semi-arbitrary choosing of the proper weighting scheme. We try to move towards the methodology, called *computing with words* [10], in which the objects of computation are given by a natural language. Computing with words, in general, is inspired by the human capability to perform a wide variety of tasks without any measurements and any quantizations.

Goguen generalized fuzzy sets to L-fuzzy sets in [3]. An *L-fuzzy set* φ on U is a mapping $\varphi\colon U \to L$, where L is a 'transitive partially ordered set'. In this work, we assume that $(L, \leq)$ is a preordered set. Notice that it is natural to assume that the relation $\leq$ is not antisymmetric; if $x, y \in L$ are synonyms, that is, words or expressions that are used with the same meaning, then $x \leq y$ and $x \geq y$, but still x and y are distinct words.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 221–229, 2007.

Example 1. Suppose that U consists of a group of people. The L-fuzzy set, whose membership function φ is depicted in Fig. 1, describes how well the persons in U can ski. For instance, there exist people who can ski very well, some ski badly, and some are moderate skiers.

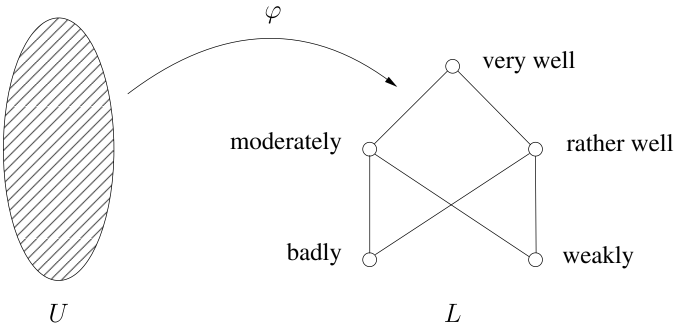


Fig. 1.

As noted by Goguen [3], the set of all L-fuzzy sets on a set U can be equipped whatever operations L has, and these inherited operations obey any law valid in L which extends pointwise. This implies that if L is, for example, a Boolean lattice, then also the set all of L-fuzzy sets on U forms a Boolean lattice. Formally, if φ and ψ are L-fuzzy sets on U, then for any $x \in U$,

$$\begin{aligned}(\varphi \vee \psi)(x) &= \varphi(x) \vee \psi(x)\\ (\varphi \wedge \psi)(x) &= \varphi(x) \wedge \psi(x)\\ \varphi'(x) &= \varphi(x)' .\end{aligned}$$

In this paper, we show how to define unions, intersections, and complements of L-fuzzy sets in cases L is just a preordered set, which means that joins, meets, and complements are not defined in L. The presented approach also handles the union and the intersection of an L_1-fuzzy set φ and an L_2-fuzzy set ψ on the same universe U, but not necessarily on the same preordered set. This means that we can, for example, combine with 'or', 'and', and 'not' judgements of evaluators all wanting to use their own words and expressions. Our key idea is that the order determined by membership values is essential, not the values themselves. It should be noted that some ideas presented in this work appear already in [4,5].

2 Preorders and Alexandrov Topologies

Preorders and Alexandrov topologies have a major role in this paper. Therefore, we begin with presenting some results concerning them. This section contains also many lattice-theoretical notions which can be found in [1,2,4], for example.

Let U be any set and let R be a binary relation on U. Then, the relation R is a *preorder*, if

(i) for all $x \in U$, $x \, R \, x$ (reflexive)
(ii) for all $x, y, z \in U$, $x \, R \, y$ and $y \, R \, z$ imply $x \, R \, z$ (transitive)

The pair $(U, \leq)$ is called a *preordered set*. Note that often we say simply that 'U is a preordered set'.

We may depict preorders by Hasse diagrams as in case of partially ordered sets. The only difference is that preorders are not necessarily antisymmetric, meaning that there may exist elements $x \neq y$ such that $x \leq y$ and $x \geq y$. However, such elements can simply be represented as collections of $\approx$-equivalent elements, where the equivalence $\approx$ is defined by

$$x \approx y \text{ if and only if } x \leq y \text{ and } x \geq y.$$

This means that 'synonymous' elements are represented by a same point in a Hasse diagram, but still they all preserve their identities.

Let us denote by $\mathrm{Pre}(U)$ the set of all preorders on the set U. The set $\mathrm{Pre}(U)$ can be ordered with the usual set-inclusion relation, because relations are just sets of ordered pairs. First we recall the following well-known lemma that is clear since the intersection of any subset of $\mathrm{Pre}(U)$ is a preorder. Note that generally the union of preorders is not a preorder.

Lemma 2. *For any set U, $\mathrm{Pre}(U)$ is a complete lattice with respect to the set-inclusion relation.*

Since $\mathrm{Pre}(U)$ is a *closure system*, that is, a family of sets closed under arbitrary intersections, we have that for any $\mathcal{H} \subseteq \mathrm{Pre}(U)$, the meet $\bigwedge \mathcal{H}$ is the intersection $\bigcap \mathcal{H}$ and the join $\bigvee \mathcal{H}$ is the intersection of all preorders including $\bigcup \mathcal{H}$. We will present another description of joins later in this section. Furthermore, the 'all relation' $\nabla = \{(x, y) \mid x, y \in U\}$ is the greatest element and the 'identity relation' $\Delta = \{(x, x) \mid x \in U\}$ is the least element of $\mathrm{Pre}(U)$.

A *topological space* is a pair $(U, \mathcal{T})$, where U is a set and $\mathcal{T}$ is a collection of subsets of U such that

(i) $\emptyset, U \in \mathcal{T}$;
(ii) for all $\mathcal{H} \subseteq \mathcal{T}$, $\bigcup \mathcal{H} \in \mathcal{T}$;
(iii) for all $X, Y \in \mathcal{T}$, $X \cap Y \in \mathcal{T}$.

The collection $\mathcal{T}$ is called a *topology*.

An *Alexandrov topology* is a topology $\mathcal{T}$ that contains also all arbitrary intersections of its members. This means that for Alexandrov topologies, condition (iii) is replaced by condition

(iii)° for all $\mathcal{H} \subseteq \mathcal{T}$, $\bigcap \mathcal{H} \in \mathcal{T}$.

The pair $(U, \mathcal{T})$ is referred to as an *Alexandrov space*.

Every Alexandrov topology $\mathcal{T}$ has the property that each point $x \in U$ has a *smallest neighbourhood* $N_{\mathcal{T}}(x) = \bigcap \{X \in \mathcal{T} \mid x \in X\}$. This means that $N_{\mathcal{T}}(x)$ is the smallest set in the topology $\mathcal{T}$ containing the point x.

Let us denote by $\mathrm{Alex}(U)$ the set of all Alexandrov topologies. Obviously, also $\mathrm{Alex}(U)$ can be ordered by the set-inclusion relation. Because the intersection of Alexandrov topologies is an Alexandrov topology, we may write the next lemma.

Lemma 3. *For any set U,* $\mathrm{Alex}(U)$ *is a complete lattice with respect to the set-inclusion relation.*

Clearly, $\mathrm{Alex}(U)$ is a closure system and hence for any $\mathcal{H} \subseteq \mathrm{Alex}(U)$, $\bigwedge \mathcal{H}$ is equal to the intersection $\bigcap \mathcal{H}$ and $\bigvee \mathcal{H}$ is the intersection of all Alexandrov topologies including $\bigcup \mathcal{H}$. In addition, the 'discrete topology' $\mathcal{T}_\Delta = \{X \mid X \subseteq U\}$ is the greatest element and the 'trivial topology' $\mathcal{T}_\nabla = \{\emptyset, U\}$ is the smallest element of $\mathrm{Alex}(U)$.

There is a close connection between preorders and Alexandrov topologies. Let $\leq$ be a preorder on a set U. We may now define an Alexandrov topology $\mathcal{T}_\leq$ on U consisting of all upward-closed subsets of U with respect to the relation $\leq$, that is,

$$\mathcal{T}_\leq = \{X \subseteq U \mid (\forall x, y \in U)\, x \in X \ \ \& \ \ x \leq y \Longrightarrow y \in X\}\,.$$

Let us denote for any $x \in U$, the principal filter of x by $\uparrow x = \{y \in U \mid x \leq y\}$. Now we can give the following lemma.

Lemma 4. *If $\leq$ is a preorder on U, then the following assertions hold for all $X \subseteq U$ and $x \in U$:*

(i) *$X \in \mathcal{T}_\leq$ if and only if $X = \bigcup\{\uparrow x \mid x \in X\}$;*
(ii) *$\uparrow x$ is the smallest neighbourhood of x in the Alexandrov topology $\mathcal{T}_\leq$.*

Proof. (i) Assume that $X \in \mathcal{T}_\leq$. If $x \in X$, then $x \leq x$ gives $x \in \uparrow x$. Thus, $X \subseteq \bigcup\{\uparrow x \mid x \in X\}$. On the other hand, if $y \in \bigcup\{\uparrow x \mid x \in X\}$, then there exists $x \in X$ such that $x \leq y$. Since $X \in \mathcal{T}_\leq$, we obtain $y \in X$. Hence, also $\bigcup\{\uparrow x \mid x \in X\} \subseteq X$.

Conversely, suppose $X = \bigcup\{\uparrow x \mid x \in X\}$, $x \in X$, and $x \leq y$. Then $y \in \uparrow x$ and so $y \in X$. Therefore, X is upward closed and $X \in \mathcal{T}_\leq$.

(ii) It is clear that $x \in \uparrow x \in \mathcal{T}_\leq$ and if $x \in X \in \mathcal{T}_\leq$, then $\uparrow x \subseteq X$ by (i). □

By the above lemma, $\uparrow x$ is the smallest neighbourhood of the point x in the Alexandrov topology $\mathcal{T}_\leq$ and clearly $y \in \uparrow x$ if and only if $x \leq y$. This hints how we may also define preorders by means of Alexandrov topologies. If $\mathcal{T}$ is an Alexandrov topology on U, then we define a preorder $\leq_\mathcal{T}$ on U by setting

$$x \leq_\mathcal{T} y \iff y \in N_\mathcal{T}(x).$$

The following theorem by Steiner [8] is essential for our studies.

Theorem 5. *For any set U, the complete lattice* $\mathrm{Pre}(U)$ *of all preorders on U is dually isomorphic to* $\mathrm{Alex}(U)$*, the complete lattice of all Alexandrov topologies on U; in symbols* $(\mathrm{Pre}(U), \subseteq) \cong (\mathrm{Alex}(U), \supseteq)$.

A nice property of set unions and intersections is that they distribute over each other. Therefore, it is a natural question to ask whether joins and meets defined in a particular lattice have analogous properties. Formally, a lattice L is *distributive* if it satisfies either (and therefore both) of the distributive laws:

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$
$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z).$$

Furthermore, L is *modular* if

$$x \leq z \Longrightarrow x \vee (y \wedge z) = (x \vee y) \wedge z.$$

Trivially, each distributive lattice is modular.

Steiner noted that $\mathrm{Pre}(U)$ and $\mathrm{Alex}(U)$ are distributive if U has fewer than three elements. If U has three or more elements, $\mathrm{Pre}(U)$ and $\mathrm{Alex}(U)$ are not even modular.

Next we present a simpler way to determine the joins in $\mathrm{Pre}(U)$ and $\mathrm{Alex}(U)$. Recall that in $(\mathrm{Alex}(U), \subseteq)$, the meet is the intersection of Alexandrov topologies. Thus, the join in its dual $(\mathrm{Alex}(U), \supseteq)$ is the intersection of Alexandrov topologies, that is,

$$\mathcal{T}_1 \vee \mathcal{T}_2 \ = \ \mathcal{T}_1 \cap \mathcal{T}_2.$$

By Theorem 5, $(\mathrm{Pre}(U), \subseteq) \cong (\mathrm{Alex}(U), \supseteq)$, which implies that in $(\mathrm{Pre}(U), \subseteq)$,

$$\leq_1 \vee \leq_2 \ = \ \leq_{(\mathcal{T}_1 \cap \mathcal{T}_2)},$$

where $\mathcal{T}_1$ and $\mathcal{T}_2$ are the Alexandrov topologies determined by $\leq_1$ and $\leq_2$. Similarly, in $(\mathrm{Alex}(U), \subseteq)$,

$$\mathcal{T}_1 \vee \mathcal{T}_2 = \mathcal{T}_{(\sqsubseteq_1 \cap \sqsubseteq_2)},$$

where $\sqsubseteq_1$ and $\sqsubseteq_2$ are the preorders of $\mathcal{T}_1$ and $\mathcal{T}_2$.

Next we study complementation in these isomorphic lattices. A *lattice-complement* of a preorder R is a preorder R' such that $R \vee R' = \nabla$ and $R \wedge R' = \Delta$. The next important theorem is also proved by Steiner [8].

Theorem 6. *The lattice* $\mathrm{Pre}(U)$ *is complemented.*

It is trivial that the set-theoretical complement R^c of a preorder R cannot serve as the lattice-theoretical complement, because R^c is not a preorder and $R \wedge R^c = \emptyset \neq \Delta$. Next we describe the lattice-theoretical complement R' of R in $\mathrm{Pre}(U)$. Let R^E be the smallest equivalence including R. Further, let $\{X_i \mid i \in I\}$ be the set of equivalence classes of R^E. By the Axiom of Choice we may pick an element from each equivalence class. Let us denote the representative of the class X_i by x_i. Next we derive two new relations R_1 and R_2 from R by setting

$$R_1 = \{(y, x) \mid x\,R\,y \ \&\ (y, x) \notin R\} \cup \Delta$$

and

$$R_2 = \{(x_i, x_j) \mid i, j \in I\} \cup \Delta.$$

It is easy to see that R_1 and R_2 are preorders. The lattice-theoretical complement R' of R is defined by

$$R' = R_1 \vee R_2.$$

It is known that if a lattice is distributive, the complements – if they exist – are unique. We have already mentioned that $\mathrm{Pre}(U)$ is not distributive when $|U| \geq 3$. This implies that the complements are not necessarily unique. Namely, if R is a preorder such that R^E has at least two equivalence classes of which at least one is non-singleton, then the complement of R depends on the choice function $U/R^E \to U$. On the other hand, if R^E has only one equivalence class U, then $R_2 = \mathit{\Delta}$ and the complement of R is R_1 which clearly is unique. In such a case, the Hasse diagram of R' is just the Hasse diagram of R turned upside down with its equivalent elements being separated. Note also that R^E has only one equivalence class if and only if R is *connected*, that is, for any $x, y \in U$, there exists a sequence $a_0, a_1, \ldots, a_n$ of elements of U such that $a_0 = x$, $a_n = y$, and $a_i \, R \, a_{i+1}$ or $a_{i+1} \, R \, a_i$ for $i = 0, \ldots, n-1$.

We end this section by noting that Theorem 6 has the following obvious corollary.

Corollary 7. *The lattice* $\mathrm{Alex}(U)$ *is complemented.*

3 *L*-Fuzzy Sets and Their Operations

In this section our aim is to define set operations for L-fuzzy sets.

Let U be a set and let L be an arbitrary preordered set. Any L-fuzzy set φ on U determines naturally a preorder on U, as suggested by Kortelainen in [6]. A preorder $\lesssim_\varphi$ is defined by setting for all $x, y \in U$,

$$x \lesssim_\varphi y \iff \varphi(x) \leq \varphi(y).$$

By Theorem 5 there is one-to-one correspondence between preorders and Alexandrov topologies on U. This implies directly that each L-fuzzy set induces also an Alexandrov topology $\mathcal{T}_\varphi$ consisting of upward-closed subsets of $\lesssim_\varphi$. Let us denote the principal filter $\uparrow\! x$ of x with respect to the preorder $\lesssim_\varphi$ by $N_\varphi(x)$, that is, $N_\varphi(x) = \{y \mid \varphi(x) \leq \varphi(y)\}$. By Lemma 4, it is clear that

$$X \in \mathcal{T}_\varphi \iff X = \bigcup \{N_\varphi(x) \mid x \in U\}$$

and $N_\varphi(x)$ is the smallest neighbourhood of x in the Alexandrov topology $\mathcal{T}_\varphi$.

Next we show how Alexandrov topologies determine fuzzy sets. Let $\mathcal{T}$ be an Alexandrov topology on a set U. Let us denote by $\mathcal{T}^{\mathrm{op}}$ the ordered set $(\mathcal{T}, \supseteq)$. Now the mapping

$$\varphi_{\mathcal{T}} \colon U \to \mathcal{T}^{\mathrm{op}}, \ x \mapsto N_{\mathcal{T}}(x)$$

is a $\mathcal{T}^{\mathrm{op}}$-fuzzy set. It is also easy to observe that if φ is an L-fuzzy set on U, then $\varphi^* \colon U \to {\mathcal{T}_\varphi}^{\mathrm{op}}, \ x \mapsto N_\varphi(x)$ is a fuzzy set such that the preorder $\lesssim_\varphi$ of φ is equal to the preorder $\lesssim_{\varphi^*}$ determined by φ^*. Furthermore, $\varphi^{**} = \varphi^*$. Thus, φ^* can be identified as a *canonical representation* of φ, as is done in [5].

Let us denote by $\mathrm{Fuzzy}(U)$ the class of all fuzzy sets on U, that is, the collection of all such mappings $\varphi \colon U \to L$ that L is any arbitrary preordered set. We noted in the previous section that $(\mathrm{Pre}(U), \subseteq)$ is a complemented lattice. Because each element in Fuzzy(U) determines a unique preorder, we may now define the *union*, the *intersection*, and the *complement* for any elements $\varphi\colon U \to L_1$ and $\psi\colon U \to L_2$ of Fuzzy(U) as follows:

$$\begin{aligned} \varphi \cup \psi \ &:= \ \lesssim_\varphi \vee \lesssim_\psi && (1)\\ \varphi \cap \psi \ &:= \ \lesssim_\varphi \wedge \lesssim_\psi && (2)\\ \varphi^c \ &:= \ {\lesssim_\varphi}' . && (3)\end{aligned}$$

Note that there always exists a fuzzy set in Fuzzy(U) corresponding to the results of these operations. For example, let us consider the union $\varphi \cup \psi$. As we have shown, the Alexandrov topology $\mathcal{T}_{\varphi\cup\psi}$ determines a fuzzy set

$$(\varphi \cup \psi)^*\colon U \to {\mathcal{T}_{\varphi\cup\psi}}^{\mathrm{op}},\ x \mapsto N_{(\varphi\cup\psi)}(x).$$

Using preorders as results of set operations is useful also because in applications we are often interested in the order of elements with respect to aggregation of some criteria.

Example 8. Assume that $U = \{x, y, z, w\}$ consists of four applicants of a certain academic position and that $\varphi\colon U \to L_1$ and $\psi\colon U \to L_2$ represent how two experts evaluate the suitability of the applicants by using some expressions and attributes L_1 and L_2 of their own languages. The fuzzy sets φ and ψ are given in Fig. 2 of page 228. The induced preorders are

$$\lesssim_\varphi = \{(y,x),(z,x),(w,x),(w,y),(w,z)\} \cup \Delta$$

and

$$\lesssim_\psi = \{(w,x),(z,x),(z,y)\} \cup \Delta.$$

These preorders and the canonical representations $\varphi^*\colon U \to \mathcal{T}_\varphi^{\mathrm{op}}$ and $\psi^*\colon U \to \mathcal{T}_\psi^{\mathrm{op}}$ are also depicted in Fig. 2. We define the union, the intersection, and the complements as described in (1)–(3). The results of these operations can be found in Fig. 2 as well.

Now $\varphi \cap \psi$ can be viewed as an order that takes into account the opinions of both the experts. The applicants x and y must be considered as suitable for the open position, but z and w should not be selected, since the both experts have the opinion that they are weaker than x. Let us consider the applicants in the view of the union $\varphi \cup \psi$. According to it, the applicant x should be chosen, because there exists one expert evaluating x as the best candidate, and this is not true for the others. The complements φ^c and ψ^c can be considered as orders totally opposite to the opinions of the expert.

Notice also that the De Morgan laws do not hold, because

$$\varphi^c \cap \psi^c \neq (\varphi \cup \psi)^c \ \text{ and } \ \varphi^c \cup \psi^c \neq (\varphi \cap \psi)^c.$$

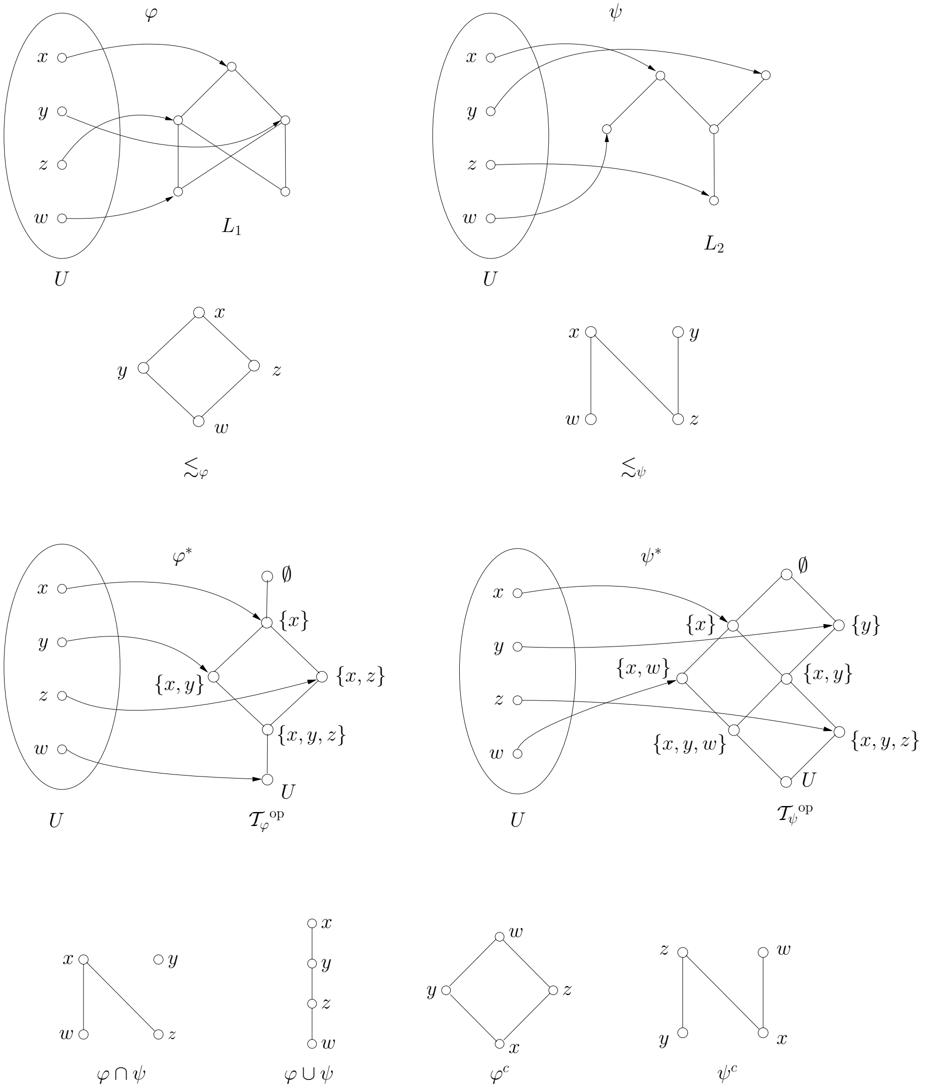


Fig. 2.

Some Concluding Remarks and Acknowledgements

In this paper we have introduced unions, intersections and complements for preorder-based fuzzy sets on a given universe U. Our work was based on the observation that each preorder-based fuzzy set determines a preorder and an Alexandrov topology on U. We have described how the results of these set operations can be easily formed. Importantly, the presented approach can handle the union and the intersection of an L_1-fuzzy set φ and an L_2-fuzzy set ψ of the universe U also in the case L_1 and L_2 are different preordered sets. This enables us to cope with the common situation in which one has different people giving judgements and they all like to use their own language and expressions.

The author thanks the anonymous referees for their comments and suggestions that helped to improve the paper.

References

1. Birkhoff, G.: Lattice Theory, Colloquim publications, 3rd edn. vol. XXV, American Mathematical Society (AMS), Providence, Rhode Island (1995)
2. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order, 2nd edn. Cambridge University Press, Cambridge (2002)
3. Goguen, J.A.: L-fuzzy sets. Journal of Mathematical Analysis and Applications 18, 145–174 (1967)
4. Järvinen, J.: Lattice theory for rough sets. In: Transactions on Rough Sets VI. LNCS, vol. 4374, pp. 400–498. Springer, Heidelberg (2007)
5. Järvinen, J., Kortelainen, J.: A unifying study between modal-like operators, topologies, and fuzzy sets. Fuzzy Sets and Systems 158, 1217–1225 (2007)
6. Kortelainen, J.: On relationship between modified sets, topological spaces and rough sets. Fuzzy Sets and Systems 61, 91–95 (1994)
7. Lowen, R.: Fuzzy Set Theory: Basic Concepts, Techniques and Bibliography. Kluwer Academic Publishers, Norwell, MA, USA (1996)
8. Steiner, A.K.: The lattice of topologies: structure and complementation. Transactions of the American Mathematical Society 122, 379–398 (1966)
9. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)
10. Zadeh, L.A.: From computing with numbers to computing with words – from manipulation of measurements to manipulation of perceptions. International Journal of Applied Mathematics and Computer Science 12, 307–324 (2002)

Linguistic Summarization of Time Series Under Different Granulation of Describing Features

Janusz Kacprzyk, Anna Wilbik, and Sławomir Zadrożny

Systems Research Institute, Polish Academy of Sciences
ul. Newelska 6, 01-447 Warsaw, Poland
{kacprzyk, wilbik, zadrozny}@ibspan.waw.pl

Abstract. We consider an extension to a new approach to the linguistic summarization of time series data proposed in our previous papers. We summarize trends identified here with straight segments of a piecewise linear approximation of time series. Then we employ, as a set of features, the duration, dynamics of change and variability, and assume different, human consistent granulations of their values. The problem boils down to a linguistic quantifier driven aggregation of partial trends that is done via the classic Zadeh's calculus of linguistically quantified propositions but with different *t*-norms. We show an application to linguistic summarization of time series data on daily quotations of an investment fund over an eight year period.

1 Introduction

A linguistic data (base) summary, meant as a concise, human-consistent description of a (numerical) data set, was introduced by Yager [18] and then further developed by Kacprzyk and Yager [11], and Kacprzyk, Yager and Zadrożny [12]. The contents of a database is summarized via a natural language like expression semantics provided in the framework of Zadeh's calculus of linguistically quantified propositions [21]. Since data sets are usually large, it is very difficult for a human being to capture and understand their contents. As natural language is the only fully natural means of articulation and communication for a human being, such linguistic descriptions are the most human consistent.

In this paper we consider a specific type of data, namely time series. In this context it might be good to obtain a brief, natural language like description of trends present in the data on, e.g., stock exchange quotations, sales, etc. over a certain period of time.

Though statistical methods are widely used, we wish to derive (quasi)natural language descriptions to be considered to be an additional form of data description of a remarkably high human consistency. Hence, our approach is not meant to replace the classical statistical analyses but to add a new quality.

The summaries of time series we propose refer in fact to the summaries of trends identified here with straight line segments of a piece-wise linear approximation of time series. Thus, the first step is the construction of such an approximation. For this purpose we use a modified version of the simple, easy to use Sklansky and Gonzalez algorithm presented in [16].

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 230–240, 2007.

Then we employ a set of features (attributes) to characterize the trends such as the slope of the line, the fairness of approximation of the original data points by line segments and the length of a period of time comprising the trend.

Basically the summaries proposed by Yager are interpreted in terms of the number or proportion of elements possessing a certain property. In the framework considered here a summary might look like: "Most of the trends are short" or in a more sophisticated form: "Most long trends are increasing". Such expressions are easily interpreted using Zadeh's calculus of linguistically quantified propositions. The most important element of this interpretation is a linguistic quantifier exemplified by "most". In Zadeh's [21] approach it is interpreted in terms of a proportion of elements possessing a certain property (e.g., a length of a trend) among all the elements considered (e.g., all trends).

In Kacprzyk, Wilbik and Zadrożny [6] we proposed to use Yager's linguistic summaries, interpreted in the framework of Zadeh's calculus of linguistically quantified propositions, for the summarization of time series. In our further papers (cf. Kacprzyk, Wilbik and Zadrożny [8,9,10]) we extended this idea by proposing other types of summaries and the use of other mathods, notably the Choquet and Sugeno integrals. All these approaches have been proposed using a unified perspective given by Kacprzyk and Zadrożny [13] that is based on Zadeh's [22] protoforms.

In this paper we employ the classic Zadeh's calculus of linguistically quantified propositions. However, we will extend the idea proposed in our source paper (Kacprzyk, Wilbik and Zadrożny [6]) by using various t-norms and show results of an application to data on daily quotations of a mutual (investment) fund over an eight year period.

The paper is in line with some modern approaches to a human consistent summarization of time series – cf. Batyrshin and his collaborators [1,2], or Chiang, Chow and Wang [4] but we use a different approach.

One should mention an interesting project coordinated by the University of Aberdeen, UK, `SumTime`, an EPSRC Funded Project for Generating Summaries of Time Series Data[1]. Its goal is also to develop a technology for producing English summary descriptions of a time-series data set using an integration of time-series and natural language generation technology. Linguistic summaries obtained related to wind direction and speed are, cf. Sripada et al. [17]:

- WSW (West of South West) at 10-15 knots increasing to 17-22 knots early morning, then gradually easing to 9-14 knots by midnight,
- During this period, spikes simultaneously occur around 00:29, 00:54, 01:08, 01:21, and 02:11 (o'clock) in these channels.

They do provide a higher human consistency as natural language is used but they capture imprecision of natural language to a very limited extent. In our approach this will be overcome to a considerable extent.

[1] `www.csd.abdn.ac.uk/research/sumtime/`

2 Temporal Data and Trend Analysis

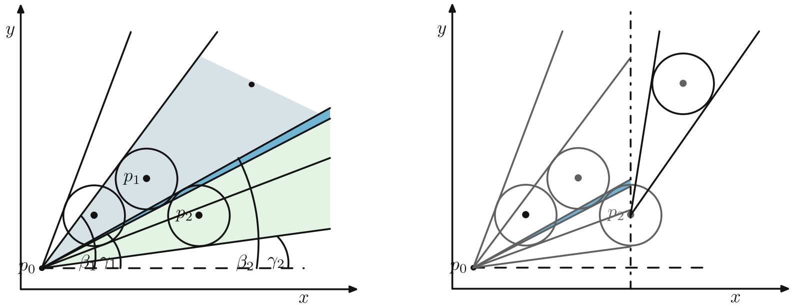
We identify trends as linearly increasing, stable or decreasing functions, and therefore represent given time series data as piecewise linear functions of some slope (intensity of an increase and decrease). These are partial trends as a global trend concerns the entire time span. There also may be trends that concern more than a window taken into account while extracting partial trends by using the Sklansky and Gonzalez [16] algorithm.

We use the concept of a uniform partially linear approximation of a time series. Function f is a uniform ε-approximation of a set of points $\{(x_i, y_i)\}$, if for a given, context dependent $\varepsilon > 0$, there holds

$$\forall i: \ |f(x_i) - y_i| \leq \varepsilon \tag{1}$$

and if f is linear, then such an approximation is a linear uniform ε-approximation.

We use a modification of the well known Sklansky and Gonzalez [16] algorithm that finds a linear uniform ε-approximation for subsets of points of a time series. The algorithm constructs the intersection of cones starting from point p_i of the time series and including a circle of radius ε around the subsequent points p_{i+j}, $j = 1, 2, \ldots$, until the intersection of all cones starting at p_i is empty. If for p_{i+k} the intersection is empty, then we construct a new cone starting at p_{i+k-1}. Figures 1(a) and 1(b) present the idea of the algorithm. The family of possible solutions is indicated as a gray area. For other algorithms, see,e.g., [15].



(a) the intersection of the cones is indicated by the dark grey area

(b) a new cone starts in point p_2

Fig. 1. An illustration of the algorithm for the uniform ε-approximation

First, denote:`p_0` – a point starting the current cone, `p_1` – the last point checked in the current cone, `p_2` – the next point to be checked, `Alpha_01` – a pair of angles (γ_1, β_1), meant as an interval, that defines the current cone as in Fig. 1(a), `Alpha_02` – a pair of angles of the cone starting at `p_0` and inscribing the circle of radius ε around `p_2` (cf. (γ_2, β_2) in Fig. 1(a)), function `read_point()` reads a next point of data series, function `find()` finds a pair of

```
read_point(p_0);
read_point(p_1);
while(1)
{
  p_2=p_1;
  Alpha_02=find();
  Alpha_01=Alpha_02;
  do
  {

    Alpha_01 = Alpha_01 $\cap$ Alpha_02;

    p_1=p_2;
    read_point(p_2);
    Alpha_02=find();

  } while(Alpha_01 $\cap$ Alpha_02 $\neq \emptyset$);

  save_found_trend();
  p_0=p_1;
  p_1=p_2;
}
```

Fig. 2. Pseudocode of the modified Sklansky and Gonzalez [16] algorithm for extracting trends

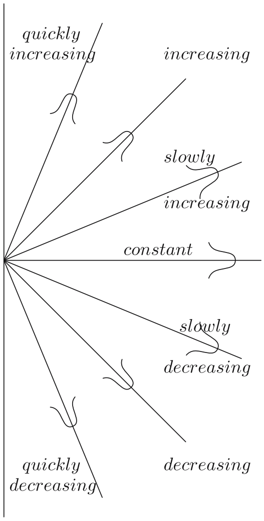


Fig. 3. A visual representation of angle granules defining the dynamics of change

angles of the cone starting at p_0 and inscribing the circle of radius ε around p_2. Then, a pseudocode of the algorithm that extracts trends is given in Fig. 2.

The bounding values of Alpha_02 (γ_2, β_2), computed by function find() correspond to the slopes of two lines tangent to the circle of radius ε around $p_2 = (x_2, y_2)$ and starting at $p_0 = (x_0, y_0)$. Thus, if $\mathit{\Delta} x = x_0 - x_2$ and $\mathit{\Delta} y = y_0 - y_2$ then:

$$\gamma_2 = arctg\left[\left(\mathit{\Delta} x \cdot \mathit{\Delta} y \pm \varepsilon\sqrt{(\mathit{\Delta} x)^2 + (\mathit{\Delta} y)^2 - \varepsilon^2}\right) / \left((\mathit{\Delta} x)^2 - \varepsilon^2\right)\right]$$

The resulting linear ε-approximation of a group of points p_0, ... ,p_1 is either a single segment, chosen as, e.g., a bisector of the cone, or one that minimizes the distance (e.g., the sum of squared errors, SSE) from the approximated points, or the whole family of possible solutions, i.e., the rays of the cone.

3 Dynamic Characteristics of Trends

While summarizing trends in time series data, we consider the following three aspects: (1) dynamics of change, (2) duration, and (3) variability, and by trends we mean here global trends, concerning the entire time series (or some, probably a large, part of it), not partial trends concerning in the (partial) trend extraction phase via the Sklansky and Gonzales [16] algorithm. In what follows we will briefly discuss these factors.

Dynamics of change

By *dynamics of change* we understand the speed of changes. It can be described by the slope of a line representing the trend, (cf. any angle η from the interval $\langle \gamma, \beta \rangle$ in Fig. 1(a)). Thus, to quantify dynamics of change we may use the interval of possible angles $\eta \in \langle -90; 90 \rangle$.

For practical reasons, we use a fuzzy granulation via a scale of linguistic terms as, e.g.: *quickly decreasing*, *decreasing*, *slowly decreasing*, *constant*, *slowly increasing*, *increasing*, *quickly increasing*, as illustrated in Fig. 3. Batyrshin et al. [1,2] give some methods for constructing such a fuzzy granulation.

We map a single value α (or the interval of angles corresponding to the gray area in Fig. 1(b)) characterizing the dynamics of change into a fuzzy set (linguistic label) best matching a given angle, and we can say that a given trend is, e.g., "decreasing to a degree 0.8".

Duration

Duration describes the length of a single trend, meant as a linguistic variable and exemplified by a "long trend" defined as a fuzzy set.

Variability

Variability refers to how "spread out" (in the sense of values) a group of data is. Traditionally, the following five statistical measures of variability are widely used:

- The range (maximum – minimum).
- The interquartile range (IQR) calculated as the third quartile (the 75th percentile) minus the first quartile (the 25th percentile) that may be interpreted as representing the middle 50% of the data.
- The variance is calculated as $\sum_i (x_i - \bar{x})^2/n$, where $\bar{x}$ is the mean value.
- The standard deviation – a square root of the variance.
- The mean absolute deviation (MAD), calculated as $\sum_i |x_i - \bar{x}|/n$.

We measure the variability of a trend as the distance of the data points from its linear uniform ε-approximation (cf. Section 2). We propose to employ a distance between a point and a family of possible solutions, indicated as a gray cone in Fig. 1(a). Equation (1) assures that the distance is definitely smaller than ε. The normalized distance equals 0 if the point lays in the gray area and otherwise is equal to the distance to the nearest point belonging to the cone, divided by ε.

Then, we find for a given value of variability obtained a best matching fuzzy set (linguistic label).

4 Linguistic Data Summaries

A linguistic summary is meant as a (short) natural language like sentence(s) that subsumes the very essence of a (numeric, usually large) set of data (cf. Kacprzyk and Zadrożny [13], [14]). In Yager's approach (cf. Yager [18], Kacprzyk and Yager [11], and Kacprzyk, Yager and Zadrożny [12]) the following perspective for linguistic data summaries is assumed:

- $Y = \{y_1, \ldots, y_n\}$ is a set of objects in a database, e.g., the set of workers;
- $A = \{A_1, \ldots, A_m\}$ is a set of attributes characterizing objects from Y, e.g., salary, and $A_j(y_i)$ is a value of attribute A_j for object y_i.

A linguistic summary of a data set consists of:

- a summarizer P, i.e. an attribute together with a linguistic value (fuzzy predicate) defined on the domain of attribute A_j (e.g. "low" for attribute "salary");
- a quantity in agreement Q, i.e. a linguistic quantifier (e.g. most);
- truth (validity) $\mathcal{T}$ of the summary, i.e. a number from the interval $[0, 1]$ assessing the truth (validity) of the summary (e.g. 0.7); usually, only summaries with a high value of $\mathcal{T}$ are interesting;
- optionally, a qualifier R, i.e. another attribute together with a linguistic value (fuzzy predicate) defined on the domain of attribute A_k determining a (fuzzy subset) of Y (e.g. "young" for attribute "age").

Thus, a linguistic summary may be exemplified by

$$\mathcal{T}(\text{most of employees earn low salary}) = 0.7 \tag{2}$$

or, in a richer (extended) form, including a qualifier (e.g. young), by

$$\mathcal{T}(\text{most of young employees earn low salary}) = 0.9 \tag{3}$$

Thus, basically, the core of a linguistic summary is a *linguistically quantified proposition* in the sense of Zadeh [21] which, for (2) and (3), respectively, may be written as

$$Qy\text{'s are } P \qquad\qquad QRy\text{'s are } P \tag{4}$$

Then, $\mathcal{T}$ directly corresponds to the truth value of (4) that may be calculated by Zadeh's calculus of linguistically quantified propositions (cf. [21] or the next section), or other interpretations of linguistic quantifiers (cf. [7]).

5 Protoforms of Linguistic Trend Summaries

As shown by Kacprzyk and Zadrożny [13], Zadeh's [22] concept of a protoform is convenient for dealing with linguistic summaries. A protoform is defined as a more or less abstract prototype (template) of a linguistically quantified proposition. Then, the summaries mentioned above may be represented by two types of the protoforms:

- a protoform of a short form of linguistic summaries:

$$Q \text{ trends are } P \tag{5}$$

and exemplified by: *Most* of trends are of a *large variability*

– a protoform of an extended form of linguistic summaries:

$$QR \text{ trends are } P \tag{6}$$

and exemplified by: *Most* of *slowly decreasing trends* are of a *large variability*

Their truth values will be found using the classic Zadehs calculus of linguistically quantified propositions as it is effective and efficient, and provides the best conceptual framework for a linguistic quantifier driven aggregation of partial trends.

6 The Use of Zadeh's Calculus

Using Zadeh's [21] fuzzy logic based calculus of linguistically quantified propositions, a (proportional, nondecreasing) linguistic quantifier Q is assumed to be a fuzzy set defined, i.e. $\mu_Q : [0,1] \longrightarrow [0,1]$, $\mu_Q(x) \in [0,1]$. We consider *regular non-decreasing monotone* quantifiers, as e.g. "most" given by (8):

$$\mu(0) = 0; \qquad \mu(1) = 1; \qquad x_1 \leq x_2 \Rightarrow \mu_Q(x_1) \leq \mu_Q(x_2) \tag{7}$$

$$\mu_Q(x) = \begin{cases} 1 & \text{for } x \underline{>} 0.8 \\ 2x - 0.6 & \text{for } 0.3 < x < 0.8 \\ 0 & \text{for } x \underline{<} 0.3 \end{cases} \tag{8}$$

The truth values (from [0,1]) of (5) and (6) are calculated, respectively, as

$$\mathcal{T}(Qy\text{'s are } P) = \mu_Q\left(\frac{1}{n}\sum_{i=1}^{n}\mu_P(y_i)\right) \tag{9}$$

$$\mathcal{T}(QRy\text{'s are } P) = \mu_Q\left(\frac{\sum_{i=1}^{n}(\mu_R(y_i) \wedge \mu_P(y_i))}{\sum_{i=1}^{n}\mu_R(y_i)}\right) \tag{10}$$

where $\wedge$ is the minimum (more generally, e.g., a t-norm).

Both the fuzzy predicates P and R are assumed of a simplified, atomic form referring to one attribute, but can be extended to cover some confluences of various, multiple attribute values.

A *t-norm* is a $t : [0,1] \times [0,1] \longrightarrow [0,1]$, such that, for each $a, b, c \in [0,1]$:

1. it has 1 as the unit element, i.e. $t(a,1) = a$,
2. it is monotone, i.e. $a \leq b \Longrightarrow t(a,c) \leq t(b,c)$,
3. it is commutative, i.e. $t(a,b) = t(b,a)$,
4. it is associative, i.e. $t[a, t(b,c)] = t[t(a,b), c]$.

Some more relevant examples of t-norms are: (1) the minimum $t(a,b) = a \wedge b = \min(a,b)$ which is the most widely used, also here, (2) the algebraic product $t(a,b) = a \cdot b$, (3) the Łukasiewicz t-norm $t(a,b) = \max(0, a+b-1)$, and (4) the drastic t-norm $t(a,b) = \begin{cases} b \ a = 1 \\ a \ b = 1 \\ 0 \ otherwise \end{cases}$.

These operations can be in principle used in Zadeh's calculus but, clearly, their use may result in different results of the linguistic quantifier driven aggregation. Some examples will be shown in the next section.

7 Numerical Experiments

The method was tested on real data of daily quotations, from April 1998 to December 2006, of an investment fund that invests at most 50% of assets in shares, cf. Fig. 4, with the starting value of one share equal to PLN 10.00 and the final one equal to PLN 45.10 (PLN stands for the Polish Zloty); the minimum was PLN 6.88 while the maximum was PLN 45.15, and the biggest daily increase was PLN 0.91, while the biggest daily decrease was PLN 2.41.

For $\varepsilon = 0.25$ (PLN 0.25), we obtained 255 extracted trends, ranging from 2 to 71 time units (days). The histogram of duration is in Fig. 5.

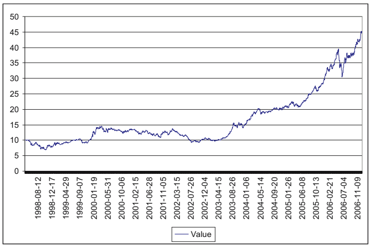


Fig. 4. A view of the original data

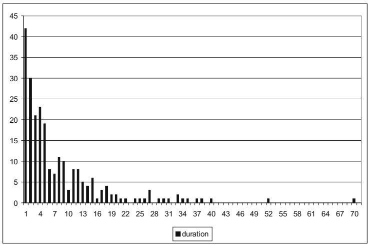


Fig. 5. Histogram of duration of trends

Figure 6 shows the histogram of angles (dynamics of change) and the histogram of variability of trends (in %) is in Fig. 7.

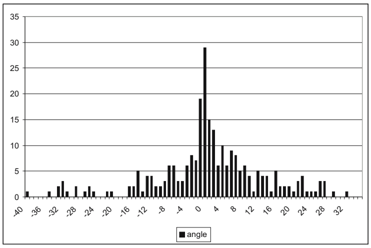


Fig. 6. Histogram of angles decscribing dynamic of change

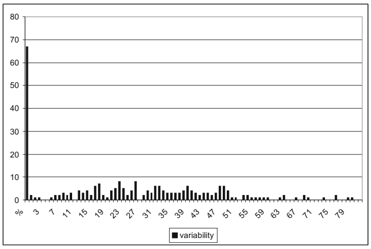


Fig. 7. Histogram of variability of trends

Some interesting summaries obtained, for different granulations of the dynamics of change, duration and variability, are:

- for 7 labels for the dynamics of change (*quickly increasing, increasing, slowly increasing, constant, slowly decreasing, decreasing and quickly decreasing*), 5 labels for the duration (*very long, long, medium, short, very short*) and 5 labels the variability (*very high, high, medium, low, very low*):

  - Most trends are very short, $\mathcal{T} = 0.78$
  - for different t-norms are shown in Table 1.
- 5 labels for the dynamics of change (*increasing, slowly increasing, constant, slowly decreasing, decreasing*), 3 labels for the duration (*short, medium, long*) and 5 labels for the variability (*very high, high, medium, low, very low*):
  - Most trends are of medium length, $\mathcal{T} = 0.431$
  - for different t-norms are shown in Table 2.

Table 1. Truth values for extended form summaries with different t-norms for the first granulation

Summary	minimum	product	Łukasiewicz	drastic
Most trends with a low variability are constant	0.974	0.944	0.911	0.85
Most slowly decreasing trends are of a very low variability	0.636	0.631	0.63	0.589
Almost all short trends are constant	1	1	1	1

Table 2. Truth values for extended form summaries with different t-norms for the second granulation

Summary	minimum	product	Łukasiewicz	drastic
Almost all decreasing trends are short	1	1	1	1
Almost all increasing trends are short	0.58	0.514	0.448	0.448
At least a half of medium length trends are constant	0.891	0.877	0.863	0.863
Most of slowly increasing trends are of a medium length	0.798	0.773	0.748	0.748
Most of trends with a low variability are constant	0.567	0.517	0.466	0.466
Most of trends with a very low variability are short	0.909	0.9	0.891	0.891
Most trends with a high variability are of a medium length	0.801	0.754	0.707	0.707
None of trends with a very high variability is long	1	1	1	1
None of decreasing trends is long	1	1	1	1
None of increasing trends is long	1	1	1	1

The particular linguistic summaries obtained, and their associated truth values, are intuitively appealing. In addition, these summaries were found interesting by domain experts though a detailed analysis from the point of view of financial analyses is beyond the scope of this paper. The results obtained for different t-norms are similar and, of course, the truth value for the case of the minimum is the highest.

8 Concluding Remarks

We proposed new types of lingustic summaries of time series. The derivation of a linguistic summary of a time series was related to a liguistic quantifier driven aggregation of trends, and we employed the classic Zadeh's calculus of linguistically quantified propositions with different t-norms, not only the classic minimum. We showed an application to the analysis of time series data on daily quotations of an investment fund over an eight year period, present some interesting lingustic sumaries obtained, and showed results for different t-norms. They suggest that varous t-norms exhibit slightly different behavior and they choice may be crucial for a particular application. The results are very promising.

References

1. Batyrshin, I.: On granular derivatives and the solution of a granular initial value problem. International Journal Applied Mathematics and Computer Science 12(3), 403–410 (2002)
2. Batyrshin, I., Sheremetov, L.: Perception based functions in qualitative forecasting. In: Batyrshin, I., Kacprzyk, J., Sheremetov, L., Zadeh, L.A. (eds.) Perception-based Data Mining and Decision Making in Economics and Finance, Springer, Heidelberg (2006)
3. Berndt, D.J., Clifford, J.: Finding patterns in time series: a dynamic programming approach. In: Advances in Knowledge Discovery and Data Mining, pp. 229–248. AAAI/MIT Press, Menlo Park, CA (1996)
4. Chiang, D.-A., Chow, L.R., Wang, Y.-F.: Mining time series data by a fuzzy linguistic summary system. Fuzzy Sets and Systems 112, 419–432 (2000)
5. Das, G., Lin, K., Mannila, H., Renganathan, G., Smyth, P.: Rule discovery from time series. In: Proc. of the 4th Int'l Conference on Knowledge Discovery and Data Mining. New York, NY, pp. 16–22 (1998)
6. Kacprzyk, J., Wilbik, A., Zadrożny, S.: Linguistic summarization of trends: a fuzzy logic based approach. In: Proceedings of the 11th International Conference Information Processing and Management of Uncertainty in Knowledge-based Systems, Paris, France, July 2-7, 2006, pp. 2166–2172 (2006)
7. Kacprzyk, J., Wilbik, A., Zadrożny, S.: Linguistic summaries of time series via a quantifier based aggregation using the Sugeno integral. In: Kacprzyk, J., Wilbik, A. (eds.) Proceedings of 2006 IEEE World Congress on Computational Intelligence, Vancouver, BC, Canada, July 16-21, 2006, pp. 3610–3616. IEEE Press, New York (2006)
8. Kacprzyk, J., Wilbik, A., Zadrożny, S.: On some types of linguistic summaries of time series. In: Proceedings of the 3rd International IEEE Conference Intelligent Systems, London, UK, pp. 373–378. IEEE Press, New York (2006)
9. Kacprzyk, J., Wilbik, A., Zadrożny, S.: A linguistic quantifier based aggregation for a human consistent summarization of time series. In: Lawry, J., Miranda, E., Bugarin, A., Li, S., Gil, M.A., Grzegorzewski, P., Hryniewicz, O. (eds.) Soft Methods for Integrated Uncertainty Modelling, pp. 186–190. Springer, Heidelberg (2006)
10. Kacprzyk, J., Wilbik, A., Zadrożny, S.: Capturing the essence of a dynamic behavior of sequences of numerical data using elements of a quasi-natural language. In: Proceedings of the 2006 IEEE International Conference on Systems, Man, and Cybernetics, Taipei, Taiwan, 2006, pp. 3365–3370. IEEE Press, New York (2006)
11. Kacprzyk, J., Yager, R.R.: Linguistic summaries of data using fuzzy logic. International Journal of General Systems 30, 33–154 (2001)
12. Kacprzyk, J., Yager, R.R., Zadrożny, S.: A fuzzy logic based approach to linguistic summaries of databases. International Journal of Applied Mathematics and Computer Science 10, 813–834 (2000)
13. Kacprzyk, J., Zadrożny, S.: Linguistic database summaries and their protoforms: toward natural language based knowledge discovery tools. Information Sciences 173, 281–304 (2005)
14. Kacprzyk, J., Zadrożny, S.: Fuzzy linguistic data summaries as a human consistent, user adaptable solution to data mining. In: Gabrys, B., Leiviska, K., Strackeljan, J. (eds.) Do Smart Adaptive Systems Exist? pp. 321–339. Springer, Heidelberg (2005)

15. Keogh, E., Pazzani, M.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In: Proc. of the 4th Int'l Conference on Knowledge Discovery and Data Mining. New York, NY, pp. 239–241 (1998)
16. Sklansky, J., Gonzalez, V.: Fast polygonal approximation of digitized curves. Pattern Recognition 12(5), 327–331 (1980)
17. Sripada, S., Reiter, E., Davy, I.: SumTime-Mousam: Configurable Marine Weather Forecast Generator. Expert Update 6(3), 4–10 (2003)
18. Yager, R.R.: A new approach to the summarization of data. Information Sciences 28, 69–86 (1982)
19. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Transactions on Systems, Man. and Cybernetics SMC 18, 183–190 (1988)
20. Yager, R.R., Kacprzyk, J.: The Ordered Weighted Averaging Operators: Theory and Applications. Kluwer, Boston (1997)
21. Zadeh, L.A.: A computational approach to fuzzy quantifiers in natural languages. Computers and Mathematics with Applications 9, 149–184 (1983)
22. Zadeh, L.A.: A prototype-centered approach to adding deduction capabilities to search engines – the concept of a protoform. In: Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS, 2002) pp. 523–525 (2002)
23. Zadeh, L.A., Kacprzyk, J. (eds.): Computing with Words in Information/Intelligent Systems: 1. Foundations, 2. Applications. Physica-Verlag, Heidelberg (1999a)

Type-2 Fuzzy Summarization of Data: An Improved News Generating

Adam Niewiadomski

Institute of Computer Science, Technical University of Lodz
ul. Wólczańska 215, 90-924 Łódź, Poland
aniewiadomski@ics.p.lodz.pl

Abstract. The paper introduces an improved method of intelligent summarization of large datasets. Previously, the author's solution for automated generating of textual news and comments, based on the standard Yager's method and ordinary fuzzy sets, has been published in [1]. In this paper, a type-2-fuzzy-set-based extension of the concept can be now introduced. Type-2 membership functions are originally applied to build new summarization methods. The approach generalizes the previous methods which are based on traditional fuzzy sets. Moreover, new quality measures of summaries are proposed and used in selecting the optimal and the most specific summaries as the components of textual news. Finally, the method is implemented and evaluated.

1 Motivation and Problem Study

The problem of distilling useful and ready-to-use knowledge from huge amounts of unstructured and dispersed data, is very present now. The original concept of *a linguistic summary of a database* introduced by R. R. Yager in 1982 [2] appeared a simple and effective methods. Linguistic summaries are natural language sentences that approximately but clearly describe properties of objects, e.g. *About 100 of my students are excellent programmers*, where *students* is the subject of summary, and *about 100* and *excellent programmers* are pronouncements of amount and property, respectively, both handled by fuzzy logic [3,4].

The gist of the paper is to enhance the Yager method with the use of *Type-2 Fuzzy Sets* [5]. They extend the Zadeh idea, and enable representing imprecise information via type-2 membership functions which are fuzzy-valued functions. Since traditional membership functions may appear inconsistent as they represent imperfect information via precise and crisp numbers, the use of type-2 membership functions as models of vague quantities and features needs to be discussed. Some research on type-2 fuzzy sets in linguistic data summarization have already been made by the author in [6,7,8,9].

The main motivation to generalize the Yager approach is that membership degrees of properties or phenomena under many circumstances may be inexpressible in terms of crisp values. Type-1 membership functions are frequently constructed based on preferences of one expert. However, it may look arbitrary, since it seems more natural when two or more opinions are given to illustrate

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 241–250, 2007.

e.g. a linguistic term, to model it as objectively as possible. Traditional fuzzy sets dispose no methods of handling these, usually different, opinions. The average or median of several membership degrees keep no information about those natural differences. For instance, the question *What is the compatibility level of the* $36.5°C$ *with "temperature of a healthy human body"?* can be answered 0.5, 1.0, 1.0 by three doctors, respectively, but the average, 0.866, does not show that one of them remains unconvinced.

Extending real (type-1) membership levels to fuzzy-valued (type-2) provides additional computational tools – secondary membership degrees. They may be interpreted as *possibility levels* that primary degrees describe memberships appropriately, but from the point of view of the linguistic summarization, interpreting them as *weights* [5] is practicable. Thanks to it, different expert opinions on a membership degree may be described by "confidence levels" which express e.g. expert's experience. See the example on the temperature (above): the proposed compatibility values may be presented as $(0.5, 0.2), (1.0, 1.0), (1.0, 0.9)$ which says that the first expert is much less experienced than the others and this information is stored in the resulting fuzzy set. This set may be – but *need not* to be – defuzzified or averaged. The goal is to use different types of fuzzy sets when generating summaries, and to maintain the understandable semantics of results (real degrees of truth and other quality measures) proposed by Yager. Thus, we present a general method of summarization in which many types of fuzzy sets may be applied, and the differences among them are hidden for an end-user.

2 Information Representation Via Type-2 Fuzzy Sets

2.1 Basic Definitions

The idea of a type-2 fuzzy set extends an ordinary membership function to a *type-2 membership function*. This is a family of type-1 sets in $[0, 1]$ assigned to elements of a universe of discourse. A *type-2 fuzzy set* $\tilde{A}$ in $\mathcal{X}$ is defined $\tilde{A} = \int_{\mathcal{X}} \mu_{\tilde{A}}(x)/x$ and $\mu_{\tilde{A}}: \mathcal{X} \to \mathcal{F}([0, 1])$ is the type-2 membership function, such that $\mu_{\tilde{A}}(x) = \int_{u \in J_x} \mu_x(u)/u$, $J_x \subseteq [0, 1]$. Each u has its own membership degree assigned. Moreover, many u's can be assigned to a given x, and each has its separated *secondary* membership degree $\mu_x(u)$. For a fixed x', $\mu_{x'}$ is the membership function for the type-1 set which expresses the membership of x' to $\tilde{A}$, i.e. for the $\mu_{\tilde{A}}(x')$ value. Secondary degrees may be viewed as *weights* or as *possibility levels*, cf. [5].

The set-theoretical operations on type-2 sets are extensions of the analogous ones in other fuzzy set theories. Let $\tilde{A}$, $\tilde{B}$ be type-2 sets in $\mathcal{X}$. Let t_1, t_2 be t-norms. The intersection of $\tilde{A}$ and $\tilde{B}$ is the type-2 set $\tilde{A} \cap \tilde{B}$, the membership function of which is defined in terms of *the meet* operation:

$$\mu_{\tilde{A} \cap \tilde{B}}(x) = \mu_{\tilde{A}}(x) \sqcap \mu_{\tilde{B}}(x) = \int_{u_{\tilde{A}}} \int_{u_{\tilde{B}}} (\mu_x(u_{\tilde{A}}) \; t_1 \; \mu_x(u_{\tilde{B}}))/(u_{\tilde{A}} \; t_2 \; u_{\tilde{B}}) \qquad (1)$$

where $u_{\tilde{A}}$, $u_{\tilde{B}}$ – primary membership degrees of x in $\tilde{A}$, $\tilde{B}$, respectively; $\mu_x(u_{\tilde{A}})$, $\mu_x(u_{\tilde{B}})$ – secondary degrees of x in $\tilde{A}$, $\tilde{B}$, respectively. Eq. (1) is applied as a model for the AND connective that combines single summarizers, see Sec. 3.

The concept of *embedded fuzzy set* appears useful in defining other concepts. *an embedded type-1 set* A_λ for a type-2 fuzzy set $\tilde{A}$ in $\mathcal{X}$, is defined. Let $\forall_{x\in\mathcal{X}}\ \lambda_x \in J_x \subseteq [0,1]$. The membership function for A_λ is given as $\mu_{A_\lambda}(x) = \lambda_x$,

2.2 Cardinality, Support, and Degree of Imprecision of Type-2 Sets

Cardinality of a crisp set A' in $\mathcal{X}$ is the sum of the $\xi_{A'}$ characteristic function values $card(A') = \sum_{x\in\mathcal{X}} \xi_{A'}(x)$. The cardinality of a type-1 set A in $\mathcal{X}$ [10]

$$card(A) = \sum\nolimits_{x\in\mathcal{X}} \mu_A(x) \tag{2}$$

The cardinality of a type-2 set, *non-fuzzy sigma count*, assumes that membership of x in $\tilde{A}$ in $\mathcal{X}$ is a fuzzy number. Hence nfσ-count($\tilde{A}$) is defined:

$$\text{nf}\sigma\text{-count}(\tilde{A}) = \sum\nolimits_{x\in\mathcal{X}} \max\{u \in J_x \colon \mu_x(u) = 1\} \tag{3}$$

The given definition is a generalization of the analogous definition for an ordinary fuzzy set, given by de Luca and Termini [10].

The support of a type-1 set is defined as

$$supp(A) =_{df} \{x \in \mathcal{X} \colon \mu_A(x) > 0\} \tag{4}$$

and is applied to measure the goodness of summaries. We propose to extend it to *the fuzzy support* – a set of type-1 associated with a given type-2 set.

Definition 1. *Let $\widetilde{A}$ be a type-2 set in $\mathcal{X}$.* The fuzzy support of $\widetilde{A}$ *is the type-1 set* $supp(\widetilde{A}) =_{df} \{\langle x, \mu_{supp(\widetilde{A})}\rangle \colon x \in \mathcal{X}\}$ *where*

$$\mu_{supp(\widetilde{A})}(x) = \sup_{u\in J_x\setminus\{0\}} \mu_x(u) \tag{5}$$

Proposition 1. *For each type-1 fuzzy set A, $\mu_{supp(A)}(x) = \xi_{A_0}(x)$*

Proof. Let A be a type-1 fuzzy set in $\mathcal{X}$. Hence, each its element has only one primary membership value assigned, $u(x)$, and $\forall_{x\in\mathcal{X}}\ \mu_x(u) = 1$, so the supremum in (5) can be omitted. Thus, $supp(A)$ – the zero-cut of A, is a crisp set.

Definition 2. *Let $\widetilde{A}$ be a type-2 set in $\mathcal{X}$. The degree of fuzziness of $\widetilde{A}$ is defined:*

$$in(\widetilde{A}) =_{df} card(supp(\widetilde{A}))/card(\mathcal{X}) \tag{6}$$

The definition extends the concept for type-1 sets, and is applied to determine quality indices of type-2 summaries in Sec. 3.3.

2.3 Type-1 Fuzzy Quantification of Type-2 Fuzzy Propositions

The canonical forms of linguistically quantified propositions are defined in [4]. We originally generalize them with type-2 sets as models of S_1, S_2

Definition 3. *Let $\widetilde{S}_1$, $\widetilde{S}_2$ be type-2 sets representing linguistic propositions, and Q – a type-1 fuzzy quantifier. The formulae*

$$Q \textit{ x's are } \widetilde{S_1} \tag{7}$$

$$Q \textit{ x's being } \widetilde{S_2} \textit{ are } \widetilde{S_1} \tag{8}$$

are the first *(Q^I) and* the second canonical form (Q^{II}) *of the linguistically quantified proposition. Degrees of truth of (7) and (8) are assessed as*

$$T(\ Q \textit{ x's are } \widetilde{S_1}\ ) = \mu_Q(card(\widetilde{S_1})/M) \tag{9}$$

where $card(\widetilde{S_1})$ is a real number, see (3), $M = card(\mathcal{X})$ if Q is relative, or $M = 1$ if Q is absolute, and

$$T(\ Q \textit{ x's being } \widetilde{S_2} \textit{ are } \widetilde{S_1}) = \mu_Q(card(\widetilde{S_1} \cap \widetilde{S_2})/card(\widetilde{S_2})) \tag{10}$$

where $\widetilde{S_1} \cap \widetilde{S_2}$ is given in (1).

Examples for Q^I, Q^{II}, are Many *students are intelligent* and Many *of young students are intelligent*, respectively, in which Many=Q, *intelligent*=S_1, and *young*=S_2. Similarly to the propositions represented by type-1 sets, only relative quantification is possible in (8).

3 Type-2 Linguistic Summaries of Data

The section introduces the linguistic data summarization algorithms innovated by the use of type-2 fuzzy logic. In particular, we are interested in the Q P are/have $\widetilde{S}$ $[T]$, form of summary, in which $\widetilde{S}$ is a summarizer represented by a type-2 fuzzy set, and Q, P, T are interpreted as in type-1 summaries.

3.1 Type-2 Summaries in the First Canonical Form

We introduce the type-2 summaries based on Q^I, see (7). The goal is to find a quality index for a given summary in the form of Q P are/have $\widetilde{S}$. We assume here, that Q is represented by a type-1 fuzzy set and the cardinality of $\widetilde{S}$ is computed via (3). The degree of truth of such a summary is a real number

$$T\left(Q\ P\ \text{ are/have } \widetilde{S}\right) = \mu_Q(\text{nf}\sigma\text{-count}(\widetilde{S})/M) \tag{11}$$

where $M = 1$ if Q is absolute, or $M = m = card(\mathcal{D})$ if Q is relative. Assume that n fuzzy sets $\widetilde{S}_1$,. . .,$\widetilde{S}_n$ are chosen and at least one of them is of type-2. They represent linguistically expressed properties of objects $y_1, ..., y_m$ described by records d_1,. . .,d_m. The membership function of the type-2 composite summarizer $\widetilde{S} = \widetilde{S}_1$ AND $\widetilde{S}_2$ AND . . . AND $\widetilde{S}_n$ is computed as

$$\mu_{\widetilde{S}}(d_i) = \mu_{\widetilde{S}_1 \cap \widetilde{S}_2 \cap \ldots \cap \widetilde{S}_n}(d_i) \tag{12}$$

where the intersection is given by (1). Notice that (12) describes the extension of the George and Srikanth approach [11], and, in consequence, for $n = 1$, also the Yager method of summarization.

3.2 Summaries Based on the Second Canonical Form

Linguistic summaries based on Q^{II}, see (8), are in the form of

$$Q\ P \text{ being } \widetilde{w}_g \text{ are/have } \widetilde{S}\ [T] \tag{13}$$

in which $\widetilde{w}_g$ is represented by a type-2 fuzzy set, and $\widetilde{S}$ is a type-2 or type-1, composite or single summarizer. Similarly to the method presented in [12], the use of the additional fuzzy set enables producing much more interesting summaries. Hence, according to (10), the $\mu_{\widetilde{S}}(d_i)$ is intersected with the membership to the $\widetilde{w}_g$ query:

$$\mu_{\widetilde{w}_g \cap \widetilde{S}}(d_i) = \mu_{\widetilde{w}_g}(d_i) \cap \underbrace{\mu_{\widetilde{S}_1}(d_i) \cap \ldots \cap \mu_{\widetilde{S}_n}(d_i)}_{\mu_{\widetilde{S}}} \tag{14}$$

Step 1. For each $i = 1, ..., m$ compute $\mu_{\widetilde{w}_g}(d_i) \in \mathcal{F}([0, 1])$
Step 2. Construct the base $\mathcal{D} \supseteq \mathcal{D}' = \{d_i \colon \mu_{\widetilde{w}_g}(d_i) \neq \emptyset\}$, $m' = card(\mathcal{D}') \leq m$
Hence, the degree of truth of the (13) summary is a real number

$$T = \mu_Q(\text{nf}\sigma\text{-count}(\widetilde{w}_g \cap \widetilde{S})/\text{nf}\sigma\text{-count}(\widetilde{w}_g)) \tag{15}$$

Thanks to Steps 1, 2, the computational cost is reduced from $m \cdot (n + 1)$ to at most $m' \cdot n + m$ membership assessments.

3.3 Quality Measures for Type-2 Summaries

This section introduces the original extensions of five measures for type-1 summaries [12]. The next five indices, $T_6 - T_{10}$, are new and specific for type-2 summaries (although their versions for type-1 summaries may also be considered).

1. **Degree of Truth** – see (11), (15).
2. **Degree of Imprecision.** The degree of imprecision of a linguistic summary with a type-2 fuzzy summarizer is determined as

$$T_2 = 1 - \left(\prod_{j=1}^{n} in(\widetilde{S}_j)\right)^{1/n} \tag{16}$$

The closer to 1 is T_2, the more precise the summary.

3. **Degree of Covering.** The degree of covering is possible to be computed if a summary is based on the second canonical form, see ([8]).

$$T_3 = card(supp(\widetilde{w}_g \cap \widetilde{S}))/card(supp(\widetilde{w}_g)) \tag{17}$$

The meaning of the index is the (relative) number of objects corresponding to the query and covered by the summary.

4. **Degree of Appropriateness** – we decompose a summarizer into a number of fuzzy sets $\widetilde{S}_1, \ldots, \widetilde{S}_n$, and for each the r_j index is computed via ([11]). The degree of appropriateness is based on $g_{i,j}$

$$g_{i,j} = \mu_{supp(\widetilde{S}_j)}(d_i) \tag{18}$$

which is depends on the support of the $\widetilde{S}_j$ type-2 fuzzy set representing the j-th summarizer. Hence

$$T_4 = \left| \prod_{j=1}^{n} \frac{\sum_{i=1}^{m} g_{i,j}}{m} - T_3 \right| \tag{19}$$

5. **Length of a Type-2 Summary** – depends on $b = card(\{\widetilde{S}_1, \ldots, \widetilde{S}_b\})$ – the number of sets that represent a summarizer, $b \leq n$. The more sets, the less precise the summarizer:

$$T_5 = 2 \cdot (0.5)^b \tag{20}$$

6. **Type-2 Quantification Imprecision** – is analogous to T_2

$$T_6 = 1 - in(Q) \tag{21}$$

7. **Type-2 Quantification Cardinality**

$$T_7 = 1 - card(Q)/N \tag{22}$$

where $N = 1$ if Q is relative, or $N = card(D(Q))$ if Q is absolute.

8. **Type-2 Summarizer Cardinality**– because of possible several fuzzy sets $\widetilde{S}_1$, ..., $\widetilde{S}_n$ representing the summarizer, the form of T_8 is:

$$T_8 = 1 - \left(\prod_{j=1}^{n} \text{nf}\sigma\text{-count}(\widetilde{S}_j)/card(\mathcal{X}_j) \right)^{1/n} \tag{23}$$

9. **Imprecision of The Type-2 Query** T_9 is determined by the degree of imprecision of the query in a summary based on the second canonical form:

$$T_9 = 1 - in(\widetilde{w}_g) \tag{24}$$

10. **Cardinality of The Type-2 Query**

$$T_{10} = \text{nf}\sigma\text{-count}(\widetilde{w}_g)/card(\mathcal{D}(\widetilde{w}_g)) \tag{25}$$

4 An Improved News Generating

4.1 The Algorithm

The general assumptions of the system which produces compact textual messages from large sets of numerical data, are given in [1]. Below, we present the improved version of the algorithm that generates news with the use of type-2 fuzzy summarizers $\widetilde{S_1}, ..., \widetilde{S_n}$. The measures described in Section 3.3 are applied to select the summaries of the highest goodness (i.e. the most informative).

```
// generating summaries in the form of $Q^I$
1. for each non-empty $\hat{S} \subseteq \{\widetilde{S_1}, ..., \widetilde{S_z}\}$
1.1. determine $\mu_{\hat{S}}(d_i)$ via (12)
1.2. for each quantifier $Q_h$, $h = 1, ..., k$
     compute $T_{1,h}$, $T_{6,h}$, and $T_{7,h}$ via (11), (21), and (22), respectively
1.3. compute $T_{h_{\max}} = \max_{h \in \{1,...,k\}} \{t\colon t = w_1 T_{1,h} + w_6 T_{6,h} + w_7 T_{7,h}\}$, remember $h_{\max}$
1.4. compute $T_2$, see (16)
// $T_3$, $T_9$, $T_{10}$ are not assessed, because of no $\widetilde{w_g}$ queries in $Q^I$
1.5. compute $T_4$, via (18), (19), for $T_3 = 0$
1.6. compute $T_5$ via (20)
1.7. compute $T_8$ via (23)
1.8. $T = T_{h_{\max}} + w_2 \cdot T_2 + w_4 \cdot T_4 + w_5 \cdot T_5 + w_8 \cdot T_8$
1.9. generate the summary $Q_{h_{\max}}$ $P$ are/have $\hat{S}$ [$T$]

// generating summaries in the form of $Q^{II}$
2. for each non-empty query $\widetilde{S_w} \subsetneq \{\widetilde{S_1}, ..., \widetilde{S_z}\}$
   and for each non-empty summarizer $\hat{S} \subseteq \{\widetilde{S_1}, ..., \widetilde{S_z}\} \setminus \widetilde{S_w}$
   2.1. determine $\mu_{\widetilde{S_w}}(d_i)$ via (14)
   2.2. determine $D \supseteq D_w = \{d_i \in D\colon \mu_{\widetilde{S_w}}(d_i) \neq \emptyset\}$
   2.3. for each $d_i \in D_w$ determine $\mu_{\hat{S}}(d_i)$
   2.4. for each relative quantifier $Q_h\colon h \in \{1, ..., k\}$
        compute $T_{1,h}$, $T_{6,h}$, and $T_{7,h}$ via (15), (21), and (22), respectively
   2.5. compute $T_{h_{\max}}$ analogously to 1.3., remember $h_{\max}$
   2.6. compute $T_2$ analogously to 1.4.
   2.7. compute $T_3$ according to (17)
   2.8. compute $T_4 = \left| \prod_{\widetilde{S_j} \in \hat{S}} \frac{\sum_{d_i \in D_w} g_{i,j}}{card(D_w)} - T_3 \right|$, via (18)-(19)
   2.9. compute $T_5$ analogously to 1.6.
   2.10. compute $T_8$ analogously to 1.7.
   2.11. compute $T_9$ and $T_{10}$ via (24), (25), resp.
   2.12. $T = T_{h_{\max}} + \sum_{i=2}^{5} w_i \cdot T_i + \sum_{i=8}^{10} w_i \cdot T_i$
   2.13. generate summary $Q_{h_{\max}}$ $P$ being $S_w$ are/have $\hat{S}$ [$T$]
```

Ad. 1. In this step, finding all non-empty subsets of $\{\widetilde{S_1}, ..., \widetilde{S_z}\}$ is required; the number of such subsets is exactly $2^z - 1$. In the implementation, the problem is resolved via generating binary forms of all natural numbers between 0 and $2^z - 1$. The forms are taken as characteristic vectors of the sought subsets.

Ad. 1.3. and 1.8. $w_1 + w_2 + w_4 + w_5 + w_6 + w_7 + w_8 = 1$.

Ad. 2.10. $w_1 + \ldots + w_{10} = 1$.

4.2 Implementation and Results

The algorithm has been implemented on .NET platform in the C# language. The database (in MS SQL Server (*.mdf) and MS Access (*.mdb) formats, has consisted of ca 10,000 records on workers of a company. The view containing tuples in the form of $\langle$`Age, Education, Salary`$\rangle$ has been generated. The summarizers have been determined as values of linguistic variables $L_1 = Age$, $L_2 = Education$, $L_3 = Salary$, e.g. $H(Age) = \{$young, middle-aged, experienced, about 40, about 30$\}$. Each label of L_1, L_3 have been represented by a type-2 fuzzy set, and of L_2 – by crisp sets. Sample results for S_1=about 30, S_2=high school, and S_3=about 4000 is presented:

```
About half of workers are ab. 30 [0.47]. Much more than 2000 workers
graduated from high school [0.74]. About half of workers earn ab.
4000 [0.54]. Many workers graduated from high schools and earn ab.
4000 [0.37]. Many workers graduated from high schools and are ab. 30
[0.38]. Many workers earn ab. 4000 and are ab. 30 [0.37]. Ab. half of
workers graduated from high schools are ab. 30 [0.46].
```

Finally, we notice the results obtained are at least of the same quality that similar given by type-1 summarization methods, see Sec. 5.

5 Evaluating the Success of the Type-2 Summarization

The introduced type-2 linguistic summarization is a generalization of the existing methods based on type-1 fuzzy sets, i.e. summarizers, quantifiers, and queries, are now represented by type-2 membership functions, the values of which are fuzzy numbers. Since a real number is a specific case of fuzzy, type-1 methods can be applied together, because the new approach includes them as specific cases.

However, type-2 membership functions are more complicated than type-1. They are families of at least several type-1 functions that represent given data, e.g. preferences of experts. Unfortunately, they are more time-consuming because more membership values, primary and secondary, must be assessed, see e.g. the definitions of cardinalities for type-1, cf. [10] and type-2 sets, cf. ([3]).

Hence, although type-2 summaries are more time consuming, we expect that they allow to produce the results that cover also type-1 summaries, in particular, summaries at least as informative as the obtained through type-1 methods, according to the measures of informativeness presented in Sec. 3.3.

Assumptions for comparing type-1 and type-2 summaries. We compare type-1 and type-2 summaries under the following assumptions:

(A1) *The same set of records described by attributes* V_1, ..., V_n *is summarized both under type-1 and type-2 methods.*

(A2) *The* $\mathcal{X}_1$, ..., $\mathcal{X}_n$ *sets are the domains of* V_1, ..., V_n*, respectively, and* $\forall_{i=1,\ldots,n}$, $\mathcal{X}_i \subseteq R$

(A3) *If a type-1 set A and a type-2 $\widetilde{A}$ in $\mathcal{X}_i$, $i \leq n$, represent the same linguistic term, then A is considered as an embedded type-1 fuzzy set in $\widetilde{A}$*[1].

Comparing type-1 and type-2 quality measures. The quality indices for summaries are based on cardinality and support, (2), (4) for type-1, and (3), (5) for type-2. From these equations, and from the concept of embedded type-1 set:

Proposition 2. *For each type-1 A embedded in type-2 $\widetilde{A}$ in $\mathcal{X}$*

$$card(A) \leq card(\widetilde{A}) \tag{26}$$

Proof. Let $x \in \mathcal{X}$, $u_A = \mu_A(x)$. A is of type-1, hence, $\mu_x(u_A) = 1$. Furthermore, $u_A \in \{u_{\widetilde{A}} \colon \mu_x(u_{\widetilde{A}}) = 1\}$. Thus, $u_A \leq \max\{u_{\widetilde{A}} \colon \mu_x(u_{\widetilde{A}}) = 1\}$, and from (2), (3), we have $\sum_{x\in\mathcal{X}} u_A \leq \sum_{x\in\mathcal{X}} \max\{u_{\widetilde{A}} \in J_x \colon \mu_x(u_{\widetilde{A}}) = 1\}$.

Proposition 3. *For each type-1 A embedded in type-2 $\widetilde{A}$ in $\mathcal{X}$*

$$supp(A) = supp(\widetilde{A}) \quad \wedge \quad card(supp(A)) = card(supp(\widetilde{A})) \tag{27}$$

Proof. Let $x \in \mathcal{X}$, $u_A = \mu_A(x), u_A > 0$. Hence $\xi_{supp(A)}(x) = 1$. Since A is of type-1, $\mu_x(u_A) = 1$. Hence, from (5), we have $\mu_{supp(\widetilde{A})} = \sup_{u\in J_x\setminus\{0\}} \mu_x(u) = 1$. Thus, $\forall_{x\in\mathcal{X}}\ \xi_{supp(A)}(x) = \mu_{supp(\widetilde{A})}(x)$.

Proposition 4. *Let (A1)–(A3) are fulfilled. Let type-1 $S_1, ..., S_n, w_g$ in $\mathcal{X}_1, ...,$ $\mathcal{X}_{n+1}$ be embedded in type-2 $\widetilde{S}_1, ..., \widetilde{S}_n, \widetilde{w}_g$ in $\mathcal{X}_1, ..., \mathcal{X}_{n+1}$. Let Q ba a fuzzy quantifier. Let us denote by $T_i(Q, S_1, ..., S_n, w_g)$, and $T_i(Q, \widetilde{S}_1, ..., \widetilde{S}_n, \widetilde{w}_g)$, $i = 1 \ldots 10$, the measures described in Sec 3.3, for Q, S_1,...,S_n, w_g and for Q, $\widetilde{S}_1$,...,$\widetilde{S}_n$, $\widetilde{w}_g$.*

$$T_1(Q, S_1, ..., S_n, w_g) \leq T_1(Q, \widetilde{S}_1, ..., \widetilde{S}_n, \widetilde{w}_g) \ \textit{ from } (26),\ (11),\ (15) \tag{28}$$

$$T_7(Q, S_1, ..., S_n, w_g) \leq T_7(Q, \widetilde{S}_1, ..., \widetilde{S}_n, \widetilde{w}_g) \ \textit{ from } (26),\ (22) \tag{29}$$

$$T_8(Q, S_1, ..., S_n, w_g) \leq T_8(Q, \widetilde{S}_1, ..., \widetilde{S}_n, \widetilde{w}_g) \ \textit{ from } (26),\ (23) \tag{30}$$

$$T_{10}(Q, S_1, ..., S_n, w_g) \leq T_{10}(Q, \widetilde{S}_1, ..., \widetilde{S}_n, \widetilde{w}_g) \ \textit{ from } (26),\ (25) \tag{31}$$

Besides, for $i = 2 \div 6, 9$, $T_i(Q, S_1, ..., S_n, w_g) = T_i(Q, \widetilde{S}_1, ..., \widetilde{S}_n, \widetilde{w}_g)$, see (27).

We conclude from Prop. 4 that the measures based on cardinalities, T_1, T_7, T_8, T_{10} take values greater or equal for type-2 than for type-1 summaries, while measures based on supports, T_2, T_3, T_4, T_6, T_9, take the same values for type-1 and type-2 summaries. Thus, the proposed type-2 summarization allows to achieve the results which are at least as informative as type-1 methods.

[1] It represents a proposed type-1 membership function "bridged" with other expert proposals, and, finally, a term is described by a type-2 membership function.

6 Conclusions

The contribution to the domain of data intelligent summarization, presented in this paper, can be, *nomen omen*, summarized in the following points:

- The original method of linguistic data summarization handled by type-2 fuzzy logic, has been presented.
- The method is an extension of the existing methods based on type-1 fuzzy logic; it covers the previous as a specific case.
- The known quality measures for type-1 summaries have been enhanced to their type-2 versions, and new quality measures of type-2 summaries have been proposed, also applying to type-1 summaries.
- The improved algorithm for finding optimal and the most specific type-2 summaries, has been presented. It is applied to the task and schema presented in [1], and generalizes it.
- The new method produces summaries that are based on more experts preferences. Hence, the results are more informative.

References

1. Niewiadomski, A.: News generating via fuzzy summarization of databases. LNCS, vol. 3831, pp. 419–429. Springer, Heidelberg (2006)
2. Yager, R.: A new approach to the summarization of data. Inf. Sci. 28, 69–86 (1982)
3. Zadeh, L.A.: The concept of linguistic variable and its application for approximate reasoning (i). Information Sciences 8, 199–249 (1975)
4. Zadeh, L.A.: A computational approach to fuzzy quantifiers in natural languages. Computers and Maths with Applications 9, 149–184 (1983)
5. Mendel, J.M. (ed.): Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, Upper Saddle River, NJ (2001)
6. Niewiadomski, A.: On two possible roles of type-2 fuzzy sets in linguistic summaries. Lecture Notes in Artificial Intelligence 3528, 341–347 (2005)
7. Niewiadomski, A., Bartyzel, M.: Elements of type-2 semantics in linguistic summaries of databases. Lecture Notes in Artificial Intelligence 4029, 278–287 (2006)
8. Niewiadomski, A., Ochelska, J., Szczepaniak, P.S.: Interval-valued linguistic summaries of databases. Control and Cybernetics (2), 415–444 (2005)
9. Niewiadomski, A., Szczepaniak, P.S.: News generating based on interval type-2 linguistic summaries of databases. In: Proceedings of IPMU 2006 Conference, July 2–7, 2006, Paris, France, pp. 1324–1331 (2006)
10. De Luca, A., Termini, S.: A definition of the non-probabilistic entropy in the setting of fuzzy sets theory. Information and Control 20, 301–312 (1972)
11. George, R., Srikanth, R.: Data summarization using genetic algorithms and fuzzy logic. In: Herrera, F., Verdegay, J. (eds.) Genetic Algorithms and Soft Computing, pp. 599–611. Physica–Verlag, Heidelberg (1996)
12. Kacprzyk, J., Yager, R.R.: Linguistic summaries of data using fuzzy logic. Int. J. of General Systems 30, 133–154 (2001)

A Note on Granular Sets and Their Relation to Rough Sets

Antoni Ligęza and Marcin Szpyrka

Institute of Automatics
AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
ali@agh.edu.pl, mszpyrka@agh.edu.pl

Abstract. The paper discusses mathematical concept of granular model for data and knowledge manipulation. In order to overcome the difficulties caused by extensive data representation a new model based on *granular sets* and *granular relations* is put forward. The key idea is that the notion of set may consist of basic elements grouped into bigger granules. A granular set is formed from a universe set and a semi-partition defining granules of its elements. Formal definition of granular sets and some basic algebraic operations on granular sets are introduced in the paper. Further, the concept of granular relation is also defined and some possibilities of application of granular sets and relations to knowledge representation are put forward.

1 Introduction

Representation of data and knowledge with *adaptable granularity* of details seems to be an interesting issue for efficient dealing with large data sets. The paper presents a relatively new concept of a *granular set* and *granular relation* [5], [6]. A granular set is a structure composed of a set and a number of disjoint subsets embedded in it (the so-called semi-partition). An algebra of such sets can be constructed. Granular relation can be defined as a subset of Cartesian product of granular sets. A *granular relational algebra* can be defined as a tool for knowledge manipulation. It can be applied for verification and analysis of tabular knowledge-based systems [7] and for direct knowledge manipulation.

One reason for using granular representations can be the need for efficient dealing with large data sets. In such a case numerous detailed data are grouped into a single *granule* which can be regarded as more abstract knowledge representation. The number of detailed data items is drastically reduced and simultaneously some unimportant characteristics are hidden. In this way *knowledge extraction* from data can go on.

Another reason for using granular knowledge representation consists in the need for structuring knowledge into smaller, separate, easily manipulable chunks of knowledge. Such "knowledge granules" can be easier interpreted and understood, selected and manipulated, analyzed and verified. Granularity of knowledge seems to be an intrinsic issue in the domain of *knowledge management*.

There are a number of conceptual approaches aimed at dealing with impreciseness and knowledge abstraction. Some most important ones include *Fuzzy Sets* [13], [3],

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 251–260, 2007.

Rough Sets [9], *interval algebra*, as well as selected basic purely mathematical approaches such as ones based on *equivalence relation* and *equivalence classes*. Each of these approaches incorporates some philosophical interpretation of impreciseness. The granular set approach can be considered as an extension of interval algebra towards unordered sets and lattices or type hierarchy trees. Somewhat similar approaches are also presented in [1] and [12].

The main aim of the paper is to present a concept of granularity in sets and relational tables. The main ideas, initially presented in [5] and [6], are recapitulated in a slightly changed framework and the relationship with rough sets is discussed. Introducing granularity in the sets of data items is aimed at a more general knowledge representation, and knowledge manipulation is moved to higher abstraction level. The analysis is moved to a more abstract level of granularity, which improves efficiency – instead of atomic values of attribute domains one considers now a set or granular values. Algebraic operations on semi-partitions, granular sets and granular relations are defined. The level of granularity is adaptable – it changes according to details of knowledge representation and operations performed.

2 Granular Sets and Their Properties

A *granular set* is a structure composed of a set and a number of its disjoint subsets. It allows to consider arbitrary *granules* of the elements of the base set instead of too numerous and too detailed atomic elements. A granular set with finite number of granules can be constructed even for continuous infinite sets. Moreover, in contrast to discretization methods (where the original set is replaced with a new discrete one), it is still possible to manipulate the atomic elements or to change the partitioning of the base set.

Let a set V and some subsets $V_1, V_2, \ldots V_k$ of V be given.

Definition 1. *The sets $V_1, V_2, \ldots V_k$ form a* partition *of V, iff:*

(1) $V_1 \cup V_2 \cup \ldots \cup V_k = V$ (i.e. partition satisfies the completeness condition*),*
(2) $V_i \cap V_j = \emptyset$ for any $i \neq j$ (i.e. partition satisfies the separation condition*).*

A partition is usually induced by an equivalence relation defined over V. The sets $V_1, V_2, \ldots, V_k$ are equivalence classes; here they are called *blocks*. Note that in practice, we often do not have the possibility to consider all the subsets necessary to form a partition. In such a case the completeness condition is not satisfied. The separation condition is also not necessary however we will often expect that a semi-partition satisfies it.

Definition 2. *A* semi-partition *of V is any collection of its subsets $V_1, V_2, \ldots, V_k$. A semi-partition is* normalized *(in normal form) iff $V_i \cap V_j = \emptyset$ for all $i \neq j$.*

A semi-partition will be also called an *incomplete partition*, or an s-partition for short. An s-partition of V will be denoted as $\sigma(V)$. If not stated explicitly, all the considerations will concern normalized s-partitions. Examples of Fig. 1 show such s-partitions for a nominal and an ordered set.

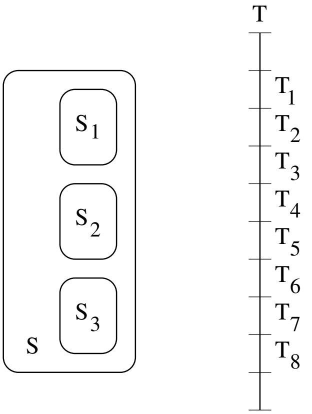


Fig. 1. Two examples of granular sets

If $\sigma(V) = \{V_1, V_2, \ldots, V_k\}$ is an s-partition, then the set of all the elements of V occurring in the s-partition $\sigma(V)$ will be called the *support* of it and it will be denoted as $[\![\sigma(V)]\!]$, and determined as $[\![\sigma(V)]\!] = V_1 \cup V_2 \cup \ldots \cup V_k$.

Note that any family of subsets of some set V can be transformed into a normalized s-partition of V having the same support. Let us consider an arbitrary collection of subsets of V, say $V'_1, V'_2, \ldots, V'_m$ (not necessarily disjoint ones). By subsequent replacing any two sets V'_i and V'_j $(i \neq j)$ such that $V'_i \cap V'_j \neq \emptyset$, with three sets: $V'_i \setminus V'_j$, $V'_j \setminus V'_i$ and $V'_i \cap V'_j$ one can generate an s-partition $\sigma(V)$.

For a given set V a granular set over V is defined as follows.

Definition 3. *A* granular set *G is a pair $G = \{V, \sigma(V)\}$, where $\sigma(V)$ is any s-partition defined on V. If the s-partition $\sigma(V)$ is unnormalized, then the granular set is also said to be an* unnormalized *one.*

The set V is called the domain *of the granular set, while the s-partition $\sigma(V)$ defines the so-called* signature of granularity.

Consider some two granular sets $G = (V, \sigma(V))$ and $G' = (V, \sigma'(V))$, where $\sigma(V) = \{V_1, V_2, \ldots, V_k\}$ and $\sigma'(V) = \{V'_1, V'_2, \ldots, V'_m\}$.

Definition 4. *The support of granular set G is bigger (smaller) than the support of granular set G' iff $[\![\sigma'(V)]\!] \subseteq [\![\sigma(V)]\!]$ ($[\![\sigma(V)]\!] \subseteq [\![\sigma'(V)]\!]$).*

Again, compare two granular sets with the same domain but different signatures. A granular set can provide finer or more rough signature of granularity.

Definition 5. *An s-partition $\sigma'(V) = \{V'_1, V'_2, \ldots, V'_m\}$ is* finer *than an s-partition $\sigma(V) = \{V_1, V_2, \ldots, V_k\}$ iff any set $V_i \in \sigma(V)$ can be expressed as $V_i = V'_{i_1} \cup V'_{i_2} \cup \ldots \cup V'_{i_n}$, where $V'_{i_1}, V'_{i_2}, \ldots V'_{i_n} \in \sigma'(V)$.*

In other words, a finer granular set (or s-partition) is build from smaller blocks and can be used to re-build the more rough one. In general, it can also contain some additional blocks not used for reconstructing the ones of the more rough s-partition.

Now, we will introduce a partial order relation among granular sets and s-partitions. For intuition, a more general granular set (its signature) covers a less general one iff any block of the latter is covered by some block of the former one.

Definition 6. *A set G is* more general *than a set G' ($G \geq G'$) iff $\sigma(V) \geq \sigma'(V)$, where the latter condition means that $\forall V_i' \in \sigma'(V)\ \exists V_j \in \sigma(V)$: $V_i' \subseteq V_j$.*

In fact Def. 6 introduces the Hoare order. The s-partition $\sigma(V)$ will be called *more general* as one operating at more abstract level of granularity. As straightforward consequences of Def. 6 we have the following propositions.

Proposition 1. *The relation of being* more general *defined by Def. 6 is an ordering relation.*

Proposition 2. *If $G \geq G'$ then also $[\![\sigma'(V)]\!] \subseteq [\![\sigma(V)]\!]$.*

Obviously, the inverse proposition is not true in general case. However, the following one holds.

Proposition 3. *Assume that $[\![\sigma'(V)]\!] \subseteq [\![\sigma(V)]\!]$. Then there exists an s-partition $\sigma^0(V)$ such that $[\![\sigma^0(V)]\!] = [\![\sigma(V)]\!]$ and $\sigma^0(V) \geq \sigma'(V)$. Simultaneously, there exists an s-partition $\sigma''(V)$ such that $[\![\sigma'(V)]\!] = [\![\sigma''(V)]\!]$ and $\sigma(V) \geq \sigma''(V)$.*

The meaning of the above proposition is simple: a bigger s-partition (i.e. one having bigger support) can always be transformed into one being also *more general* than the smaller one, but with the same support (intuitively: by gluing together some of its granules; this process is also called reduction). Further to that, a smaller s-partition can always be transformed into one having the same support but also *less general* one (the operation is based on split of the granules).

For granular sets (s-partitions) we can define typical algebraic operations. The *product* of such two s-partitions $\sigma(V)$, $\sigma'(V)$ is defined as:

$$\sigma(V) \cdot \sigma'(V) = \{V_{ij}\colon V_{ij} = V_i \cap V_j \wedge V_i \in \sigma(V) \wedge V_j \in \sigma'(V) \wedge V_{ij} \neq \emptyset\}. \quad (1)$$

Obviously, the product of two s-partitions is an s-partition. Roughly speaking, the product of two s-partitions is the s-partition composed of all nonempty intersections of their blocks. The product of two s-partitions is less general than any of them, i.e. $\sigma(V) \cdot \sigma'(V) \leq \sigma(V)$ and $\sigma(V) \cdot \sigma'(V) \leq \sigma'(V)$.

In a similar way a *composition* of s-partitions can be defined. Let a semi-partition $\sigma(V) = \{V_1, V_2, \ldots, V_k\}$ be given. For any two sets $V_i, V_j \in \sigma(V)$ we define the following partition generation operation as $V_i \sqcap V_j = \{V_i \setminus V_j, V_j \setminus V_i, V_i \cap V_j\}$. The operation can be extended to the whole semi-partition. The semi partition $\prod(\sigma(V))$ is evaluated as follows:

1. $\prod(\sigma(V))_0 = \sigma(V)$
2. If there exist sets $V_i, V_j \in \prod(\sigma(V))_n$ such that $V_i \cap V_j \neq \emptyset$ then $\prod(\sigma(V))_{n+1} = \prod(\sigma(V))_n - \{V_i, V_j\} \cup V_i \sqcap V_j$, else $\prod(\sigma(V))_{n+1} = \prod(\sigma(V))_n$.
3. If $\prod(\sigma(V))_{n+1} = \prod(\sigma(V))_n$ then $\prod(\sigma(V)) = \prod(\sigma(V))_n$.

The semi-partition $\bigsqcap(\sigma(V))$ is a normalized one and the result is independent on the order of applying of the $\sqcap$ operator. Let $x \in [\![\sigma(V)]\!]$ and let $V_x \in \bigsqcap(\sigma(V))$ denote the set containing x. The semi-partition can be divided into two disjoint subsets: $\sigma_x(V) = \{V_i \in \sigma(V) \colon x \in V_i\}$ and $\sigma_{\bar{x}}(V) = \sigma(V) - \sigma_x(V)$. Hence, the following equality holds:

$$V_x = \bigcap \sigma_x(V) - \bigcup \sigma_{\bar{x}}(V). \tag{2}$$

The *composition* of s-partitions $\sigma(V)$ and $\sigma(V)$ is defined as follows:

$$\sigma(V) \circ \sigma'(V) = \bigsqcap(\sigma(V) \cup \sigma'(V)). \tag{3}$$

For any two s-partitions $\sigma(V)$ and $\sigma'(V)$ we define also a *cover* of them, i.e. an s-partition covering all the elements of V belonging to some component set of at least one of them. For a semi-partition $\sigma(V) = \{V_1, V_2, \ldots, V_k\}$ the following sum operation is introduced:

1. $\bigsqcup(\sigma(V))_0 = \sigma(V)$
2. If there exist sets $V_i, V_j \in \bigsqcup(\sigma(V))_n$ such that $V_i \cap V_j \neq \emptyset$ then $\bigsqcup(\sigma(V))_{n+1} = \bigsqcup(\sigma(V))_n - \{V_i, V_j\} \cup \{V_i \cup V_j\}$, else $\bigsqcup(\sigma(V))_{n+1} = \bigsqcup(\sigma(V))_n$.
3. If $\bigsqcup(\sigma(V))_{n+1} = \bigsqcup(\sigma(V))_n$ then $\bigsqcup(\sigma(V)) = \bigsqcup(\sigma(V))_n$.

The semi-partition $\bigsqcup(\sigma(V))$ is a normalized one and the result is independent on the order of applying of the $\sqcup$ operator. Let $x \in [\![\sigma(V)]\!]$ and let $V_x \in \bigsqcup(\sigma(V))$ denotes the set containing x. Let the set $\sigma_x(V)$ be defined as follows:

1. $\sigma_x(V)_0 = \{V_i\}$, where $V_i \in \sigma(V)$ and $x \in V_i$.
2. $\sigma_x(V)_{n+1} = \sigma_x(V)_n \cup \{V_i \in \sigma(V) - \sigma_x(V)_n \colon V_i \cap (\bigcup \sigma_x(V)_n) \neq \emptyset\}$.
3. If $\sigma_x(V)_{n+1} = \sigma_x(V)_n$ then $\sigma_x(V) = \sigma_x(V)_n$.

Hence, $V_x = \bigcup \sigma_x(V)$.

The *cover* of s-partitions $\sigma(V)$ and $\sigma'(V)$ is defined as follows:

$$\sigma(V) + \sigma'(V) = \bigsqcup(\sigma(V) \cup \sigma'(V)). \tag{4}$$

For intuition, both s-partitioning and generating a cover are kinds of operations preserving covering of the same elements of V which are covered by the initial family of subsets (the support). However, in case of s-partitioning one preserves also the definition of initial signatures (structuring) (e.g. the boundaries of intervals of characteristic subsets of V), while in the case of cover generation a kind of maximal reduction of the subsets is performed. There is also $\sigma(V) + \sigma'(V) \geq \sigma(V)$ and $\sigma(V) + \sigma'(V) \geq \sigma'(V)$.

Consider a *reduction* operation of transforming an s-partition into another, more general one, by *gluing* some of its elements (non-overlapping ones). The reduction of an s-partition consists in replacing several blocks with an equivalent single block. The generated output is aimed to be a normalized s-partition, so in the case of intervals, gluing is allowed only for intervals which meet or the so-called non-convex intervals must be admitted. The generated s-partition is equivalent with regard to the elements covered, but simultaneously it is more general than the input one.

Finally, consider the so-called *split* operation. The operation consists in replacing each element V_i of the initial s-partition with a family of its subsets such that the sum of them is equal to V_i. The result of the split operation is less general than the initial s-partition. In general, the split operation gives no unique result. For this reason, it may be useful to define the so-called *induced split*, where the result depends on another s-partition which is compared with the one under interest.

3 Granular Sets and Rough Sets

Granular sets as introduced in Section 2 may constitute a tool for defining rough sets. However, since the assumptions of the presented approach are weaker than in case of a partition induced by an equivalence relation, it is not always possible to define the upper approximation.

Consider a set V and an s-partition $\sigma(V) = \{V_1, V_2, \ldots, V_n\}$. Let X denote some subset of V, i.e. $X \subseteq V$. The lower approximation of X with s-partition $\sigma(V)$ is defined as

$$\underline{R}X = \{V_i \in \sigma(V) \colon V_i \subseteq X\}.$$

The lower approximation of X always exists; in some cases it can be the empty set. Also $\underline{R}X \subseteq X$ in the sense $[\![\underline{R}X]\!] \subseteq X$.

Contrary to classical rough set theory, the upper approximation defined with a particular s-partition may not exist, i.e. it may be empty. We define an approximation of set X in the following way:

$$RX = \{V_i \in \sigma(V) \colon V_i \cap X \neq \emptyset\}.$$

Note that contrary to the case of partitions based on equivalence relation, it can be the empty set. Further, in some cases the basic property that $X \subseteq \overline{R}X$ (in the sense that $X \subseteq [\![\overline{R}X]\!]$) may be violated.

For practical reasons, to obtain the upper approximation covering X (e.g. when verifying completeness of systems) it may be of interest to look for the uncovered cases, i.e. the completion of an approximation to an upper approximation – such that all elements of X will be covered. In order to do that we first define the completion of an approximation as

$$\overline{RX} = X \setminus RX.$$

The upper approximation can be defined now as

$$\overline{R}X = RX \cup \overline{RX}.$$

4 Granular Relations

Using the presented idea of granular set, a granular relation can be defined in a straightforward way. Consider some collection of sets $D_1, D_2, \ldots, D_n$. Let there be defined some granular sets on them, i.e. $G_1 = (D_1, \sigma_1(D_1)), G_2 = (D_2, \sigma_2(D_2)), \ldots, G_n = (D_n, \sigma_n(D_n))$.

Definition 7. *A* granular relation $R(G_1, G_2, \ldots, G_n)$ *is any set* $R_G \subseteq U_G$ *where*

$$U_G = \sigma_1(D_1) \times \sigma_2(D_2) \times \ldots \times \sigma_n(D_n). \tag{5}$$

The set U_G *will be referred to as* granular universe *or* granular space. *If at least one of the granular sets was unnormalized, the relation is also said to be unnormalized one.*

The elements (rows) of a granular relation will be called *boxes*. Note that in fact a granular relation defines a kind of meta-relation, i.e. one based on sets instead of single elements. In fact, if R is a relation defined as $R \subseteq D_1 \times D_2 \times \ldots \times D_n$, then any tuple of R is like a thread in comparison to elements of R_G which are like a cord or a pipe.

Consider an example concerned with time-table development for a university or a school. First, there is certainly a finite set of students, say S. Instead of specifying for each student his personal schedule, the university authorities consider "granules" of them, i.e. years, groups, etc. If S_1, S_2, and S_3 are the groups of the first year, then a granular structure $G(S) = (S, \{S_1, S_2, S_3\})$ can be considered useful when assigning classes to the students of the first year. Further, time is also considered granular – instead of precise exact time one would rather consider traditional intervals, such as lessons (e.g. 45 or 55 minutes each) or periods of the length 1h30min which form a frame for constructing the schedule. Let T be the discrete set of time values from 7:00 to 21:00, and let T_1=[8:00,9:30], T_2=[9:30,11:00], T_3=[11:00,12:30], T_4=[12:30,14:00], T_5=[14:00,15:30], T_6=[15:30,17:00], T_7=[17:00,18:30] and T_8=[18:30,20:00]. Some other sets, such as the set of professors P, the set B of rooms or the set of classes (subjects) C are considered here at the level of single items. For simplicity, we focus on the schedule for some specific day, so the problem is to assign each group a professor, room and subject for any legal time interval. If $P = \{p_1, p_2, p_3\}$ is the set of professors, $B = \{b_1, b_2\}$ is the set of rooms and $C = \{c_1, c_2, c_3\}$ is the set of subjects, the relation representing the schedule can be as shown in Fig. 2.

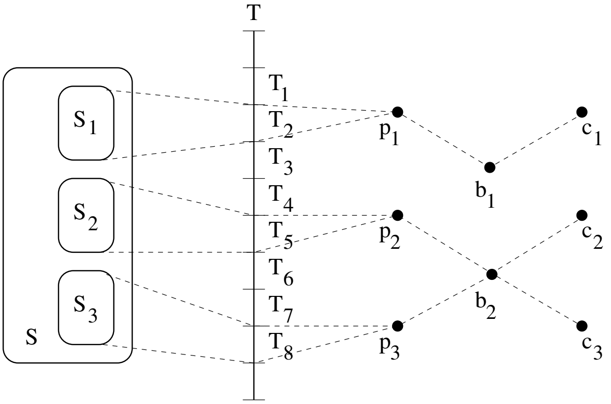


Fig. 2. Example of a granular relation

The relation shown in Fig. 2 is defined by the following tuples: $\{(S_1, T_2, p_1, b_1, c_1), (S_2, T_6, p_2, b_2, c_2), (S_3, T_8, p_3, b_2, c_3)\}$. In the tuples of this relation only the first two elements of each tuple are granules; the other three are basic items.

5 Granular Knowledge Representation Systems

Granular sets and granular relations can be applied to develop *granular knowledge representation systems* (also called *extended tabular systems*) [7]. In comparison with knowledge representation systems considered in [9], nonatomic values of attributes are admissible. In similar way granular decision tables can be introduced.

Table 1. Optician Decision Table

Number	Age	Spectacle	Astigmatic	Tear p.r.	Decision
1	y	m	y	n	h
2	y	h	y	n	h
3	p	m	y	n	h
4	q	m	y	n	h
5	y	m	n	n	s
6	y	h	n	n	s
7	p	m	n	n	s
8	p	h	n	n	s
9	q	h	n	n	s
10	y	m	n	r	n
11	y	m	y	r	n
12	y	h	n	r	n
13	y	h	y	r	n
14	p	m	n	r	n
15	p	m	y	r	n
16	p	h	n	r	n
17	p	h	y	r	n
18	p	h	y	n	n
19	q	m	n	r	n
20	q	m	n	n	n
21	q	m	y	r	n
22	q	h	n	r	n
23	q	h	y	r	n
24	q	h	y	n	n

After Pawlak [9] let us consider the following decision table (see Tab. 1). The attributes and their domains are as follows:

- A_1 := age; $D_1 = \{y, p, q\}$, where: y – young, p – pre-presbyotic, q – presbyotic,
- A_2 := spectacle; $D_2 = \{m, h\}$, where: m – myope, h – hypermyope,
- A_3 := astigmatic; $D_3 = \{n, y\}$, where: n – no, y – yes,
- A_4 := tear production rate; $D_4 = \{r, n\}$, where: r – reduced, n – normal,

- D := type of contact lenses (decision attribute); $D_D = \{h, s, n\}$, where: h – hard contact lenses, s – soft contact lenses, n – no contact lenses.

The considered table is complete and deterministic. Methods based on the rough set theory can be used to reduce such a decision table. The reduction algorithm consists in the elimination of conditions from a decision table, which are unnecessary to make decisions specified in the table. Finally, a table with only nine decision rules can be received (see Tab.2).

Table 2. The reduced form of the Optician Decision Table (using rough set approach)

Number	Age	Spectacle	Astigmatic	Tear p.r.	Decision
1	y	–	y	n	h
2	–	m	y	n	h
3	y	–	n	n	s
4	p	–	n	n	s
5	–	h	n	n	s
6	–	–	–	r	n
7	p	h	y	–	n
8	q	h	y	–	n
9	q	m	n	–	n

If granular sets and relations are considered further reduction is possible. The most reduced form of the considered decision table is presented in Tab. 3. The third and sixth row contains non-atomic values of the attribute Age.

Table 3. The reduced form of the Optician Decision Table

Number	Age	Spectacle	Astigmatic	Tear p.r.	Decision
1	y	–	y	n	h
2	–	m	y	n	h
3	$\{y, p\}$	–	n	n	s
4	–	h	n	n	s
5	–	–	–	r	n
6	$\{p, q\}$	h	y	–	n
7	q	m	n	–	n

Row 3 of table 3 is the result of gluing rows 3 and 4 of table 2. Similarly, row 6 of table 3 is the result of gluing rows 7 and 8 of table 2.

6 Summary

The paper presents a concept of granular knowledge representation and manipulation. The key notions discussed here are the one of granular set and granular relation, both of them base on the idea of s-partition of a set. It has been shown that granular sets and relations can be applied to develop granular knowledge representation systems, which enable to represent knowledge in more condense form.

The granular approach presented in the paper can be used for efficient knowledge representation in rule-based systems [7]. Some directions of possible future works include development of efficient algorithms for verification of theoretical properties, such as subsumption among rules, completeness of sets of rules, possibility of reduction, etc. Moreover, further extensions of granular attributive logic are also explored [8].

References

1. Järvinen, J., Kortelainen, J.: A Note on Definability in Rough Set Theory. In: De Baets, B., De Caluwe, R., De Tré, G., Fodor, J., Kacprzyk, J., Zadrożny, S. (eds.): Current Issues in Data and Knowledge Engineering, Akademicka Oficyna Wydawnicza EXIT, Warsaw, Poland (2004)
2. Peters, J.F., Pawlak, Z., Skowron, A.: A rough set approach to measuring information granules. In: Proc. of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment (COMPSAC'02), Oxford, pp. 1135–1139 (2002)
3. Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic. In: Theory and Applications, Prentice-Hall, Englewood Cliffs (1995)
4. Ligęza, A.: Intelligent data and knowledge analysis and verification; towards a taxonomy of specific problems. In: Vermesan, A., Coenen, F. (eds.) Validation and Verification of Knowledge Based Systems: Theory, Tools and Practice, pp. 313–325. Kluwer Academic Publishers, Boston (1999)
5. Ligęza, A.: Granular sets and granular relations: Towards a higher abstraction level in knowledge representation. In: Intelligent Information Systems 2002, pp. 331–340. Physica-Verlag, Heidelberg (2002)
6. Ligęza, A.: Granular sets and granular relations for algebraic knowledge management. In: Smart engineering system design: neural networks, fuzzy logic, evolutionary programming, complex systems and artificial life: Proc. of the Artificial Neural Networks in Engineering Conference (ANNiE 2003). ASME Press Series on Intelligent Engineering Systems Through Artificial Neural Networks, St. Louis, Missouri, USA, vol. 13, pp. 169–174 (2003)
7. Ligęza, A.: Logical foundations of rule-based systems. In: Studies in Computational Intelligence, vol. 11, Springer, Berlin (2006)
8. Ligęza, A., Nalepa, G.J.: Knowledge Representation with Granular Attributive Logic for XTT-based Expert Systems. In: Prof. of the 20th International FLAIRS Conference, Key West, Florida (to appear)
9. Pawlak, Z.: Rough Sets. In: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht, The Netherlands (1991)
10. Peters, J.F., Skowron, A., Suraj, Z., Rząsa, W., Borkowski, M.: Clustering: A rough set approach to constructing information granules. In: Proceedings of the 6th International Conference on Soft Computing and Distributed Processing (SCDP'02), Rzeszów, pp. 57–61 (2002)
11. Peters, J.F., Pawlak, Z., Skowron, A.: A rough set approach to measuring information granules. In: Proc. of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment, Oxford, 2002, pp. 1135–1139 (2002)
12. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
13. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)

Inference and Reformation in Flow Graphs Using Granular Computing

Huawen Liu[1,2], Jigui Sun[1,2], Changsong Qi[1,3], and Xi Bai[1,2]

[1] College of Computer Science, Jilin University, Changchun 130012, China
Huaw.Liu@gmail.com, JgSun@jlu.edu.cn, Baixi2@163.com
[2] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Changchun 130012, China
[3] Department of Computer, Tonghua Normal College, Tonghua 134002, China
ChangSongQi@gmail.com

Abstract. Flow graph (FG) is a new mathematical model which can be used for representing, analyzing, and discovering knowledge in databases. Due to its well-structured characteristics of network, FG is naturally consistent with granular computing (GrC). Meanwhile, GrC provides us with both structured thinking at the philosophical level and structured problem solving at the practical level. In this paper, the relationship between FG and GrC will be discussed from three aspects under GrC at first, and then inference and reformation in FG can be easily implemented in virtue of decomposition and composition of granules, respectively. As a result of inference and reformation, the reformed FG is a reduction of the original one.

Keywords: Flow Graphs, Granular Computing, Reduction.

1 Introduction

Recently, Pawlak introduced flow graphs(shortly, FG), which is a new graphical model for representing knowledge and reasoning data, in his initiated paper [7]. After then, series of related papers, such as [8,9,10,11,12,13], continuously have been put forward to place emphasis upon its importance in data analysis. In these literatures, Pawlak have discussed the relationships among FG, rough sets, decision systems, Bayes' theorem, data mining and decision tree in theory aspects and these works pave the way for application of FG in many fields [13]. For instance, FG was linked up with decision systems in [9,11] and [8], and tied up with rough sets in his recent paper [12].

Instead of optimal flow, FG is in pursuit of information flow distribution study in quantified view. It is also a graph representation of decision algorithm in some ways. In FG, each node denotes one element set and branch describes flow distribution between nodes. In addition, every path from the root to a leaf is a decision rule, and a FG is the set of decision rules. Each branch or path associates with three coefficients, i.e., the strength, certainty and coverage factors. Moreover, these information flow distributions in FG are governed by Bayes' formula and abide by flow conservation equations [12].

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 261–270, 2007.

As a mathematical model of finding and mining knowledge, FG has some advantages, such as intuitional representation, straightforward computation, explicit relations and parallel processing. Owing to these, FG has been received much considers by researchers after it was proposed. For example, Butz et al. argued that a FG can be transformed into a Bayesian network in polynomial times [1]. In addition, they figured out that flow graph inference algorithm has exponential complexity and then presented a polynomial time complexity algorithm for inference in FG [2]. While Kostek and Czyzewski successfully applied FG in musical metadata retrieval, in order to improve its efficiency [3,4].

Although quantification measures play important roles in data mining, however, qualitative analysis is also necessary. The main reason is that it can depict accurately the complex problems using less information and make a reasoning from data efficiently. Based on this fact, Sun et al. proposed an extended FG(shortly, EFG) in [16], which can exactly describe the relationships among nodes in network. This extension of FG not only has the capability of FG in the quantification aspect, but also can be interpreted by information systems or granular computing(GrC) from qualitative view. Motivated by the practical needs for simplification, clarity, low cost, approximation, and tolerance of uncertainty, GrC is more about a philosophical way of thinking and a practical methodology of problem solving deeply rooted in human mind. By effectively using levels of granularity, GrC provides a systematic, natural way to analyze, understand, represent, and solve real world problems [21].

Due to its well-structural network, EFG is consistent with granular computing(GrC) in natural way [17]. In this paper, we will firstly investigate to the relationship between EFG and GrC from three aspects in GrC, namely, granulation of the universe, relationships of granules and computing with granules [22]. As we know, reasoning from data is very vital in data mining and it determines whether the model is practicable and acceptable by users or not. However, it is more preferable if FG can employ less data and compact structure to reasoning data without loss of its power. There is no exception to our cases. We will discuss some issues of inference and reformation in FG in details under the framework of GrC later, and corresponding algorithms will also be given. After performed decoding and encoding of nodes one after the other, a new EFG, which is a reduction of the original one, can be achieved.

The structure of the rest is organized as follows. Section 2 briefly recalls some concepts of flow graph and its extension. In Section 3, the relationship between EFG and GrC will be discussed. Section 4 presents the inference and reformation procedures in EFG according to decomposition and composition on granules, respectively. Moreover, the corresponding algorithms will be given. Finally, some concluding remarks are shown in Section 5.

2 Flow Graphs and Its Extension

In this section, some concepts of flow graph and its extension will be recalled briefly. More notations can be consulted [10] and [16].

A flow graph (FG) is a *directed, acyclic, finite* graph $G = (N, B, \varphi)$, where N is a set of *nodes*, $B \subseteq N \times N$ is a set of *directed branches*, $\varphi : B \to R^+$ is a *flow function* and R^+ is the set of non-negative reals [10].

If $(n, n')\in B$ then n is an *input* of n' and n' is an *output* of n. $\varphi(n, n')$ is the *throughflow* from n to n'. $I(n)$ and $O(n)$ are the sets of all inputs or outputs of n, respectively, that is, $I(n)=\{n'\in N|(n', n)\in B\}$ and $O(n)=\{n'\in N|(n, n')\in B\}$. For each node n in FG, its *inflow* and *outflow* are $\varphi_+(n)=\sum_{n'\in I(n)} \varphi(n', n)$ and $\varphi_-(n)=\sum_{n'\in O(n)} \varphi(n, n')$, respectively.

From these definitions, we have the fact that FG is a quantification graph, that is, it represents simply relations among nodes using information flow distribution. Although some valuable results can be achieved using quantitative factors, however, it is not sufficient to depict concretely and exactly relationships among nodes. In addition, qualitative factors, as well as quantitative ones, play very important roles in data mining for they can bring more reasonable outcomes to data analysis. Therefore, an extension of FG has been proposed in [16] according to the information or objects flowing in the network.

An extension of flow graph (EFG) is a *directed, acyclic, finite* graph $G=(E, N, B, \varphi, \alpha, \beta)$, where E and N are the set of *objects* and *nodes* respectively. $B\subseteq N\times N$ is *directed branches* set, $\varphi{:}B\to 2^E$ is the set of *objects* which flow through branches and $\alpha, \beta : B\to[0, 1]$ are thresholds of *certainty* and *decision*, respectively.

Node n is *input(father)* of n', if $(n, n') \in B$. Likewise, n' is *output(child)* of n. The sets of *fathers* and *children* of node n denote respectively as $I(n)$ and $O(n)$ as defined in FG. A node n is called a *root* if $I(n) = \emptyset$ holds and n is a *leaf* if $O(n) = \emptyset$. n is a *internal* node if n is neither a *root* nor a *leaf*. The *inflow* and *outflow* of node n are respectively defined as $\varphi_+(n) = \bigcup_{n'\in I(n)} \varphi(n', n)$ and $\varphi_-(n) = \bigcup_{n'\in O(n)} \varphi(n, n')$. In addition, we assume that for any *internal* node n, $\varphi(n) = \varphi_+(n) = \varphi_-(n)$ in EFG. Similarly, *Input* and *output* of G are $I(G) = \{n \in N|I(n) = \emptyset\}$ and $O(G) = \{n \in N|O(n) = \emptyset\}$, respectively.

In an EFG G, each branch $(n, n') \in B$ is also associate with three factors, i.e., *strength*, *certainty* and *coverage*. A *directed path* from n to n', denoted by $[n...n']$, is a sequence of nodes $n, ..., n'$, where $(n_i, n_{i+1}) \in B$ for $1 \leq i \leq m - 1$, $n_1 = n, n_m = n'$ and $\bigcap_{i=1}^{m} \varphi(n_i, n_{i+1}) \neq \emptyset$. What's more, the *support* of the path $[n_1...n_m]$ is $\varphi(n_1...n_m) = \bigcap_{i=1}^{m} \varphi(n_i, n_{i+1})$. Interested readers can consult Ref. [16] to get more about EFG.

The related definitions about EFG tell us that if we only cast our lights on quantity of objects flowing through branches in G rather than concrete objects, i.e. $|\varphi(n_i, n_j)|/|E|$, and $\alpha = 0, \beta = 0$, then the EFG can be degraded into FG, where $|X|$ is the cardinality of X. That is, an EFG has the capability of FG.

For convenience, in this paper, we assume that an EFG is organized as several layers. In the same layer, there does not exist any branch among nodes and one object only belongs to one node, that is, branch only exists between nodes in different layers. In addition, if $(n, n') \in B$ in EFG, then $\varphi(n_i, n_j) \neq \emptyset$.

Example 1. Six patients have been arranged into four groups(five layers) according to different symptoms, such as *temperature*, *headache*, *muscle pain* and *flu*, in testing whether a patient has catched flu or not, and its corresponding EFG $G =$

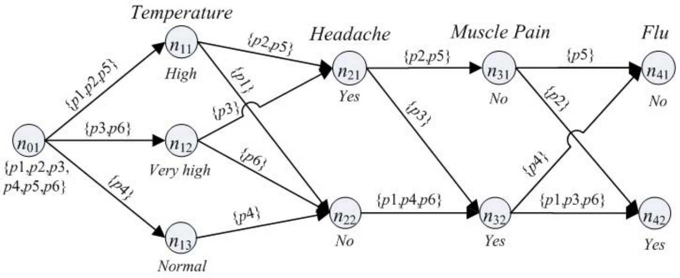


Fig. 1. An EFG G

$(E, N, B, \varphi, \alpha, \beta)$ is presented in Fig. 1, where $E = \{p_1, p_2, p_3, p_4, p_5, p_6\}, N = \{n_{01}, n_{11}, n_{12}, n_{13}, n_{21}, n_{22}, n_{31}, n_{32}\} \cup \{n_{41},\ n_{42}\}, \alpha = 0, \beta = 0$.

In this EFG, the root of G is n_{01} and leaves are n_{41} and n_{42}. The input and output of node n_{21} are $I(n_{21}) = \{n_{11}, n_{12}\}$ and $O(n_{21}) = \{n_{31}, n_{32}\}$, respectively. With branch $(n_{11}, n_{21}) \in B$, its throughflow, $\varphi(n_{11}, n_{21}) = \{p2, p5\}$, denotes the temperature of patients $p2$ and $p5$ is high and they have headache. For node n_{21}, its inflow is $\varphi_+(n_{21}) = \varphi(n_{11}, n_{21}) \cup \varphi(n_{12}, n_{21}) = \{p2, p3, p5\}$ and outflow is $\varphi_-(n_{21}) = \varphi(n_{21}, n_{31}) \cup \varphi(n_{21}, n_{32}) = \{p2, p3, p5\}$, i.e., $\varphi(n_{21}) = \varphi_+(n_{21}) = \varphi_-(n_{21})$. In addition, the sequence of $n_{01}, n_{12}, n_{21}, n_{32}$ is a path and its degrees of certainty and coverage are $cer(n_{01}, n_{12}, n_{21}, n_{32}) = 1$ and $cov(n_{01}, n_{12},\ n_{21}, n_{32}) = 1/4$, respectively. □

3 Relationship Between EFG and GrC

As a tool of data analysis in data mining, FG has been interpreted by decision algorithms, probability and rough sets [12]. Since EFG has some excellent formal features in describing information flow and shares some common with GrC in structural way, however, the relationship between EFG and GrC will be investigated in this section.

3.1 Granulation

With respect to a layer $l \in L$ in EFG G, two objects $x, y \in E$ may flow through the same node n, i.e., $x, y \in \varphi(n)$. In this case, one can not distinguish x and y according to $n \in l$. This means that x, y can be grouped into a *granule*.

Definition 1. *Let $G = (E, N, B, \varphi, \alpha, \beta)$ be an EFG, if $x, y \in E$ flow through $n \in N$, i.e., $x, y \in \varphi(n)$, then we will say x, y belong to the same granule $g(n)$, denoted as a pair $(n, m(n))$, where n and $m(n)$ are the descriptor and meanings of the granule, respectively, and $x, y \in m(n)$.*

In other words, the meanings of the granule $g(n)$ consist of all objects flowing through node n, i.e., the equation $m(n) = \varphi(n)$ holds for $\forall n \in N$ in EFG.

Definition 2. *Let $G = (E, N, B, \varphi, \alpha, \beta)$ be an EFG, a granule $g(n)$ will be called an element granule if n is an single node, i.e., $n \in N$.*

As we know, each object $x \in E$ only flows through one node in layer l in EFG. Therefore, the family of granules $F(\{l\}) = \{g(n)|m(n) \neq \emptyset, \forall n \in l\}$ forms a partition over E, denoted by E/l, and its corresponding equivalence relation R_l on E is $xR_ly \Leftrightarrow x \in m(n) \wedge y \in m(n) \wedge n \in l$. In addition, the equivalence class of x with reference to R_l is $[x]_{R_l} = \{y \in E|xR_ly\}$ and each $[x]_{R_l}$ is a granule, i.e., $g(n) = (n, [x]_{R_l})$, where $n \in R_l$.

Definition 3. *Let $g(n), g(n')$ be two granules, the combined granule $g(n \wedge n')$ of the granules is $g(n \wedge n') = g(n \wedge n', m(n) \cap m(n'))$.*

In EFG G, each node is an element granule. Thus, the combined granule of two element granules, e.g. n and n', is the branch (n, n') and its meanings is the flowthrough $\varphi(n, n')$. In other words, every node or branch in EFG is a granule. Furthermore, each path is also considered as a granule by combining the nodes and branches and its flowthrough is the meanings of the granule. For example, let $[n...n', n'']$ be a path, then $\varphi(n...n', n'') = \varphi(n...n') \cap \varphi(n'')$ holds. This means that the granule $g(n \wedge ... \wedge n' \wedge n'')$ consists of the granules $g(n \wedge ... \wedge n')$ and $g(n'')$ and its meanings $m(n \wedge ... \wedge n' \wedge n'') = \varphi(n...n', n'')$.

In EFG G, $l, l' \in L$ are different layers. Since $m(n \wedge n') = \varphi(n) \cap \varphi(n')$ holds for any $n \in l, n' \in l'$, the family of granules $F(\{l, l'\}) = \{g(n, n')|m(n, n') \neq \emptyset, \forall n \in l \wedge \forall n' \in l'\}$ also forms a partition $E/\{l, l'\}$ over E.

Definition 4. *A granule $g(n)$ is finer than granule $g(n')$, denoted as $g(n) \subseteq g(n')$, if $m(n) \subseteq m(n')$. A family of granules $F(L)$ is finer that another $F(L')$, denoted as $F(L) \subseteq F(L')$, if there exists a granule $g(n')$ in $F(L')$ for granule $\forall g(n) \in F(L)$, such that $g(n) \subseteq g(n')$.*

Obviously, $F(L) \subseteq F(L')$, if $L' \subseteq L$. In the light of Def. 3, we have the fact that the longer the path is, the less of the flowthrough and the finer the corresponding granule is. What's more, the all paths, which start at the root in EFG, with the same length compose a partition on E.

Example 2. **(cont.)** In Fig.1, *temperature*(l_1), *headache*(l_2), *muscle pain*(l_3) and *flu*(l_4) form four partitions over E and its granules are shown below:
$l_1 : g(n_{11}) = (n_{11}, \{p_1, p_2, p_5\})$, $g(n_{12}) = (n_{12}, \{p_4\})$, $g(n_{13}) = (n_{12}, \{p_3, p_6\})$;
$l_2 : g(n_{21}) = (n_{21}, \{p_2, p_3, p_5\})$, $g(n_{22}) = (n_{22}, \{p_1, p_4, p_6\})$;
$l_3 : g(n_{31}) = (n_{31}, \{p_2, p_5\})$, $g(n_{32}) = (n_{32}, \{p_1, p_3, p_4, p_6\})$;
$l_4 : g(n_{41}) = (n_{41}, \{p_1, p_2, p_3, p_6\})$, $g(n_{42}) = (n_{42}, \{p_4, p_5\})$;
The meanings of granules with respect to $L' = \{l_1, l_2\}$ and $L'' = \{l_1, l_2, l_3\}$ are
$L' : m(n_{11} \wedge n_{21}) = \{p_2, p_5\}$, $m(n_{11} \wedge n_{22}) = \{p_1\}$, $m(n_{12} \wedge n_{22}) = \{p_4\}$,
$m(n_{13} \wedge n_{21}) = \{p_3\}$, $m(n_{13} \wedge n_{22}) = \{p_6\}$
$L'' : m(n_{11} \wedge n_{21} \wedge n_{31}) = \{p_2, p_5\}$, $m(n_{11} \wedge n_{22} \wedge n_{32}) = \{p_1\}$,
$m(n_{12} \wedge n_{22} \wedge n_{32}) = \{p_4\}$, $m(n_{13} \wedge n_{21} \wedge n_{32}) = \{p_3\}$, $m(n_{13} \wedge n_{22} \wedge n_{32}) = \{p_6\}$.
Obviously, the granules in $g(L'')$ are finer than those in $g(L')$. □

3.2 Decomposition and Composition of Granules

Decomposition and composition in problem-solving are necessary capabilities for granules, because these operations can carry the point to traverse views among different levels of granularity. In GrC, Granule decomposition deals with the change from a coarse granule to finer ones in order to provide more details for data analysis, whereas composition deals with the shift from several fine granules to a coarser one to make distinct granules no longer differentiable by discarding some details [21]. However, there is no exception in EFG.

According to the analysis in subsection 3.1, we know that the granule model of EFG is a partition one [20] and the switch from one to another can be easily achieved under the framework of quotient space theory [23]. When a granule is decomposing into several ones, more details or extra information are needed. As a result, the new granules are usually finer than their father. In addition, information is represented as layers in EFG and each layer divides E into a partition according to Def.2. Hence, granule decomposition is in fact that granules in one partition are broken down into finer ones by using those granules in another partition, and the finer granules in EFG are, the more layers are required.

Definition 5. *In GrC model of EFG, granules decomposition function is a mapping $Dec : \mathcal{F}\times\mathcal{F}\rightarrow\mathcal{F}$, where $\mathcal{F}$ is the family of granules which compose a partition over E, that is, $\bigcup_{g(n)\in\mathcal{F}} m(n)=E$ and $m(n)\cap m(n')=\emptyset$ for $\forall g(n), g(n')\in\mathcal{F}$.*

Example 3. **(cont.)** Assume that partitions $F(\{l_1, l_2\})$ and $F(\{l_3\})$ are known. Since $F(\{l_1, l_2\})$ is too coarse, we can use $F(\{l_3\})$ to divide it into finer one, $F(L'')$, as shown in *Example* 2. □

On the contrary, granule composition is the procedure that extracts the common information from granules regardless of distinct ones for the purpose of generalization from specificity. In EFG, the common information means that the descriptors of granules contain in one same layer, that is, the corresponding paths have the same sub-path. For example, one granule $g(n_{11})$ is the composition of granules $g(n_{11} \wedge n_{21})$ and $g(n_{11} \wedge n_{32})$ for they share the common information *temperature*, where $n_{11} \in l_1$ is a node in *temperature* layer in Example 2.

Definition 6. *In GrC model of EFG, granules composition function is a mapping $Com : \mathcal{F}\rightarrow\mathcal{F}\times\mathcal{F}$, where $\mathcal{F}$ is the family of granules which compose a partition over E, that is, $\bigcup_{g(n)\in\mathcal{F}} m(n)=E$ and $m(n)\cap m(n')=\emptyset$ for $\forall g(n), g(n')\in\mathcal{F}$.*

In EFG, the size of a partition $F(L\cup L')$ on E denotes how much the knowledge we have. So the function *Com* gets the shared knowledge $F(L)$ from all granules in $F(L \cup L')$ and changes the former knowledge into a coarser one $F(L')$ at the same time. Since *Dec* and *Com* work under a partition of granule model, they are the special cases of binary neighborhood relations [5]. Usually, $g(n) \in Dec(Com(g(n)))$ and $g(n) = Com(Dec(g(n)))$ hold for granule $g(n)$.

4 Inference and Reformation in EFG

In GrC model of EFG, a directed path represents a granule and the longer the path, the finer the granule. Thus a nested granulations hierarchy is constituted by all granules corresponding to paths starting from the root. Hereafter, granules denote those paths stemming from the root in this paper and *leaves* are arranged in *decision layer* DL and others in *condition layers* CL. An EFG is the graph model of rules in some way. Each path from the root to a leaf is a decision rule, where the leaf is the decision part and others belong to the condition part. Inference in EFG in fact is a procedure of granule decomposition. According to Def. 3, Def. 4 and Def. 5, we can immediately obtain the following proposition.

Proposition 1. *In GrC model of EFG,* $F(L \cup \{l\})$*=*$Dec(F(L), F(\{l\}))$*, where* L*⊆*CL*,* l*∈*CL *and* $F(L), F(\{l\})$ *are families of granules generated by* L *and* $\{l\}$*, respectively.*

In the granules hierarchy, the granules in the same layer form a partition over E and the granules in the i-th layer are finer than those in j-th layer if $i > j$. Moreover, a granule in high levels can be split into several disjoint finer granules in the next levels by granule decomposition, that is, the inference can be easily achieved by employing granule decomposition. Meanwhile, more details about the granule can be obtained. Based on this fact, granules decomposition (or composition) can be implemented in top-down (or bottom-up) method.

The main idea of the inference is that E has been parted by the root firstly, and then the partition $F(L)$ is divided by layer $\{l\}$∈CL, step by step, into a new one $F(\{l\}$∪$L)$. If a granule $g(n)$∈$F(\{l\}$∪$L)$ is finer than a decision granule $g(n')$⊆$F(DL)$, then $g(n)$ will be removed from $F(\{l\}$∪$L)$, otherwise it would be divided farther. The algorithm will be terminated when all granules are classified or all layers in CL are used out. More details about inference are given in Alg. 1.

In contrast, the common knowledge $F(\{l\})$ can also be drawn from the specified knowledge $F(L \cup \{l\})$ by granule composition. Similarly, the following fact holds in the light of Def. 3, Def. 4 and Def. 6.

Proposition 2. *In GrC model of EFG,* $Com(F(L$∪$\{l\}))$*=*$(F(L), F(\{l\}))$*, where* L*⊆*CL*,* l*∈*CL *and* $F(\{l\})$ *is the family of coarse granules(i.e., common knowledge) generated* $\{l\}$ *and* $F(L)$ *is the knowledge without common knowledge with respect to* l*.*

According to this proposition, a hierarchy graph can be constructed by continuously granule composition, where each layer means the common knowledge and different knowledge lies in different layers. In the process of composition, coarser granules can be created and put into a higher abstract level regardless of some inessential information from several ones in the same layer. However, this is also the thought of reformation of EFG(Alg. 2). The Alg. 2 begins from the finest granules $F(L \cup \{l\}))$, and then extracts continuously their common knowledge $F(\{l\})$ which forms a new layer l in EFG. The algorithm will be ended if there is no different knowledge in $F(L)$, i.e., $|L| = 1$.

Algorithm 1. Inference algorithm in EFG

```
Input  : An EFG $G = (E, N, B, \varphi, \alpha, \beta)$.
Output: A new EFG $G' = (E, N', B', \varphi, \alpha, \beta)$.
$F(L) = \{E\}; F(DL) = \{g(n)|n \in N \text{ is a leaf}\}; B' = \emptyset; N' = \{n|n \in N \text{ is a leaf}\}$;
while $F(L) \neq \emptyset$ and $CL \neq \emptyset$ do
    $F(L) = F(L \cup \{l\}); CL = CL - \{l\}$; //Select $l$ from $CL$ to part $F(L)$ ;
    for $\forall g(n) \in F(L)$ do
        If $\exists g(n') \in F(DL)$ and $g(n) \subseteq g(n')$ then
            $F(L) = F(L) - \{g(n)\}; N' = N' \cup \{n\}; B' = B' \cup \{(n, n')\}$;
    end
end
if $F(L) \neq \emptyset$ and $CL = \emptyset$ then
    //In this case, there exists inconsistent path in EFG ;
    for $\forall g(n) \in F(L)$ do
        If $\exists g(n') \in F(DL)$ and $g(n) \cap g(n') \neq \emptyset$ then $B' = B' \cup \{(n, n')\}$;
        $N' = N' \cup \{n\}; F(L) = F(L) - \{g(n)\}$;
    end
end
```

Algorithm 2. Reformation algorithm in EFG

```
Input  : An EFG $G = (E, N, B, \varphi, \alpha, \beta)$.
Output: A reformed EFG $G' = (E, N', B', \varphi, \alpha, \beta)$.
$B' = B; N' = N; F(CL) = \{g(n)|n \in CL\}$ ;
while $|CL| \neq 1$ do
    $l = first(CL); CL = CL - \{l\}$; // i.e., $CL = \{l\} \cup L$;
    for $\forall g(n \wedge n') \in F(CL)$ do
        if $n \in l$ then
            $F(CL) = F(CL) - \{g(n \wedge n')\}; F(CL) = F(CL) \cup \{g(n)\} \cup \{g(n')\}$ ;
            $N' = N' \cup \{n\}; B' = B' \cup \{(n, n')\}$ ;
            for $\forall (n'', n \wedge n') \in B'$ do $B' = B' \cup \{(n'', n)\}$ ;
        end
    end
end
```

Since the number of granules in each layer is at most $|E|$, the cost of granule decomposition is $|E|^2$, and Alg. 1 takes less $|CL|$ times iteration before it stops. Thus, the time complexity of Alg. 1 is $O(|E|^2|CL|)$. Likewise, the complexity of Alg. 2 is $O(|E|^2|CL|)$.

If the Alg. 1 firstly is performed on an EFG, and then the Alg. 2 is employed, a new EFG can be obtained. Meanwhile, the new EFG is a reduction of the original one, because some redundant information(i.e., node) will not be considered in the Alg. 1(*If* statement). However, if more strict constraints are imposed on statement '$F(L)$=$F(L \cup l)$', the minimal reduct of the EFG could be achieved.

Example 4. **(cont.)** Let G be an EFG in Fig. 1, the inferred EFG is given in the left of Fig. 2 after the Alg. 1 is performed. As a result of Alg. 2 being carried out on the left EFG, the right is one of the reducts of the original EFG. □

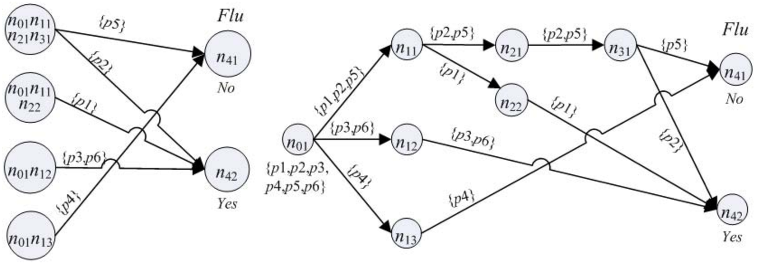


Fig. 2. A inferred EFG(left) and a reduct of the EFG(right)

5 Conclusion

An EFG is a graph model of decision algorithm, in which each path from the root to a leaf represents a rule. In virtue of its well-structured representation, EFG has a close relationship with GrC. As one of purposes of this paper, a interpretation of EFG using GrC is given. In GrC model of EFG, each node, branch or path represents a granule, and the meaning of the granule is the flowthrough of its corresponding path. Moreover, the granules, whose corresponding path starts from the root of EFG, form a hierarchy. In this hierarchy, the granules with respect to the same length path are arranged in the same layer and are getting finer along with the depth of the hierarchy.

Thanks to the GrC model of EFG, which is a partition one, the transformation among granules in different layers can be freely achieved by granule composition and decomposition without loss of any knowledge. As a result of the granule composition and decomposition, inference and reformation in EFG, which is another objective of this paper, can be easily implemented and the corresponding algorithms are also presented. Furthermore, if the inference and reformation operations are successionally performed on an EFG, one of the reductions of the original one can be yielded.

Acknowledgment

The authors would like to thank the anonymous referees for their valuable and constructive remarks. This work is supported by the National NSF of China (60473003), NSFC Major Research Program(60496321), Ministry of Education Program for New Century Excellent Talents in University and Doctor Point Founds of Educational Department(20050183065).

References

1. Butz, C.J., Yan, W., Yang, B.: The Computational Complexity of Inference using Rough Set Flow Graphs. In: [15], pp. 335–344 (2005)
2. Butz, C.J., Yan, W., Yang, B.: An Efficient Algorithm for Inference in Rough Set Flow Graphs. Transaction on Rough Sets. V 5, 102–122 (2006)
3. Czyzewski,A., Szczerba,M., Kostek,B.: Musical Metadata Retrieval with Flow Graphs. In: [18], pp. 691–698 (2004)
4. Kostek,B., Czyzewski, A.: Processing of Musical Metadata Employing Pawlak's Flow Graphs. In: [14], pp. 279–298 (2004)
5. Lin, T.Y.: Granular Computing on Binary Relations I: Data Mining and Neighborhood Systems. In: Skoworn, A., Polkowski, L. (eds.) Rough Sets In Knowledge Discovery, pp. 107–121. Springer, Heidelberg (1998)
6. Lin, T.Y., Yin, P.: Heuristically Fast Finding of the Shortest Reducts. In: [18], pp. 465–470 (2004)
7. Pawlak, Z.: Decision algorithms, Bayes Theorem and Flow Graphs, In: Proceeding of the 6th International Conference on Neural Networks & Soft Computing (2002)
8. Pawlak, Z.: Flow graphs and decision algorithms. In: [19], pp. 1–11(2003)
9. Pawlak, Z.: Decision Networks. In: [18], pp. 1–7 (2004)
10. Pawlak, Z.: Some Issues on Rough Sets. In: [14], pp. 1–58 (2004)
11. Pawlak, Z.: Decisions rules and flow networks. European Journal of Operational Research 154, 184–190 (2004)
12. Pawlak, Z.: Rough Sets and Flow Graphs. In: [15], pp. 1–11 (2005)
13. Pawlak, Z.: Flow Graphs and Data Mining. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III, pp. 1–58. Springer, Heidelberg (2005)
14. Peters, J.F., Skowron, A. (eds.): Transactions on Rough Sets I. Springer, Berlin (2004)
15. Ślęzak, D., et al. (ed.): Rough Sets, Fuzzy Sets, Data Mining and Granular Computing. Springer, Berlin (2005)
16. Sun, J., Liu, H., Zhang, H.: An Extension of Pawlak's Flow Graphs. In: Proceedings of the 1st International Conference on Rough Sets and Knowledge Technology, pp. 191–199 (2006)
17. Sun, J., Liu, H., Qi, C., Zhang, H.: An Interpretation of Flow Graphs by Granular Computing, In: Greco, S., Hata, Y., Hirano, S., et al. (eds.): Rough Sets and Current Trends in Computing, pp. 448–457 (2006)
18. Tsumoto, S., Słowiński, R., Komorowski, J. (eds.): RSCTC 2004. LNCS (LNAI), vol. 3066. Springer, Heidelberg (2004)
19. Wang, G.Y., et al. (ed.): Rough Sets,Fuzzy Sets,Data Mining and Granular Computing. FSRSGrC 2005. Springer, Heidelberg (2003)
20. Yao, Y.Y.: A partition model of granular computing. In: [14], pp. 232–253 (2004)
21. Yao, Y.Y.: Perspectives of Granular Computing. In: Proceedings of 2005 IEEE International Conference on Granular Computing, vol. 1, pp. 85–90 (2005)
22. Yao, Y.Y., Zhong, N.: Granular computing using information tables. In: Lin, T.Y., Yao, Y.Y., Zadeh, L.A. (eds.) Data Mining, Rough Sets and Granular Computing, pp. 102–124. Physica-Verlag, Heidelberg (2002)
23. Zhang, L., Zhang, B.: The quotient space theory of problem solving, In: [19], pp. 11–15 (2003)

On Granular Rough Computing with Missing Values

Lech Polkowski[1,2] and Piotr Artiemjew[2]

[1] Polish–Japanese Institute of Information Technology
Koszykowa 86, 02008 Warszawa, Poland
[2] Department of Mathematics and Computer Science
University of Warmia and Mazury, Olsztyn, Poland
polkow@pjwstk.edu.pl,artem@matman.uwm.edu.pl

Abstract. Granular Computing as a paradigm in Approximate Reasoning is concerned with granulation of available knowledge into granules that consists of entities similar in information content with respect to a chosen measure and with computing on such granules. Thus, operators acting on entities in a considered universe should factor through granular structures giving values similar to values of same operators in non–granular environment. Within rough set theory, proposed 25 years ago by Zdzisław Pawlak and developed thence by many authors, granulation is also a vital area of research. The first author developed a calculus with granules as well as a granulation technique based on similarity measures called rough inclusions along with a hypothesis that granules induced in data set universe of objects should lead to new objects representing them, and such granular counterparts should preserve information content in data. In this work, this hypothesis is tested with missing values in data and results confirm the hypothesis in this context.

Keywords: rough sets, decision systems, missing values, granules of knowledge, rough inclusions, granular decision systems.

1 Rough Computing

Rough sets are centered about the notion of *indiscernibility*[7]: entities with same description are regarded as identical. In practical terms, when knowledge is encoded in an *information system* (U, A) where U is a set of *entities* and A is a set of *attributes*, with each $a : U \to V_a$ a mapping on U into a value set, indiscernibility is given as an equivalence $ind(a) = \{(u, v) : u, v \in U, a(u) = a(v)\}$ for each $a \in A$, with extensions of the form $ind(B) = \bigcap_{a \in B} ind(a)$ for any $B \subseteq A$.

Rough computing is usually performed with *descriptors* of the form $(a = v)$, $v \in V_a$, interpreted as sets $[(a = v)] = \{u \in U : a(u) = v\}$; descriptors extend to *descriptor formulas* that form the smallest set containing all descriptors and closed on the action of propositional connectives $\vee, \wedge, \neg, \Rightarrow$; descriptor formulas are interpreted via identities $[\bigwedge_i (a_i = v_i)]$=$\bigcap_i [(a_i = v_i)]$, $[\bigvee_i (a_i = v_i)]$=$\bigcup_i [(a_i =$

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 271–279, 2007.

$v_i)]$, $[\neg(a = v)]$=$U \setminus [(a = v)]$. *Decision systems* are information systems of the form $(U, A \cup \{d\})$, where d, the *decision*, is an attribute not in A; relations between the *conditional knowledge* (U, A) and the *world knowledge* (U, d) are expressed by means of *decision rules* of the form $\bigwedge_i (a_i = v_i) \Rightarrow (d = v)$; a set of decision rules is a *classifier*; its aim is to recognize decision classes of new entities on the basis of their conditional values.

2 Missing Values

An information/decision system is *incomplete* in case some values of conditional attributes from A are not known; some authors, e.g., Grzymala–Busse [2] make distinction between values that are *lost* (denoted ?), i.e., they were not recorded or were destroyed in spite of their importance for classification, and values that are *missing* (denoted $*$) as those values that are not essential for classification. Here, we regard all lacking values as missing without making any distinction among them denoting all of them with $*$. Analysis of systems with missing values requires a decision on how to treat missing values; Grzymala–Busse in his work [2], analyzes nine such methods known in the literature, among them, *1. most common attribute value*, *2. concept–restricted most common attribute value*, (...), *4. assigning all possible values to the missing location*, (...), *9. treating the unknown value as a new valid value.* Results of tests presented in [2] indicate that methods *4,9* perform very well among all nine methods. For this reason we adopt these methods in this work for the treatment of missing values and they are combined in our work with a modified method *1*: the missing value is defined as the most frequent value in the granule closest to the object with the missing value with respect to a chosen rough inclusion.

Analysis of decision systems with missing data in existing rough set literature relies on an appropriate treatment of indiscernibility: one has to reflect in this relation the fact that some values acquire a distinct character and must be treated separately; in case of missing or lost values, the relation of indiscernibility is usually replaced with a new relation called a *characteristic relation.* Examples of such characteristic functions are given in, e.g., Grzymala–Busse [3]: the function ρ is introduced, with $\rho(u, a) = v$ meaning that the attribute a takes on u the value v. Semantics of descriptors is changed, viz., the meaning $[(a = v)]$ has as elements all u such that $\rho(u, a) = v$, in case $\rho(u, a) =?$ the entity u is not included into $[(a = v)]$, and in case $\rho(u, a) = *$, the entity u is included into $[(a = v)]$ for all values $v \neq *, ?$. Then the characteristic relation is $R(B) = \{(u, v) : \forall.a \in B.\rho(u, a) =? \Rightarrow (\rho(u, a) = \rho(v, a) \vee \rho(u, a) = * \vee \rho(v, a) = *)\}$, where $B \subseteq A$. Classes of the relation $R(B)$ are then used in defining approximations to decision classes from which certain and possible rules are induced, see [3]. Specializations of the characteristic relation $R(B)$ were defined in Stefanowski–Tsoukias [18] (in case of only lost values) and in Kryszkiewicz [4](in case of only don't care missing values). An analysis of the problem of missing values along with algorithms *IApriori Certain* and *IAprioriPossible* for certain and possible rule generation was given in [5].

3 Granules of Knowledge and Granular Information/Decision Systems

Granulation of knowledge is a topic studied recently to much extent within rough set theory, see, e.g., [14],[15]. We describe briefly a method for inducing granules [10], [11] which consists in selecting a rough inclusion μ (see op.cit.), and $r \in [0, 1]$.

3.1 Rough Inclusions

Generally they are predicates of the form $\mu(u, v, r)$, where $u, v \in U$ satisfying conditions, 1. $\mu(u, u, 1)$;2. if $\mu(u, v, 1)$ then for each $w \in U$, from $\mu(w, u, r)$ it follows $\mu(w, v, r)$; 3. if $\mu(u, v, r)$ and $s < r$ then $\mu(u, v, s)$. For an analysis of various methods for inducing rough inclusions see, e.g., [10], [11]. In this work we will use exclusively the rough inclusion $\mu_L(u, v, r)$ satisfied if and only if $\frac{|IND(u,v)|}{|A|} \geq r$, where $IND(u, v) = \{a \in A : a(u) = a(v)\}$, induced by the Łukasiewicz implication (see, e.g., [10],[11]).

3.2 On Granule Formation

For a rough inclusion μ, $u \in U$, and $r \in [0, 1]$, the granule $g_\mu(u, r)$ is defined as the class $Cls\{v : \mu(v, u, r)\}$, where Cls is the class forming functor of mereology, see, e.g., [10],[11]; for the purpose of this work, one may assume that $g_\mu(u, r)$ is the list or the set of all v such that $\mu(v, u, r)$. In this work, granules are formed only by means of μ_L. In plain words, the granule $g_{\mu_L}(u, r)$ consists of all $v \in U$ with the property that $|IND(v, u)| \geq r \cdot |A|$, i.e., v, u have identical values of at least $r \cdot 100$ percent of attributes in A.

3.3 Granular Information/Decision Systems

The idea of a granular decision system was posed in [10]; for a given information system (U, A), a rough inclusion μ, and $r \in [0, 1]$, the new universe $U^G_{r,\mu}$ is given, whose elements are granules of the radius r about objects $u \in U$. We apply a strategy $\mathcal{G}$ to choose a covering $Cov^G_{r,\mu}$ of the universe U by granules from $U^G_{r,\mu}$.

We apply a strategy $\mathcal{S}$ in order to assign the value $a^*(g)$ of each attribute $a \in A$ to each granule $g \in Cov^G_{r,\mu}$: $a^*(g) = \mathcal{S}(\{a(u) : u \in g\})$. The granular counterpart to the information system (U, A) is a tuple $(U^G_{r,\mu}, \mathcal{G}, \mathcal{S}, \{a* : a \in A\})$; analogously, we define granular counterparts to decision systems by adding the factored decision $d*$. The heuristic principle that *objects, similar with respect to conditional attributes in the set A, should also reveal similar (i.e., close) decision values, and therefore, granular counterparts to decision systems should lead to classifiers satisfactorily close in quality to those induced from original decision systems*, was stated in [10], and borne out by simple hand examples. The hypothesis has been confirmed in [12] and in this work we apply this hypothesis to the problem of missing values.

4 An Approach to Missing Values in This Work

We will use the symbol $*$ commonly used for denoting the missing value; we will use two methods *4, 9* for treating $*$, i.e, either $*$ is a *don't care* symbol meaning that any value of the respective attribute can be substituted for $*$,thus $* = v$ for each value v of the attribute, or $*$ is a new value on its own, i.e., if $* = v$ then v can be only $*$.

Our procedure for treating missing values is based on the granular structure $(U^G_{r,\mu}, \mathcal{G}, \mathcal{S}, \{a* : a \in A\})$; the strategy $\mathcal{S}$ is the majority voting, i.e., for each attribute a, the value $a^*(g)$ is the most frequent of values in $\{a(u) : u \in g\}$, with ties broken randomly. The strategy $\mathcal{G}$ consists in random selection of granules for a covering.

For an object u with the value of $*$ at an attribute a, and a granule $g = g(v, r) \in U^G_{r,\mu}$, the question whether u is included in g is resolved according to the adopted strategy of treating $*$: in case $* = don't\ care$, the value of $*$ is regarded as identical with any value of a hence $|IND(u, v)|$ is automatically increased by 1, which increases the granule; in case $* = *$, the granule size is decreased. Assuming that $*$ is sparse in data, majority voting on g would produce values of a^* distinct from $*$ in most cases; nevertheless the value of $*$ may appear in new objects g^*, and then in the process of classification, such value is repaired by means of the granule closest to g^* with respect to the rough inclusion μ_L, in accordance with the chosen method for treating $*$.

In plain words, objects with missing values are in a sense absorbed by close to them granules and missing values are replaced with most frequent values in objects collected in the granule; in this way the method *4* or *9* in [3] is combined with the idea of the most frequent value *1*, in a novel way.

We have thus four possible strategies:

- Strategy A: in building granules $*$=*don't care*, in repairing values of $*$, $*$=*don't care*;
- Strategy B: in building granules $*$=*don't care*, in repairing values of $*$, $* = *$;
- Strategy C: in building granules $* = *$, in repairing values of $*$, $*$=*don't care*;
- Strategy D: in building granules $* = *$, in repairing values of $*$, $* = *$.

As data set used in experiments, Pima Indians diabetes data set [19] has been used. We first show results for this data set in granular and non–granular cases without missing values in Table 1, see [12] for a discussion of this method in more detail; then a randomly chosen collection of 10 percent of attribute values in the data set are replaced with $*$ values. Results of granular treatment in case of Strategies A,B,C,D are reported in Tables 2,3,4,5. As algorithm for rule induction, the exhaustive algorithm of the RSES system [16] has been selected, see, e.g., [1], [17], where the ideas implemented in the RSES package are discussed. 10–fold cross validation (CV–10) has been used to validate results of the experiment.

Table 1. 10-fold CV; Pima; exhaustive algorithm. r=radius,macc=mean accuracy,mcov=mean coverage,mrules=mean rule number, mtrn=mean size of granular training set

r	*macc*	*mcov*	*mrules*	*mtrn*
nil	0.6864	0.9987	7629.2	692.0
0.0	0.0	0.0	0.0	1.0
0.125	0.0618	0.0895	5.9	22.5
0.250	0.6627	0.9948	450.1	120.6
0.375	0.6536	0.9987	3593.6	358.7
0.500	0.6645	1.0	6517.7	579.4
0.625	0.6877	0.9987	7583.6	683.1
0.750	0.6864	0.9987	7629.2	692
0.875	0.6864	0.9987	7629.2	692.0

Table 2. Strategy A for missing values. 10-fold CV; Pima; exhaustive algorithm. r=radius, macc=mean accuracy, mcov=mean coverage, mrules=mean rule number, mtrn=mean size of granular training set

r	*macc*	*mcov*	*mrules*	*mtrn*
nil	0.6864	0.9987	7629.2	692.0
0.0	0.0	0.0	0.0	1.0
0.125	0.0	0.0	0.0	1.7
0.250	0.0	0.0	0.0	4.7
0.375	0.0	0.0	0.0	21.5
0.500	0.3179	0.4777	115.8	64.7
0.625	0.6692	0.9987	1654.7	220.2
0.750	0.6697	1.0	5519.3	527.0
0.875	0.6678	0.9987	7078.8	663.8

Table 3. Strategy B for missing values. 10-fold CV; Pima; exhaustive algorithm. r=radius,macc=mean accuracy, mcov=mean coverage, mrules=mean rule number, mtrn=mean size of granular training set

r	*macc*	*mcov*	*mrules*	*mtrn*
nil	0.6864	0.9987	7629.2	692.0
0.0	0.0	0.0	0.0	1.0
0.125	0.0	0.0	0.0	1.9
0.250	0.0	0.0	0.0	6.1
0.375	0.0	0.0	0.0	13.7
0.500	0.5772	0.8883	210.7	68.1
0.625	0.6467	0.9987	1785.8	229.4
0.750	0.6587	0.9987	5350.4	508.5
0.875	0.6547	0.9987	6982.7	663.4

Table 4. Strategy C for missing values. 10-fold CV; Pima; exhaustive algorithm. r=radius,macc=mean accuracy, mcov=mean coverage, mrules=mean rule number, mtrn=mean size of granular training set

r	*macc*	*mcov*	*mrules*	*mtrn*
nil	0.6864	0.9987	7629.2	692.0
0.0	0.0	0.0	0.0	1.0
0.125	0.0	0.0	0.0	21.2
0.250	0.6297	0.9948	388.9	116.9
0.375	0.6556	0.9974	3328.5	356.5
0.500	0.6433	1.0	6396.7	587.2
0.625	0.6621	1.0	7213.2	681.9
0.750	0.6640	0.9987	7306.3	691.9
0.875	0.6615	0.9987	7232.1	692.0

Table 5. Strategy D for missing values. 10-fold CV; Pima; exhaustive algorithm. r=radius, macc=mean accuracy, mcov=mean coverage, mrules=mean rule number, mtrn=mean size of granular training set

r	*macc*	*mcov*	*mrules*	*mtrn*
nil	0.6864	0.9987	7629.2	692.0
0.0	0.0	0.0	0.0	1.0
0.125	0.1471	0.1750	12.0	17.3
0.250	0.6572	0.9974	382.1	114.9
0.375	0.6491	0.9974	3400.3	355.0
0.500	0.6370	0.9974	6300.2	588.7
0.625	0.6747	0.9987	7181.2	682.3
0.750	0.6724	1.0	7231.3	691.9
0.875	0.6618	1.0	7253.6	692.0

5 Case of Real Data with Missing Values

We include results of tests with Breast cancer data set [19] that contains missing values. We show in Tables 6, 7, 8, 9 results for intermediate values of radii of granulation for strategies A,B,C,D and exhaustive algorithm of RSES [16]. For comparison, results on error in classification by the endowed system LERS from [2] for approaches similar to our strategies A and D (methods 4 and 9, resp., in Tables 2 and 3 in [2]) in which $*$ is either always $*$ (method 9) or $*$ is always *don't care* (method 4) are recalled in Tables 6 and 9. We have applied here the 1-train–and–9 test, i.e., the data set is split randomly into 10 equal parts and training set is one part whereas the rules are tested on each of remaining 9 parts separately and results are averaged.

5.1 Conclusions on Test Results

In case of perturbed Pima Indians diabetes data set, Strategy A attains accuracy value better than 97 percent and coverage value greater or equal to values in

Table 6. Breast cancer data set with missing values. Strategy A: r=granule radius, mtrn= mean granular training sample size, macc=mean accuracy, mcov=mean covering, gb=LERS method 4,[2]

r	*mtrn*	*macc*	*mcov*	*gb*
0.555556	9	0.7640	1.0	0.7148
0.666667	14	0.7637	1.0	
0.777778	17	0.7129	1.0	
0.888889	25	0.7484	1.0	

Table 7. Breast cancer data set with missing values. Strategy B: r=granule radius, mtrn= mean granular training sample size, macc=mean accuracy, mcov=mean covering

r	*mtrn*	*macc*	*mcov*
0.555556	7	0.0	0.0
0.666667	13	0.7290	1.0
0.777778	16	0.7366	1.0
0.888889	25	0.7520	1.0

Table 8. Breast cancer data set with missing values. Strategy C: r=granule radius, mtrn= mean granular training sample size, macc=mean accuracy, mcov=mean covering

r	*mtrn*	*macc*	*mcov*
0.555556	8	0.7132	1.0
0.666667	14	0.6247	1.0
0.777778	17	0.7328	1.0
0.888889	25	0.7484	1.0

Table 9. Breast cancer data set with missing values. Strategy D: r=granule radius, mtrn= mean granular training sample size, macc=mean accuracy, mcov=mean covering, gb=LERS method 9,[2]

r	*mtrn*	*macc*	*mcov*	*gb*
0.555556	9	0.7057	1.0	0.6748
0.666667	16	0.7640	1.0	
0.777778	17	0.6824	1.0	
0.888889	25	0.7520	1.0	

non–perturbed case from the radius of .625 on. With Strategy B, accuracy is within 94 percent and coverage not smaller than values in non–perturbed case from the radius of .625 on. Strategy C yields accuracy within 96.3 percent of accuracy in non–perturbed case from the radius of .625, and within 95 percent from the radius of .250; coverage is within 99.79 percent from the radius of .250. Strategy D gives results slightly better than C with the same radii. Results for C and D are better than results for A or B.

Table 10. Average number of $*$ values in granular systems. 10-fold CV; Pima; exhaustive algorithm. r=radius,mA=mean value for A, mB=mean value for B , mC=mean value for C, mD=mean value for D

r	mA	mB	mC	mD
0.375	0.0	0.0	135	132
0.500	0.0	0.0	412	412
0.625	3	4	538	539
0.750	167	167	554	554
0.875	435	435	554	554

We conclude that essential for results of classification is the strategy of treating the missing value of $*$ as $* = *$ in both strategies C and D; the repairing strategy has almost no effect: C and D differ with respect to this strategy but results for accuracy and coverage in cases C and D differ very slightly.

Let us notice that strategies C and D cope with a larger number of $*$ values to be repaired. Table 10 shows this.

In experiments with Breast cancer data set with missing values, best results are obtained with "pure" strategies A and D; strategy A gives accuracy of .7637 at $r = .(6)$ and strategy D gives accuracy of .7640 at $r = .(6)$, "mixed" strategies give best results at higher value of radius of .(7): .7474 in case of C and .7520 in case of B.

6 Conclusions

The method proposed in this work for treatment of missing values that combines either of two approaches, viz., $*$*don't care* or $* = *$ with the idea of absorbing objects with missing values into granules consisting of objects close to them to a degree specified by radii of granules, followed by the idea of replacing the missing value with the most frequent value over the granule, has proved very effective in the classification problem of data with missing values.

In the stage of repairing the missing value, strategies C and D proved most effective. Essential for results of classification is the strategy of treating the missing value of $*$ as $* = *$ in building granules as witnessed by cases of strategies C and D; strategies A and B give comparable results between them, implying that when the strategy $*=$*don't care* is used in building granules, then the choice of a repairing strategy has no practical impact.

Further research will be focused on more refined ways of granule selection, development of a granular algorithm for rule induction, and analysis of large real data with missing values.

References

1. Bazan, J.G.: A comparison of dynamic and non–dynamic rough set methods for extracting laws from decision tables. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery 1, pp. 321–365. Physica Verlag, Heidelberg (1998)

2. Grzymala–Busse, J.W., Hu, M.: A comparison of several approaches to missing attribute values in data mining. In: RSCTC 2000. LNCS (LNAI), vol. 2005, pp. 378–385. Springer, Berlin (2000)
3. Grzymala–Busse, J.W.: Data with missing attribute values: Generalization of rule indiscernibility relation and rule induction. In: Transactions on Rough Sets I, pp. 78–95. Springer, Berlin (2004)
4. Kryszkiewicz, M.: Rules in incomplete information systems. Information Sciences 113, 271–292 (1999)
5. Kryszkiewicz, M., Rybiński, H.: Data mining in incomplete information systems from rough set perspective. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.) Rough Set Methods and Applications, pp. 568–580. Physica Verlag, Heidelberg (2000)
6. Leśniewski, S.: On the foundations of set theory vol. 2, pp. 7–52 (1982)
7. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)
8. Polkowski, L.: Rough Sets. Mathematical Foundations. Physica Verlag, Heidelberg (2002)
9. Polkowski, L.: oward rough set foundations. Mereological approach (a plenary lecture). In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 8–25. Springer, Heidelberg (2004)
10. Polkowski, L.: Formal granular calculi based on rough inclusions (a feature talk) In: [14], pp. 57–62
11. Polkowski, L.: A model of granular computing with applications (a feature talk), In: [15], pp. 9–16
12. Polkowski, L., Artiemjew, P.: On granular rough computing: Factoring classifiers through granulated decision systems. In: these Proceedings
13. Polkowski, L., Skowron, A.: Rough mereology: a new paradigm for approximate reasoning. International Journal of Approximate Reasoning 15(4), 333–365 (1997)
14. Proceedings of IEEE 2005 Conference on Granular Computing. In: GrC05, Beijing, China, July 2005, IEEE Press, New York (2005)
15. Proceedings of IEEE 2006 Conference on Granular Computing. In: GrC06, Atlanta, USA, May 2006, IEEE Press, New York (2006)
16. Skowron, A., et al.: RSES: A system for data analysis, available at `http://logic.mimuw.edu.plrses/`
17. Nguyen, S.H.: Regularity analysis and its applications in Data Mining. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.) Rough Set Methods and Applications, pp. 289–378. Physica Verlag, Heidelberg (2000)
18. Stefanowski, J., Tsoukias, A.: Incomplete information tables and rough classification. Computational Intelligence 17, 545–566 (2001)
19. `http://www.ics.uci.edu.mlearn/databases/`

On Granular Rough Computing: Factoring Classifiers Through Granulated Decision Systems

Lech Polkowski[1,2] and Piotr Artiemjew[2]

[1] Polish–Japanese Institute of Information Technology
Koszykowa 86, 02008 Warszawa, Poland
[2] Department of Mathematics and Computer Science
University of Warmia and Mazury, Olsztyn,Poland
polkow@pjwstk.edu.pl,artem@matman.uwm.edu.pl

Abstract. The paradigm of Granular Computing has quite recently emerged as an area of research on its own; in particular, it is pursued within rough set theory initiated by Zdzisław Pawlak. Granules of knowledge consist of entities with a similar in a sense information content. An idea of a granular counterpart to a decision/information system has been put forth, along with its consequence in the form of the hypothesis that various operators, aimed at dealing with information, should factorize sufficiently faithfully through granular structures [7], [8]. Most important such operators are algorithms for inducing classifiers. We show results of testing few well-known algorithms for classifier induction on well–used data sets from Irvine Repository in order to verify the hypothesis. The results confirm the hypothesis in case of selected representative algorithms and open a new prospective area of research.

Keywords: rough inclusion, similarity, granulation of knowledge, granular systems and classifiers.

1 Rough Computing

Knowledge is represented as a pair (U, A), called an *information system* [4], where U is a set of *objects*, and A is a collection of *attributes*, each $a \in A$ construed as a mapping $a : U \to V_a$ from U into the *value set* V_a. The collection $IND = \{ind(a) : a \in A\}$ of *a–indiscernibility relations*, where $ind(a) = \{(u, v) : u, v \in U, a(u) = a(v)\}$ for $a \in A$, can be restricted to any set $B \subseteq A$, yielding the *B–indiscernibility relation* $ind(B) = \bigcap_{a \in B} ind(a)$. A *concept* is any subset of the set U. By a *proper rough entity*, we mean any entity e constructed from objects in U and relations in R such that its action $e \cdot u$ on each object $u \in U$ satisfies the condition: if $(u, v) \in r$ then $e \cdot u = e \cdot v$ for each $r \in R$; in particular, proper rough concepts are called *exact*, improper rough concepts are called *rough.* A particular case of an information system is a *decision system*, i.e., a pair $(U, A \cup \{d\})$ in which d is a singled out attribute called the *decision.*

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 280–289, 2007.

Basic primitives in any reasoning based on rough set theory, are *descriptors*, see, e.g., [4], of the form $(a = v)$, with semantics of the form $[(a = v)] = \{u \in U : a(u) = v\}$, extended to the set of formulae by means of sentential connectives, with appropriately extended semantics. In order to relate the *conditional* knowledge (U, IND) to the *world knowledge* $(U, \{ind(d)\})$, *decision rules* are in use; a decision rule is an implication of the form,

$$\bigwedge_{a \in A} (a = v_a) \Rightarrow (d = w). \tag{1}$$

A *classifier* is a set of decision rules.

2 Rough Mereology: Rough Inclusions

We outline it here as a basis for discussion of granules in the wake of [7], [8]. Rough Mereology is concerned with the theory of the predicate of Rough Inclusion.

2.1 Rough Inclusions

A *rough inclusion* $\mu_\pi(x, y, r)$, where x, y are individual objects, $r \in [0, 1]$, does satisfy the following requirements, relative to a given part relation π on a set U of individual objects,see [6], [7], [8], [9],

$$\begin{gathered} 1.\ \mu_\pi(x, y, 1) \Leftrightarrow x\ ing_\pi\ y; \\ 2.\ \mu_\pi(x, y, 1) \Rightarrow [\mu_\pi(z, x, r) \Rightarrow \mu_\pi(z, y, r)]; \\ 3.\ \mu_\pi(x, y, r) \wedge s < r \Rightarrow \mu_\pi(x, y, s). \end{gathered} \tag{2}$$

Those requirements seem to be intuitively clear: 1. demands that the predicate μ_π is an extension to the relation ing_π of the underlying system of Mereology; 2. does express monotonicity of μ_π, and 3. assures the reading: "to degree at least r". We use here only one rough inclusion, albeit a fundamental one, viz., see [6],[7] for its derivation,

$$\mu_L(u, v, r) \Leftrightarrow \frac{|IND(u, v)|}{|A|} \geq r, \tag{3}$$

where $IND(u, v) = \{a \in A : a(u) = a(v)\}$.

3 Granules

A granule $g_\mu(u, r)$ about $u \in U$ of the radius r, relative to μ, is defined by letting,

$$g_\mu(u, r) \text{ is } ClsF(u, r), \tag{4}$$

where the property $F(u, r)$ is satisfied with an object v if and only if $\mu(v, u, r)$ holds, and Cls is the class operator, see, e.g., [6]. Practically, in case of μ_L, the granule $g(u, r)$ collects all $v \in U$ such that $|IND(v, u)| \geq r \cdot |A|$.

For a given granulation radius r, we form the collection $U^G_{r,\mu} = \{g_\mu(u, r)\}$.

3.1 Granular Decision Systems

The idea of a granular decision system was posed in [7]; for a given information system (U, A), a rough inclusion μ, and $r \in [0, 1]$, the new universe $U^G_{r,\mu}$ is given. We apply a strategy $\mathcal{G}$ to choose a covering $Cov^G_{r,\mu}$ of the universe U by granules from $U^G_{r,\mu}$.

We apply a strategy $\mathcal{S}$ in order to assign the value $a^*(g)$ of each attribute $a \in A$ to each granule $g \in Cov^G_{r,\mu}$: $a^*(g) = \mathcal{S}(\{a(u) : u \in g\})$. The granular counterpart to the information system (U, A) is a tuple $(U^G_{r,\mu}, \mathcal{G}, \mathcal{S}, \{a* : a \in A\})$; analogously, we define granular counterparts to decision systems by adding the factored decision $d*$. The heuristic principle that *objects, similar with respect to conditional attributes in the set A, should also reveal similar (i.e., close) decision values, and therefore, granular counterparts to decision systems should lead to classifiers satisfactorily close in quality to those induced from original decision systems*, was stated in [7], and borne out by simple hand examples. In this work we verify this hypothesis with real data sets.

4 Classifiers: Rough Set Methods

Classifiers are evaluated by *error* which is the ratio of the number of correctly classified objects to the number of recognized test objects (called also *total accuracy*) and *total coverage*, $\frac{rec}{test}$, where *rec* is the number of recognized test cases and *test* is the number of test cases.

We test LEM2 algorithm due to Grzymala–Busse, see, e.g., [2] and covering as well as exhaustive algorithm in RSES package [12], see [1], [13], [16],[17].

4.1 On the Approach in This Work

For $g(u, r)$ with r fixed and attribute $a \in A \cup \{d\}$, the factored value $a^*(g)$ is defined as $\mathcal{S}(\{a(u) : u \in g\})$ for a strategy $\mathcal{S}$, each granule g does produce a new object g^*, with attribute values $a(g^*) = a^*(g)$ for $a \in A$, possibly not in the data set universe U.

From the set $U^G_{r,\mu}$, see sect.3.1, of all granules of the form $g_\mu(u, r)$, by means of a strategy $\mathcal{G}$, we choose a covering $Cov^G_{r,\mu}$ of the universe U. Thus, a decision system D^*=$\{g^* : g \in Cov^G_{r,\mu}\}, A^* \cup \{d^*\})$ is formed, called the *granular counterpart relative to strategies* $\mathcal{G}, \mathcal{S}$ to the decision system $D = (U, A \cup \{d\})$; this new system is substantially smaller in size for intermediate values of r, hence, classifiers induced from it have correspondingly smaller number of rules.

As stated above, the hypothesis is that the granular counterpart D^* at sufficiently large granulation radii r preserves knowledge encoded in the decision system D to a satisfactory degree so given an algorithm $\mathcal{A}$ for rule induction, classifiers obtained from the training set *D(trn)* and its granular counterpart $D^*(trn)$ should agree with a small error on the test set *D(tst)*.

5 Experiments

In experiments with real data sets, we accept total accuracy and total coverage coefficients as quality measures in comparison of classifiers given in this work.

We make use of some well–known real life data sets often used in testing of classifiers. Due to shortage of space, we include only a very few results.

The following data sets have been used: Credit card application approval data set (Australian credit), see [14]; Pima Indians diabetes data set [14].

As representative and well–established algorithms for rule induction in public domain,we have selected

- the RSES exhaustive algorithm, see [12];
- the covering algorithm of RSES with p=.1[12];
- LEM2 algorithm, with p=.5, see [2], [12].

Table 1 shows a comparison of these algorithms on the data set Australian credit split into the training and test sets with the ratio 1:1.

Table 1. Comparison of algorithms on Australian credit data. 345 training objects, 345 test objects

algorithm	*accuracy*	*coverage*	*rule number*
$covering(p = .1)$	0.670	0.783	589
$covering(p = .5)$	0.670	0.783	589
$covering(p = 1.0)$	0.670	0.783	589
exhaustive	0.835	1.0	5149
$LEM2(p = .1)$	0.810	0.061	6
$LEM2(p = .5)$	0.906	0.368	39
$LEM2(p = 1.0)$	0.869	0.643	126

In rough set literature there are results of tests with other algorithms on Australian credit data set; we recall some best of them in Table 2 and we include also best granular cases from this work.

For any granule g and any attribute b in the set $A \cup d$ of attributes, the reduced attribute's $\overline{b}$ value at the granule g has been estimated by means of the majority voting strategy and ties have been resolved at random; majority voting is one of most popular strategies and was frequently applied within rough set theory, see, e.g., [13], [16].

We also use the simplest strategy for covering finding, i.e., we select coverings by ordering objects in the set U and choosing sequentially granules about them in order to obtain an irreducible covering; a random choice of granules is applied in sections in which this is specifically mentioned.

The only enhancement of the simple granulation is discussed in sect. 6 where the concept–dependent granules are considered; this approach yields even better classification results.

Table 2. Best results for Australian credit by some rough set based algorithms; in case $*$, reduction in object size is 40.6 percent, reduction in rule number is 43.6 percent; in case $**$, resp. 10.5, 5.9; in case $***$, resp., 3.6, 1.9

source	*method*	*accuracy*	*coverage*
$Bazan$[1]	$SNAPM(0.9)$	$error = 0.130$	–
$S.H.Nguyen$[13]	$simple.templates$	0.929	0.623
$S.H.Nguyen$[13]	$general.templates$	0.886	0.905
$S.H.Nguyen$[13]	$closest.simple.templates$	0.821	.1.0
$S.H.Nguyen$[13]	$closest.gen.templates$	0.855	1.0
$S.H.Nguyen$[13]	$tolerance.simple.templ.$	0.842	1.0
$S.H.Nguyen$[13]	$tolerance.gen.templ.$	0.875	1.0
$J.Wroblewski$[17]	$adaptive.classifier$	0.863	–
$this.work$	$granular^{*}.r = 0.642857$	0.867	1.0
$this.work$	$granular^{**}.r = 0.714826$	0.875	1.0
$this.work$	$granular^{***}.concept.dependent.r = 0.785714$	0.9970	0.9995

5.1 Train–and–Test at 1:1 Ratio for Australian Credit

We include here results for Australian credit. Table 3 shows size of training and test sets in non–granular and granular cases as well as results of classification versus radii of granulation. Table 4 shows absolute differences between non–granular case (r=nil) and granular cases as well as fraction of training and rule sets in granular cases against those in non–granular case.

With covering algorithm, accuracy is better or within error of 1 percent for all radii, coverage is better or within error of 4.5 percent from the radius of 0.214860 on where training set size reduction is 99 percent and reduction in rule set size is 98 percent.

With exhaustive algorithm, accuracy is within error of 10 percent from the radius of 0.285714 on, and it is better or within error of 4 percent from the radius of 0.5 where reduction in training set size is 85 percent and reduction in rule set size is 95 percent. The result of .875 at $r = .714$ is among the best at all (see Table 2). Coverage is better from $r = .214$ in the granular case, reduction in objects is 99 percent, reduction in rule size is almost 100 percent.

LEM2 gives accuracy better or within 2.6 percent error from the radius of 0.5 where training set size reduction is 85 percent and rule set size reduction is 96 percent. Coverage is better or within error of 7.3 percent from the radius of .571429 on where reduction in training set size is 69.6 percent and rule set size is reduced by 96 percent.

5.2 CV-10 with Pima

We have experimented with Pima Indians diabetes data set using 10–fold cross–validation and random choice of a covering for exhaustive and LEM2 algorithms. Results are in Tables 5, 6.

For exhaustive algorithm, accuracy in granular case is 95.4 percent of accuracy in non–granular case, from the radius of .25 with reduction in size of the training

Table 3. Australian credit dataset:r=granule radius,tst=test sample size,trn=training sample size,rulcov=number of rules with covering algorithm,rulex=number of rules with exhaustive algorithm, rullem=number of rules with LEM2,acov=total accuracy with covering algorithm,ccov=total coverage with covering algorithm,aex=total accuracy with exhaustive algorithm,cex=total coverage with exhaustive algorithm,alem=total accuracy with LEM2, clem=total coverage with LEM2

r	*tst*	*trn*	*rulcov*	*rulex*	*rullem*	*acov*	*ccov*	*aex*	*clex*	*alem*	*clem*
nil	345	345	571	5597	49	0.634	0.791	0.872	0.994	0.943	0.354
0.0	345	1	14	0	0	1.0	0.557	0.0	0.0	0.0	0.0
0.0714286	345	1	14	0	0	1.0	0.557	0.0	0.0	0.0	0.0
0.142857	345	2	16	0	1	1.0	0.557	0.0	0.0	1.0	0.383
0.214286	345	3	7	7	1	0.641	1.0	0.641	1.0	0.600	0.014
0.285714	345	4	10	10	1	0.812	1.0	0.812	1.0	0.0	0.0
0.357143	345	8	18	23	2	0.820	1.0	0.786	1.0	0.805	0.252
0.428571	345	20	29	96	2	0.779	0.826	0.791	1.0	0.913	0.301
0.5	345	51	88	293	2	0.825	0.843	0.838	1.0	0.719	0.093
0.571429	345	105	230	933	2	0.835	0.930	0.855	1.0	0.918	0.777
0.642857	345	205	427	3157	20	0.686	0.757	0.867	1.0	0.929	0.449
0.714286	345	309	536	5271	45	0.629	0.774	0.875	1.0	0.938	0.328
0.785714	345	340	569	5563	48	0.629	0.797	0.870	1.0	0.951	0.357
0.857143	345	340	570	5574	48	0.626	0.791	0.864	1.0	0.951	0.357
0.928571	345	342	570	5595	48	0.628	0.794	0.867	1.0	0.951	0.357
1.0	345	345	571	5597	49	0.634	0.791	0.872	0.994	0.943	0.354

Table 4. Australian credit dataset:comparison; r=granule radius,acerr= abs.total accuracy error with covering algorithm,ccerr= abs.total coverage error with covering algorithm,aexerr=abs.total accuracy error with exhaustive algorithm,cexerr=abs.total coverage error with exhaustive algorithm,alemerr=abs.total accuracy error with LEM2, clemerr=abs.total coverage error with LEM2, sper=training sample size as fraction of the original size,rper= max rule set size as fraction of the original size

r	*acerr*	*ccerr*	*aexerr*	*cexerr*	*alemerr*	*clemerr*	*sper*	*rper*
nil	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
0.0	0.366+	0.234	0.872	0.994	0.943	0.354	0.003	0.024
0.0714286	0.366+	0.234	0.872	0.994	0.943	0.354	0.003	0.024
0.142857	0.366+	0.234	0.872	0.994	0.057+	0.029+	0.0058	0.028
0.214286	0.007+	0.209+	0.231	0.006+	0.343	0.340	0.009	0.02
0.285714	0.178+	0.209+	0.06	0.006+	0.943	0.354	0.012	0.02
0.357143	0.186+	0.209+	0.086	0.006+	0.138	0.102	0.023	0.04
0.428571	0.145+	0.035+	0.081	0.006+	0.03	0.053	0.058	0.05
0.5	0.191+	0.052+	0.034	0.006+	0.224	0.261	0.148	0.154
0.571429	0.201+	0.139+	0.017	0.006+	0.025	0.423+	0.304	0.403
0.642857	0.052+	0.034	0.005	0.006+	0.014	0.095+	0.594	0.748
0.714286	0.005	0.017	0.003+	0.006+	0.005	0.026	0.896	0.942
0.785714	0.005	0.006+	0.002	0.006+	0.008+	0.003+	0.985	0.994
0.857143	0.008	0.0	0.008	0.006+	0.008+	0.003+	0.985	0.998
0.928571	0.006	0.003+	0.005	0.006+	0.008+	0.003+	0.991	0.999
1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0

Table 5. 10-fold CV; Pima; exhaustive algorithm. r=radius,macc=mean accuracy,mcov=mean coverage,mrules=mean rule number, mtrn=mean size of training set

r	*macc*	*mcov*	*mrules*	*mtrn*
nil	0.6864	0.9987	7629	692
0.125	0.0618	0.0895	5.9	22.5
0.250	0.6627	0.9948	450.1	120.6
0.375	0.6536	0.9987	3593.6	358.7
0.500	0.6645	1.0	6517.6	579.4
0.625	0.6877	0.9987	7583.6	683.1
0.750	0.6864	0.9987	7629.2	692
0.875	0.6864	0.9987	7629.2	692

Table 6. 10-fold CV; Pima; LEM2 algorithm. r=radius,macc=mean accuracy,mcov=mean coverage,mrules=mean rule number, mtrn=mean size of training set

r	macc	mcov	mrules	mtrn
nil	0.7054	0.1644	227.0	692
0.125	0.900	0.2172	1.0	22.5
0.250	0.7001	0.1250	12.0	120.6
0.375	0.6884	0.2935	74.7	358.7
0.500	0.7334	0.1856	176.1	579.4
0.625	0.7093	0.1711	223.1	683.1
0.750	0.7071	0.1671	225.9	692
0.875	0.7213	0.1712	227.8	692

set of 82.5 percent, and from the radius of .5 on, the difference is less than 3 percent with reduction in size of the training set of about 16.3 percent. The difference in coverage is less than .4 percent from $r = .25$ on, where reduction in training set size is 82.5 percent.

For LEM2, accuracy in both cases differs by less than 1 percent from $r = .25$ on, and it is better in granular case from $r = .5$ on with reduction in size of the training set of 16.3 percent; coverage is better in granular case from $r = .375$ on with the training set size reduced by 48.2 percent.

5.3 A Validation by a Statistical Test

We have also carried out the test with Pima Indian Diabetes dataset [14], and random choice of coverings, taking a sample of 30 granular classifiers at the radius of .5 with train-and-test at the ratio 1:1 against the matched sample of classification results without granulation, with the covering algorithm for p=.1. The Wilcoxon [15] signed rank test for matched pairs in this case has given the p–value of .14 in case of coverage,so the null hypothesis of identical means should not be rejected, whereas for accuracy, the hypothesis that the mean in granular case is equal to .99 of the mean in non–granular case may be rejected (the p–value is .009), and the hypothesis that the mean in granular case is greater than .98

of the mean in non–granular case is accepted (the p–value is .035) at confidence level of .03.

6 Concept–Dependent Granulation

A modification of the approach presented in results shown above is the *concept dependent* granulation; a *concept* in the narrow sense is a decision/classification class, cf., e.g., [2]. Granulation in this sense consists in computing granules for objects in the universe U and for all distinct granulation radii as previously, with the only restriction that given any object $u \in U$ and $r \in [0,1]$, the new concept dependent granule $g^{cd}(u,r)$ is computed with taking into account only objects $v \in U$ with $d(v) = d(u)$, i.e., $g^{cd}(u,r) = g(u,r) \cap \{v \in U : d(v) = d(u)\}$. This method increases the number of granules in coverings but it is also expected to increase quality of classification, as expressed by accuracy and coverage.

We show that this is the case indeed, by including results of the test in which exhaustive algorithm and random choice of coverings were applied tenfold to Australian credit data set, once with the ”standard” by now granular approach and then with the concept dependent approach. The averaged results are shown in Table 7.

Conclusions for concept dependent granulation Concept dependent granulation, as expected, involves a greater number of granules in a covering, hence, a greater number of rules, which is perceptible clearly up to the radius of .714286 and for greater radii the difference is negligible. Accuracy in case of concept

Table 7. Standard and concept dependent granular systems for Australian credit data set; exhaustive RSES algorithm:r=granule radius, macc=mean accuracy, mcov=mean coverage, mrules=mean number of rules, mtrn=mean training sample size; in each column first value is for standard, second for concept dependent

r	*macc*	*mcov*	*mrules*	*mtrn*
nil	1.0; 1.0	1.0; 1.0	12025; 12025	690; 690
0.0	0.0; 0.8068	0.0; 1.0	0; 8	1; 2
0.0714286	0.0; 0.7959	0.0; 1.0	0; 8.2	1.2; 2.4
0.142857	0.0; 0.8067	0.0; 1.0	0; 8.9	2.4; 3.6
0.214286	0.1409; 0.8151	0.2; 1.0	1.3; 11.4	2.6; 5.8
0.285714	0.7049; 0.8353	0.9; 1.0	8.1; 14.8	5.2; 9.6
0.357143	0.7872; 0.8297	1.0; 0.9848	22.6; 32.9	10.1; 17
0.428571	0.8099; 0.8512	1.0; 0.9986	79.6; 134	22.9; 35.4
0.5	0.8319; 0.8466	1.0; 0.9984	407.6; 598.7	59.7; 77.1
0.571429	0.8607; 0.8865	0.9999; 0.9997	1541.6; 2024.4	149.8; 175.5
0.642857	0.8988; 0.9466	1.0; 0.9998	5462.5; 6255.2	345.7; 374.9
0.714286	0.9641; 0.9880	1.0; 0.9988	9956.4; 10344.0	554.1; 572.5
0.785714	0.9900; 0.9970	1.0; 0.9995	11755.5; 11802.7	662.7; 665.7
0.857143	0.9940; 0.9970	1.0; 0.9985	11992.7; 11990.2	682; 683
0.928571	0.9970; 1.0	1.0; 0.9993	12023.5; 12002.4	684; 685
1.0	1.0; 1.0	1.0; 1.0	12025.0; 12025.0	690; 690

dependent granulation is always better than in the standard case, the difference becomes negligible at the radius of .857143 when granules become almost single indiscernibility classes. Coverage in concept dependent case is almost the same as in the standard case, the difference between the two not greater than .15 percent from the radius of .428571, where the average number of granules in coverings is 5 percent of the number of objects. Accuracy at that radius is better by .04 i.e. by about 5 percent in the concept dependent case.

It follows that concept dependent granulation yields better accuracy whereas coverage is the same as in the standard case.

7 Conclusions

The results shown in this work confirm the hypothesis put forth in [7], [8] that granular counterparts to data sets preserve the encoded information to a very high degree. The search for theoretical explanation for this as well as work aimed at developing original algorithms for rule induction based on the discovered phenomenon are in progress to be reported.

References

1. Bazan, J.G.: A comparison of dynamic and non–dynamic rough set methods for extracting laws from decision tables. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery 1, pp. 321–365. Physica Verlag, Heidelberg (1998)
2. Grzymala–Busse, J.W.: Data with missing attribute values: Generalization of rule indiscernibility relation and rule induction. In: Transactions on Rough Sets I, pp. 78–95. Springer, Berlin (2004)
3. Leśniewski, S.: On the foundations of set theory. Topoi 2, 7–52 (1982)
4. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)
5. Polkowski, L.: Rough Sets. Mathematical Foundations. Physica Verlag, Heidelberg (2002)
6. Polkowski, L.: Toward rough set foundations. Mereological approach (a plenary lecture). In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 8–25. Springer, Heidelberg (2004)
7. Polkowski, L.: Formal granular calculi based on rough inclusions (a feature talk) in: [10], pp. 57–62
8. Polkowski, L.: A model of granular computing with applications (a feature talk) in: [11], pp. 9–16
9. Polkowski, L., Skowron, A.: Rough mereology: a new paradigm for approximate reasoning. International Journal of Approximate Reasoning 15(4), 333–365 (1997)
10. In: Proceedings of IEEE 2005 Conference on Granular Computing,GrC05, Beijing, China, July 2005, IEEE Press, New York (2005)
11. In: Proceedings of IEEE 2006 Conference on Granular Computing, GrC06, Atlanta, USA, May 2006, IEEE Press, New York (2006)
12. Skowron, A., et al.: RSES: A system for data analysis, available at `http://logic.mimuw.edu.plrses/`

13. Nguyen, S.H.: Regularity analysis and its applications in Data Mining. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.) Rough Set Methods and Applications, pp. 289–378. Physica Verlag, Heidelberg (2000)
14. http://www.ics.uci.edu.mlearn/databases/iris
15. Wilcoxon, F.: Individual comparisons by ranking method. Biometrics 1, 80–83 (1945)
16. Wojna, A.: Analogy–based reasoning in classifier construction. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets IV. LNCS, vol. 3700, pp. 277–374. Springer, Heidelberg (2005)
17. Wróblewski, J.: Adaptive aspects of combining approximation spaces. In: Pal, S.K., Polkowski, L., Skowron, A. (eds.) Rough Neural Computing, pp. 139–156. Springer, Heidelberg (2004)

A Rough Set Based Map Granule

Sumalee Sonamthiang[1], Nick Cercone[2], and Kanlaya Naruedomkul[3]

[1] Institute for Innovation and Development of Learning Process
Mahidol University, Bangkok, Thailand 10400
g4637886@mahidol.ac.th
[2] Faculty of Science & Engineering, York University
Toronto, Ontario,Canada M3J 1P3
ncercone@yorku.ca
[3] Mathematic Department, Faculty of Science
Mahidol University, Bangkok, Thailand 10400
scknr@mahidol.ac.th

Abstract. Data in an information system are usually represented and stored in a flat and unconnected structure as in a table. Underlying the data structure, there is a domain concept that is an understandable description for humans and supports other machine learning techniques. In this work, Map Granule (MG) construction is introduced. A MG comprises of multilevel granules with their hierarchy relations. We propose a rough set based granular computing to induce approximation of a domain concept hierarchy of an information system. An algorithm is proposed to select a sequence of attribute subsets which is necessary to partition a granularity hierarchically. In each level of granulation, reducts and core are applied to retain the specific concepts of a granule whereas common attributes are applied to exclude the common knowledge and generate a more general concept. The information granule relations are represented by a tree structure in which the relation strengths are defined by a rough ratio of specificness/coarseness.

Keywords: Map Granules, Information Granules, Concept Hierarchy, Rough Set Theory.

1 Introduction

An Information System (IS) in a rough set paradigm [1] is a basic knowledge representation method in an attribute-value system. An IS is represented in a table that a row keeps an object and each column keeps the value of the corresponding attribute. The tabular representation simplifies recording the objects into an IS, especially, in realtime transactions by capturing a transaction separately and using single global representation for every record in every situation. However, an occurrence of a transaction may be related to other transactions in the problem space. Representation in this fashion is seen as flat and unconnected structure which hides the meaningfully relations in the pile of data. In this work, we introduce a hierarchical granulation to construct a map of granules called a Map Granule (MG) for an IS.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 290–299, 2007.

MG is a hierarchy knowledge representation of a domain which provides a multilevel of granularity. The hierarchy can be conveniently represented by a tree structure. A tree comprises of a root node, nonroot nodes, and the relations. A node in a tree can be seen as a granule in which instances in the node hold similar properties to a certain degree and they are part of their parent. Thus, a parent holds the common properties of its children and the siblings have a certain degree of similarity to each other by the common properties.

An MG is defined under formal concept analysis (FCA) which is a theory for identifying conceptual structures among data sets. A concept can be defined by a concept's intention and extension [2]. From an intension, one can describe a concept by means of logical descriptors from ,e.g. , a set of concept's attributes and the attributes' values, terms of a concept, and metadata of a concept. The extension of a concept is an illustration by a possibly empty set of objects that belong to the concept. The formal concept approach provides two dimensions for indexing a granule in an MG. Thus, a granule is accessible either by an attribute subset or its membership functions. In this work, we describe a granule by a concept's intension using a predicate description of common properties, internal inclusion and relations to other granules, and extensions.

This paper presents a rough set based approach to construct a MG. An algorithm is proposed to select a sequence of attribute subsets which is necessary to partition a granularity hierarchically. Then rough set approach is used for granules formation. In other words, reducts and core are applied to retain the specific concepts of a granule whereas common attributes (the subset of attributes that all instances in a granule have in common) are applied to exclude the common knowledge and generate a more general concept.

The paper is organized as follows: In the next section we explore the related works in hierarchical information granulation. Section 3 gives the preliminaries of rough set theory and our approach is described in section 4. Section 5 reports our evaluation and results. Finally, section 6 presents some conclusions and discussions of possible extensions.

2 Related Works

In this section, we describe the previous works that use a rough set theory approach to extract hierarchical granules. Hoa and Son [5] introduced a complex concept approximation approach based on a layered learning method together with rough set theory. They used taxonomy as the domain knowledge and feature values in the dataset to guide composing attributes into intermediate concepts until obtaining the target concept. The target concepts are the concepts in the decision attribute. However, the domain taxonomies are usually unavailable to guide the layer learning.

Yao and Yao presented a granularity-based formal concept definition in [4]. They define a formal concept by a pair of its intension and extension $(\phi, m(\phi))$, where ϕ is a logical rule of a subset of attributes with the attributes' values and $m(\phi)$ is a granule obtained by partitioning the universe of objects using the attribute subset ϕ. Moreover, Yao [3] presented an approach to hierarchical granulation based on rough sets called stratified rough approximation. The stratified rough set approximation is a simple multi-level granulation based on nesting of one-level granulation (e.g., granulation by the equivalence relation). Yao [3] presented three methods to multi-layered granulation which are:

- nested rough set approximations induced by a nested sequence of equivalence relations,
- stratified rough set approximations induced by hierarchies, and
- stratified rough set approximations induced by neighborhood systems.

In the nested granulation approach, the granulation starts by indiscernibility relations on the set of attributes. Then the subsequent indiscernibility relations are defined by successively removing attributes from the set of remaining attributes. Sequencing of attribute subsets for partitioning is determined by dependencies between condition attributes. The sequence of attribute subset affects a granules' content and the hierarchy structure. Moreover, the order of attributes subset for partitioning is very important in the sense of the closeness between instances in the same granule. However, some ISs have no attributes' dependency. We introduce attribute subset sequencing method in section 4.1. Our method is able to partition any IS hierarchically. In the stratified rough set approximations induced by hierarchies, levels of hierarchies provides the sequence of granulation. As mentioned above, the hierarchy of a domain may be unavailable for an IS. Yao [3] also described using neighborhood systems [8] to induce hierarchial partitioning. He recommended that the stratified approximation can be used to search for an appropriate level of accuracy for an application. We see the hierarchy from a communication amongst granules [7] point of view. An MG provides rich information about domain structure. An application can uses an MG at task level as well as at application levels that shares the MG amongst applications.

In the next section, an algorithm to select attribute subset for partitioning an IS and an MG construction using rough sets are described.

3 Rough Set Preliminaries

In an information system $IS = (U, A, D, V)$, let U be the universe of the IS containing a finite and non-empty set of instances. Let x be an instance of the universe and X be a subset of U. A is the set of condition attributes of the elements which are the features of instances and D is the decision attribute. A finite and nonempty set V is the set of all attribute values $\{v_1, \ldots, v_n\}$. An information granule $X \subseteq U$ can be categorized by a pair of lower and upper approximations as follows:

$$LOWER(X) = \bigcup\{[x]_B | x \in U, [x]_B \subseteq X\}$$
$$UPPER(X) = \bigcup\{[x]_B | x \in U, [x]_B \cap X \neq \emptyset\}$$

where $[x]_B = \bigcup\{[(a, v)] | a \in B, B \subseteq A\}, f(x, a) = v$. The boundary region of rough set is defined as:

$$BND(X) = UPPER(X) - LOWER(X).$$

To partition the set U into disjoint subsets of granules, we use indiscernibility relation $IND(B)$ which is a relation on U defined for $x, y \in U$ as follows:

$$(x, y) \in IND(B) \iff f(x, a) = f(y, a), \forall a \in B.$$

The partitions of the universe is called the quotient set induced by B and is denoted by $\{B\}^*$.

The presence of an element x in a concept X (determined by a subset attributes B) is defined as the following function:

$$\mu_{B,X}(x) = \frac{|[x]_B \cap X|}{|[x]_B|}$$

which is called rough membership function. The $|X|$ denotes the cardinality of a set X. For the empty set , we define $\mu(\emptyset) = 1$. It is obvious that $\mu_{B,X}(x) = 1$ when $x \in LOWER(X)$. The accuracy of rough approximation is given by:

$$\alpha(X) = \frac{LOWER(X)}{UPPER(X)}$$

The approximation accuracy is in the range of $0 \geq \alpha(X) \geq 1$, and $\alpha(\emptyset) = 1$.

In an information system, if there is a subset of attributes that are sufficient to describe the decision attributes, the subset of attributes is called reduct (RED).

$$RED \subseteq A | [x]_{RED} = [x]_A$$
$$[x]_{RED'} = [x]_A \text{ where } \forall RED' \subset RED$$

The equivalence classes induced by RED is the same as the equivalence class induced by full attribute set A. Intersection of all the reducts is called core.

$$CORE = \bigcap\{\forall RED\}$$

The core attributes are common to all reducts; therefore, they cannot be removed from an IS without effects on the equivalence class structure.

4 Map Granule Construction

A map granule (MG) is comprised of a root, a set of nodes, and three relations between nodes. A node holds a nonempty set of instances. A node itself can be either an individual node or a MG. The three relation between nodes are parent-child, child-parent, and sibling. Parent-child relation indicates that the parent node holds common features amongst its children, whereas the nodes that have the same parent have a sibling relation to each other. In this section we introduce MG construction framework, an algorithm to select attribute subset for partitioning, description of a granule, and the measurement of internal granule and a MG.

4.1 Algorithms for Map Granule Construction

The MG construction is a recursive granulation in depth-first manner. Specifically, the recursive MG construction is given in algorithm 1.

Algorithm 1. MG Construction
Input: an information system $\{U, A, D, V\}$
Output: a map granule

```
set $g = U = root$
set $tempIS = \{g, A, D, V\}, C \subseteq A$ and set $C = \emptyset$
FUNCTION MGconstruct($g$)
  IF $g$ is discernible THEN
    1. find $C | f(x, c \in C) = v_i, \forall x \in TempIS, \forall c \in C$
       ($C$ is a subset of common attributes)
    2. IF $C \neq \emptyset$ THEN
       set $tempIS = \{g, A, D, V\}$ where $A = A - C$
    3. generate a granule description (see section 4.2)
    4. select the most dominant attribute subset $B$ of $tempIS$
       (see algorithm 2.)
    5. partition the $tempIS$ by $B, \{B\}^* = g_1, g_2, \ldots, g_n$
    6. generate parent-child, child-parent and sibling relations
    7. calculate rough measurement (see section 4.3 and 4.4)
    8. mark $g$ as granulated
    9. set $g = g_1$ (move to the next level of granularity)
    10. set $tempIS = \{g, A, D, V\}$
  ELSE make a leaf granule
  FOR $i = 1$ to $\#g$ nongranulated siblings of $g$
    MGconstruct($g_i$)
```

From the algorithm, we obtain a MG from a recursive tree construction. The process begins with finding common attribute subset. Then, a temporary IS is derived from the current IS by removing the common attributes. The derived IS is not necessary if there is no common attribute. The attribute sequencing is accomplished through local attributes subset selection in the recursive partitioning. We select the most dominant attribute subset based on the attributes' values available in the information system. We determine the domination using algorithm 2. The selected attributes subset is then used to partition the temporary IS and assign relationships between the obtained granules (children) and the original granule (parent). If a granule cannot be partitioned by indiscernibility relation, a leaf node is generated.

Algorithm 2 describes the most dominant attribute subset selection. The rough set exploration system (RSES version 2.2) [6] is used to calculate reducts of the universe. Then $CORE$ can be derived from intersection of all reducts. Given an IS(temporary), we find the N most dominant attributes toward the decision class. $CORE$ is used to preserve the specific feature(s) of instances in the granule by retaining $CORE$ until the latest granulations. N can be tuned up to the number of condition attributes to compose a concept. In other words, our algorithm allows a flexible number of attributes in a subset for partitioning. We use co-occurrence counting of attributes' values and decision classes to determine the domination degree. Once the most N dominant attributes are obtained, we determine the co-occurrences with each other to find if any combination of them can be used to approximate a concept by threshold ε. A count of co-occurrence amongst condition attributes' values implies the degree of which these attribute values can be used to compose a common concept. We tune the ε by the number of instances in working IS. The subset of attributes that are greater than the threshold is selected to partition the IS. If no combination of them meets the threshold ε, the single most dominant attribute is selected.

Algorithm 2. The Most Dominant Attribute Subset Selection
Input: an information system, $CORE$ of U, parameter N and ε (described below)
Output: the most dominant attribute subset

```
set $MostDA = null$ (the most dominant attribute)
set $TopDA = null$ (top dominant attribute subset)
FOR all attributes
    FOR all attribute's values
        count number of co-occurrences with each decision class
        set AD (attribute's domination) = maximum count
    set $MostDA$ = the attribute that has maximum $AD$
    set $TopDA$ = a set of attributes in top $N$ most dominant attribute
Set $BD = MostDA$
FOR all $B \subseteq TopDA, |B| > 1$
    count number of co-occurrences of attributes' values in $B$
    IF count $> \varepsilon$ THEN Set $BD = B$
IF $BD - CORE \neq \emptyset$ THEN set $BD = BD - CORE$
RETURN $BD$
```

4.2 Description of a Granule

A granule can be described by a logical language such as [2][4]. In this work, a description for a granule is defined by the common features amongst instances in the granule.

$$DES = \{C(v_i)\} \iff f(x,c) = v_i, \forall x \in g \text{ and } \forall c \in C$$

where a description of a granule DES is defined by a set of predicates. A predicate C is labeled by an attribute name and v_i is the predicate's argument. All instances in the granule g have the attribute's value v_i in common. If there is no such v_i, $DES = \emptyset$.

4.3 Granule's Internal Measurement

G is a set of granules from partitioning $\{B\}^*$. Given an information granule $g \in G$, g contains finite elements $x_1, \ldots, x_n$. An accuracy of approximation to a specific decision class is determined by $\alpha(g)$. For the closeness of elements in a granule, Spearman's footrule distance measurement is applied. The foot rule distance is given by:

$$d_{ij} = \sum_{k=1}^{n} |x_{ik} - x_{jk}|,$$

where n is the number of condition attributes. Thus, the average distance can represent the degree of closeness of elements in a granule.

$$Cls(g) = \frac{\sum_{i=1}^{N} d_i}{N},$$

N is the number of element pairs.

4.4 Distance Measurement in a Map Granule

For every $g \in G$, the rough distance between a pair of granules is determined as follows:

$d(g_1, g_2) = 1 - \frac{|g_1 \cap g_2|}{|g_1 \cup g_2|}, g_1 \neq g_2$

IF $0 < d(g_1, g_2) < 1, g_1 \supset g_2$ THEN $g_1$is ancestor of g_2,

IF $d(g_1, g_2) = 1$ THEN g_1, g_2 are independent.

The rough relation strength between a parent and a child defined on a target class can be calculated by a ratio of coarseness and specificness (γ) by:

$$\gamma = \frac{\alpha(g_2)}{\alpha(g_1)},$$

where g_1 is the parent of g_2. One can say that g_1 is γ times coarser than g_2 or g_2 is γ times more specific than g_1.

4.5 An Example of Map Granule Construction

This section illustrates an example of hierarchical granulation to obtain a MG. The example IS is a flu diagnosis domain provided in table 1.

Table 1. Flu diagnosis

Cases	Temperature	Headache	Nausea	Cough	Decisions(Flu)
1	high	yes	no	yes	yes
2	very high	yes	yes	no	yes
3	high	no	no	no	no
4	high	yes	yes	yes	yes
5	normal	yes	no	no	no
6	normal	no	yes	yes	no

We will describe how the algorithm works step by step. The granulation starts by partitioning the universe (given IS). In this example, the equivalence relation is used. The size of attribute subset to partition is one since there are small number of attribute. The IS is discernible by an equivalence relation. Thus, we find reducts for the IS which are, {Temperature, Headache, Nausea}, {Temperature, Nausea, Cough}, and {Headache, Nausea, Cough}, and core is {Nausea}. There is no common attribute value in this granule. We select the first attribute subset by determining the degree of attribute dominations. The first attribute subset to partition is {Headache} and $g_1 = \{1, 2, 4, 5\}$ and $g_2 = \{3, 6\}$ are obtained.

Then we continue granulate g_1 selecting the most dominant attributes for g_1. Temperature, Nausea and Cough attributes have the same degree of domination. Nausea is the core; thus, it is retained at this granulation. We can select Temperature or Cough to partition g_1. If we apply Temperature, we obtain granule $g_3 = \{1, 4\}$, $g_4 = \{5\}$, $g_5 = \{2\}$ which are children of g_1. The granule g_4 and g_5 are indiscernible so they are leaf granule. We then granulate g_3 by finding common attribute subset which is {Cough}. The Cough attribute can be now removed. The remaining attribute {Nausea} is then used to partition g_3 to obtain $g_6 = \{2\}, g_7 = \{3\}$. Since all siblings are now leaf nodes we can return to the higher levels. We continue granulate g_2. Note that the temporary table can be generated as the Headache attribute is removed. Like partitioning g_1, Nausea is retained. If we partition g_2 by Temperature, the indiscernible granule $g_8 = \{3\}$ and $g_9 = \{6\}$ are obtained. Fig. 1 shows the MG for the Flu diagnosis domain.

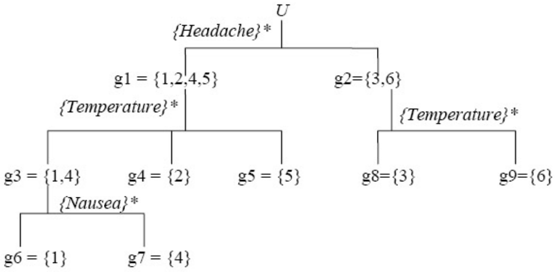


Fig. 1. A map granule for the Flu case base

5 Evaluation and Results

We evaluate our approach using the Zoo dataset which is available in the UCI machine learning data repository. This database contains 16 boolean-valued attributes, 1 numerical attribute, and a Type attribute as the class attribute. The class attribute contains 7 classes. There is no missing value in this dataset. We construct an MG for the Zoo dataset as shown in figure 1. The attribute Name is removed since it is the index of animals. Parameter $N = 3$ and $\varepsilon = 0.5$ are set. In the figure, indentation is used to show the hierarchy structure. A granule is named by G with a list of number attached. The length of a granule name indicates the level in the hierarchy starting from 1 at the root. *CORE* is {Aquatic, Legs}. The attribute subset used in first partitioning is {Feathers, Milk, Backbone}* and granule G1, G2, G3, and G4 are obtained. Then G1 is granulated and children G11, G12, and G13 of G1 are formed by the second granulation and so on.

The intermediate granules in the MG represent a cluster of animal that have some degree of similarity to other animals in the same granule. For example, G22={chicken, dove, parakeet} these animals have common features which are:

- G1: {squirrel, fruitbat, vampire, hare, vole, mole, opossum, cavy, hamster, seal, gorilla, aardvark, bear, dolphin, porpoise, wallaby, sealion, platypus, antelope, buffalo, deer, elephant, giraffe, oryx, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf, mink, girl, calf, goat, pony, reindeer, pussycat}
  - G11: {fruitbat, vampire}
  - G12: {platypus}
  - G13: {squirrel, hare, vole, mole, opossum, cavy, hamster seal, gorilla, aardvark, bear, wallaby, sealion, antelope, buffalo, deer elephant ,giraffe, oryx , boar ,cheetah ,leopard ,lion ,lynx ,mongoose, polecat, puma ,raccoon, wolf, mink, girl, dolphin, porpoise, calf, goat, pony, reindeer}
    - G131: {dolphin, porpoise}
    - G132: {squirrel, hare, vole, mole, opossum, cavy, hamster, seal, gorilla, aardvark, bear, wallaby, sealion, antelope, buffalo, deer, elephant, giraffe, oryx, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf, mink, girl calf, goat, pony, reindeer}
      - G1321: {squirrel, gorilla, wallaby, girl, hare, vole, cavy, hamster, antelope, buffalo, deer, elephant, giraffe, oryx, calf, goat, pony, reindeer, mole, opossum, aardvark, bear, boar, cheetah, leopard, lion, lynx, mongoose polecat, puma, raccoon, wolf, pussycat}
        - G13211: {hare, vole, antelope, buffalo, deer, elephant, giraffe, oryx, mole, opossum aardvark, bear, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf}
          - G132111: {mole, opossum, boar, cheetah, leopard, lion, lynx, mongoose, polecat, puma, raccoon, wolf}
          - G132112: {aardvark, bear}
          - G132113:{hare, vole, antelope, buffalo, deer, elephant, giraffe, oryx}
        - G13212: {squirrel, gorilla, wallaby}
        - G13213: {girl}
        - G13214: {cavy, hamster, calf, goat, pony, reindeer, pussycat}
          - G1321411: {cavy}
          - G1321412: {hamster calf goat pony reindeer}
          - G1321413: {pussycat}
      - G1322: {seal, sealion}
      - G1323: {mink}
- G2: {chicken, dove, duck, lark, parakeet, pheasant, sparrow, wren, kiwi, crow, gull, hawk, skimmer, skua, ostrich, flamingo, swan, penguin, rhea, vulture}
  - G21: {lark, pheasant, sparrow, wren, duck, kiwi, crow, hawk, gull, skimmer, skua}
    - G211: {kiwi}
    - G212: {lark, pheasant, sparrow, wren, duck, crow, hawk, gull, skimmer, skua}
      - G2121: {lark, pheasant, sparrow, wren, crow, hawk}
        - G21211: {lark, pheasant, sparrow, wren}
        - G21212: {crow, hawk}
      - G2122: {duck, gull, skimmer, skua}
        - G21221: {duck}
        - G21222: {gull, skimmer, skua}
  - G22: {chicken, dove, parakeet}
  - G23: {ostrich, flamingo, swan, rhea, vulture, penguin}
    - G231: {ostrich, flamingo, rhea, vulture}
      - G2311: {ostrich, rhea}
        - G23111: {ostrich}
        - G23112: {rhea}
      - G2312: {flamingo, vulture}
        - G23121: {flamingo}
        - G23122: {vulture}
    - G232: {swan penguin}
      - G2321: {swan}
      - G2322: {penguin}
- G3: {pitviper, seasnake, slowworm, tortoise, tuatara, carp, haddock, seahorse, sole, bass, catfish, chub, dogfish, herring, pike, piranha, stingray, tuna, frog, frog, newt, toad}
  - G31: {bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna}
    - G311: {carp, haddock, seahorse, sole}
    - G312: {bass, catfish, chub, herring, piranha, dogfish, pike, tuna}
      - G3121: {bass, catfish, chub, herring, piranha}
      - G3122: {dogfish, pike, tuna}
    - G313: {stingray}
  - G32: {pitviper, slowworm, tortoise, tuatara}
  - G33: {seasnake}
  - G34: {frog, frog, newt, toad}
    - G341: {frog, newt}
      - G3411: {frog}
      - G3412: {newt}
    - G342: {frog}
    - G343: {toad}
- G4: {flea, termite, gnat, honeybee, housefly, ladybird, moth, wasp, clam, scorpion, slug, worm, crab, crayfish, lobster, octopus, seawasp, starfish}
  - G41: {clam, scorpion, slug, worm, crab, crayfish, lobster, octopus, seawasp, starfish, flea, termite}
    - G411: {seawasp, starfish, flea, termite, slug, worm, crab, crayfish, lobster, clam}
      - G4111: {flea, termite, slug, worm}
        - G41111: {flea, termite}
        - G41112: {slug worm}
      - G4112: {seawasp}
      - G4113: {starfish, crab, crayfish, lobster, clam}
        - G41131: {clam}
        - G41132: {starfish, crab, crayfish, lobster}
          - G411321: {crab}
          - G411322: {starfish}
          - G411321: {crayfish, lobster}
    - G412: {scorpion}
    - G413: {octopus}
  - G42: {gnat, ladybird}
    - G421: {gnat}
    - G422: {ladybird|}
  - G43: {housefly, moth, wasp}
    - G431: {housefly, moth}
    - G432: {wasp}
  - G44: {honeybee}

Fig. 2. A map granule for the Zoo database

chicken	0	1	1	0	1	0	0	0	1	1	0	0	2	1	1	0	2
dove	0	1	1	0	1	0	0	0	1	1	0	0	2	1	1	0	2
parakeet	0	1	1	0	1	0	0	0	1	1	0	0	2	1	1	0	2

These three animals have the same features and can be only continually partitioned by Name attribute. They are in the same class and the approximation accuracy toward the class is 1. The closeness measurement for this granule is 0 which means that the average distance between instances in G22 is 0. The intension of granule G22 is: $DES_{G22} = \{hair(0), feathers(1), eggs(1), milk(0), airborne(1), aquatic(0), predator(0), toothed(0), backbone(0), fins(0), legs(2), tail(1), breathes(1), venomous(1), domestic(1), catsize(0)\}$.

6 Concluding Remarks

An approach to automatically construct a map granule from an information system is presented. A map granule represents knowledge in different level of specificness/coarseness. We demonstrated that the granules obtained from our approach have meaningful inclusion degrees by mean of sequencing attribute subsets for granulation hierarchically. We presented three dimensions to describe a granule: a predicate description of common properties, internal inclusion and relations to other granules, and granule's extensions. With these rich information, we believe that an MG can be used to support other machine learning methods. The possible extensions of this work include construction of the concept hierarchy from the more challenging original knowledge representation such as time dependence problems and user modeling.

References

1. Pawlak, Z.: Rough Sets. In: Theoretical Aspects of Reasoning about Data, Kluwer, Netherlands (1991)
2. Priss, U.: Formal Concept Analysis in Information Science. Annual Review of Information Science and Technology number 40 (2006)
3. Yao, Y.Y.: Information Granulation and Rough Set Approximation. International Journal of Information Systems 16, 87–104 (2001)
4. Yao, Y.Y., Yao, J.T.: Granular Computing as a Basis for Consistent Classification Problems. In: Proceedings of PAKDD'02 Workshop on Toward the Foundation of Data Mining 2002, pp. 101–106 (2002)
5. Hoa, N.S., Son, N.H.: Rough Set Approach to Approximation of Concepts from Taxonomy. In: Proc. of Knowledge Discovery and Ontologies Workshop (KDO-04) at ECML/PKDD 2004, pp. 13–24 (2004)
6. Bazan, J.G., Szczuka, M.: The Rough Set Exploration System. In: Transactions on Rough Sets III, pp. 37–56. Springer, Heidelberg (2005)
7. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction, pp. 11–16. Springer, Heidelberg (2002)
8. Yao, Y.Y.: Granular computing using neighborhood systems. Advances in Soft Computing: Engineering Design and Manufacturing. In: Roy, R., Furuhashi, T., Chawdhry, P.K. (eds.) The 3rd On-line World Conference on Soft Computing (WSC3), June 21-30, 1998, pp. 539–553. Springer, London (1999)

Modeling of High Quality Granules

Andrzej Skowron[1] and Jaroslaw Stepaniuk[2]

[1] Institute of Mathematics
Warsaw University
Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl

[2] Department of Computer Science
Bialystok University of Technology
Wiejska 45A, 15-351 Bialystok, Poland
jstepan@wi.pb.edu.pl

Abstract. In Granular Computing (GC) we search for granules satisfying some criteria. These criteria can be based on the minimal length principle, can express acceptable risk degrees of granules, or can use some utility functions. We discuss the role of approximation spaces in modeling granules satisfying such criteria.

1 Introduction

Information granulation can be viewed as a human way of achieving data compression and it plays a key role in implementing the divide-and-conquer strategy in human problem-solving [22]. Granules are obtained in the process of information granulation. Granular computing (GC) is based on processing of complex information entities called granules. Generally speaking, granules are collection of entities, that are arranged together due to their similarity, functional adjacency or indistinguishability [22].

One of the main branch of GC is Computing with Words and Perceptions (CWP). GC "derives from the fact that it opens the door to computation and reasoning with information which is perception - rather than measurement-based. Perceptions play a key role in human cognition, and underlie the remarkable human capability to perform a wide variety of physical and mental tasks without any measurements and any computations. Everyday examples of such tasks are driving a car in city traffic, playing tennis and summarizing a story" [22].

We consider the optimization tasks in which we are searching for optimal solutions satisfying some constraints. These constraints are often vague, imprecise, and/or specifications of concepts and dependencies between them involved in the constraints are incomplete. Decision tables [11] are examples of such constraints. Another example of constraints can be found, e.g., in [4,15] where a specification is given by a domain knowledge and data sets. Domain knowledge is represented by ontology of vague concepts and dependencies between them. In a more general case, the constraints can be specified in a simplified fragment of a natural language [22].

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 300–309, 2007.

Granules are constructed in computations aiming at solving the mentioned above optimization tasks. In our approach, we use the general optimization criterion based on the minimal length principle. In searching for (sub-)optimal solutions it is necessary to construct many compound granules using some specific operations such as generalization, specification or fusion. Granules are labeled by parameters. By tuning these parameters we optimize the granules relative to their description size and the quality of data description, i.e., two basic components on which the optimization measures are defined.

From this general description of tasks in GC it follows that together with specification of elementary granules and operation on them it is necessary to define measures of granule quality (e.g., measures of their inclusion, covering or closeness) and tools for measuring the size of granules. Very important are also optimization strategies of already constructed (parameterized) granules.

We discuss the searching process for relevant (for concept approximation) neighborhoods in approximation spaces based on modeling relevant relational and syntactical structures build from partial information about objects and concepts.

The importance in GC of risk measures defined on granules is emphasized. The values of such measures are indicating how properties of granules are changing when some of their parameters were changed.

We present an example showing how utility functions defined on granules can be used in GC. In general, utility functions are helping to relax the binary constraints by making it possible to work with constraints which should be satisfied to a degree expressed by utility functions.

This paper is structured as follows. In Section 2 we discuss definitions of approximation spaces and approximations. In Section 3 we discuss constraints that must be satisfied during the information granulation process. In Section 4 we present some remarks about risk in construction of granules.

2 Approximation Spaces and Approximations

In this section, we discuss the definition of an approximation space from [13,19]. Approximation spaces can be treated as granules used for concept approximation. They are some special parameterized relational structures. Tuning of parameters is making it possible to search for relevant approximation spaces relative to given concepts.

Definition 1. *A parameterized approximation space is a system* $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$, *where*

- U *is a non-empty set of objects,*
- $I_{\#} : U \rightarrow P(U)$ *is an uncertainty function, where* $P(U)$ *denotes the power set of* U,
- $\nu_{\$} : P(U) \times P(U) \rightarrow [0, 1]$ *is a rough inclusion function,*

and $\#, \$$ *denote vectors of parameters (the indexes* $\#, \$$ *will be omitted if it does not lead to misunderstanding).*

The uncertainty function defines for every object x, a set of objects described similarly to x. The set $I(x)$ is called the neighborhood of x (see, e.g., [11,13]).

The rough inclusion function $\nu_{\$} : P(U) \times P(U) \rightarrow [0,1]$ defines the degree of inclusion of X in Y, where $X, Y \subseteq U$.

In the simplest case it can be defined by (see, e.g., [13,11]):

$$\nu_{SRI}(X,Y) = \begin{cases} \frac{card(X \cap Y)}{card(X)} & \text{if } X \neq \emptyset \\ 1 & \text{if } X = \emptyset. \end{cases}$$

The lower and the upper approximations of subsets of U are defined as follows.

Definition 2. *For any approximation space $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$ and any subset $X \subseteq U$, the lower and upper approximations are defined by*
$LOW\left(AS_{\#,\$}, X\right) = \{x \in U : \nu_{\$}\left(I_{\#}(x), X\right) = 1\}$,
$UPP\left(AS_{\#,\$}, X\right) = \{x \in U : \nu_{\$}\left(I_{\#}(x), X\right) > 0\}$, respectively.

The lower approximation of a set X wit respect to the approximation space $AS_{\#,\$}$ is the set of all objects, which can be classified with certainty as objects of X with respect to $AS_{\#,\$}$. The upper approximation of a set X with respect to the approximation space $AS_{\#,\$}$ is the set of all objects which can be possibly classified as objects of X with respect to $AS_{\#,\$}$.

Several known approaches to concept approximations can be covered using the discussed here approximation spaces, e.g., (see, e.g., references in [13]).

One can use yet another approach to approximation based on a fusion of inclusion degree of neighborhoods in concepts and their complements in definition of approximations. Let $f : [0,1] \longrightarrow [0,1]$ denote such a fusion function. For any subset $X \subseteq U$, the lower and upper approximations are defined by

$$LOW(AS_{\#,\$}, X) = \{x \in U : f(\{\nu_{\$}(I_{\#}(y), X) : x \in I_{\#}(y)\}) = \{1\}\},$$

$$UPP(AS_{\#,\$}, X) = \{x \in U : f(\{\nu_{\$}(I_{\#}(y), X) : x \in I_{\#}(y)\}) \neq \{0\}\}.$$

The classification methods for concept approximation developed in machine learning and pattern recognition make it possible to decide for a given object if it belongs to the approximated concept or not. The classification methods yield the decisions using only partial information about approximated concepts. This fact is reflected in the rough set approach by assumption that concept approximations should be defined using only partial information about approximation spaces. To decide if a given object belongs to the (lower or upper) approximation of a given concept the rough inclusion function values are needed. In the next section, we show how such values necessary for classification making are estimated on the basis of available partial information about approximation spaces.

3 Quality of Approximation Space

A key task in granular computing is the information granulation process, which is responsible in the formation of information aggregates (patterns) from a set of available data. A methodological and algorithmic issue is the formation of transparent (understandable) information granules, meaning that they should provide a clear and understandable description of patterns held in data. Such fundamental property can be formalized by a set of constraints that must be satisfied during the information granulation process. Usefulness of these constraints is measured by quality of approximation space:

$$Quality_1 : Set\_AS \times P(U) \rightarrow [0,1]$$

where U is a non-empty set of objects and Set_AS is a set of possible approximation spaces with the universe U.

Example 1. If $UPP(AS,X)) \neq \emptyset$ for $AS \in Set\_AS$ and $X \subseteq U$ then

$$Quality_1(AS,X) = \nu_{SRI}(UPP(AS,X), LOW(AS,X)) = \frac{card(LOW(AS,X))}{card(UPP(AS,X))}$$

The value $1 - Quality_1(AS,X)$ expresses the degree of completeness of our knowledge about X, given the approximation space AS.

Example 2. In applications we usually use another quality measures based on the minimal length principle [12,21] where also the description length of approximation is included. Let us denote by $description(AS,X)$ the description length of approximation of X in AS. the description length may be measured, e.g., by the sum of description lengths of algorithms testing membership for neighborhoods used in construction of the lower approximation, the upper approximation, and the boundary region of the set X. Then the quality $Quality_2(AS,X)$ can be defined by

$$Quality_2(AS,X) = g(Quality_1(AS,X), description(AS,X))$$

where g is a relevant function used for fusion of values $Quality_1(AS,X)$ and $description(AS,X)$.

One can consider different optimization problems relative to a given class Set_AS of approximation spaces. For example, for given $X \subseteq U$ and a threshold $t \in [0,1]$ one can search for an approximation space AS satisfying the constraint $Quality(AS,X) \geq t$. Another example can be related to searching for an approximation space satisfying additionally the constraint $Cost(AS) < c$ where $Cost(AS)$ denotes the cost of approximation space AS (e.g. measured by the number of attributes used to define neighborhoods in AS) and c is a given threshold.

In the process of searching for (sub-)optimal approximation spaces different strategies are used. Let us consider one illustrative example. Let $DT = (U, A, d)$ be a decision system (a given sample of data) where U is a set of objects, A is a set of attributes and d is a decision. We assume that for any object x is accessible only a partial information equal to the A-signature of x (object signature, for short), i.e., $Inf_A(x) = \{(a, a(x)) : a \in A\}$ and analogously for any concept there is only given a partial information about this concept by a sample of objects, e.g., in the form of decision table. One can use object signatures as new objects in a new relational structure $\mathcal{R}$. In this relational structure $\mathcal{R}$ are also modeled some relations between object signatures, e.g., defined by the similarities of these object signatures. Discovery of relevant relations on object signatures is an important step in the searching process for relevant approximation spaces. In the next step, we select a language $\mathcal{L}$ of formulas expressing properties over the defined relational structure $\mathcal{R}$ and we search for relevant formulas in $\mathcal{L}$. The semantics of formulas (e.g., with one free variable) from $\mathcal{L}$ are subsets of object signatures. Observe that each object signature defines a neighborhood of objects from a given sample (e.g., decision table DT) and another set on the whole universe of objects being an extension of U. In this way, each formula from $\mathcal{L}$ defines a family of sets of objects over the sample and also another family of sets over the universe of all objects. Such families can be used to define new neighborhoods of a new approximation space, e.g., by taking unions of the described above families. In the searching process for relevant neighborhoods, we use information encoded in the given sample. More relevant neighborhoods are making it possible to define relevant approximation spaces (from the point of view of the optimization criterion). It is worth to mention that often this searching process is even more compound. For example, one can discover several relational structures (not only one, e.g., $\mathcal{R}$ as it was presented before) and formulas over such structures defining different families of neighborhoods from the original approximation space and next fuse them for obtaining one family of neighborhoods or one neighborhood in a new approximation space. Such kind of modeling is typical for hierarchical modeling [4], e.g., when we search for relevant approximation space for objects composed from parts for which some relevant approximation spaces have been already found.

Let us consider some illustrative examples of granule modeling (see Figure 1). Any object $x \in U$, in a given information system $IS_1 = (U, A)$, is perceived by means of its signature $Inf_A(x) = \{(a, a(x)) : a \in A\}$. On the first level, we consider objects with signatures represented by the information system $IS_1 = (U, A)$. Objects with the same signature are indiscernible. On the next level of modeling we consider as objects some relational structures over signatures of objects from the first level. For example, for any signature u one can consider as a relational structure a neighborhood defined by a similarity relation τ between signatures of objects from the first level (see Figure 1). Attributes of objects on the second level describe properties of relational structures. Hence, indiscernibility classes defined by such attributes are sets of relational structures; in our example sets of neighborhoods. We can continue this process of hierarchical modeling by considering as objects on the third level signatures of objects from the second level. In our example, the third level of modeling represents modeling of clusters of neighborhoods defined by the similarity relation τ. Observe that it is possible to link objects from a higher level with objects from a lower level. In our example, any object from the second level is a neighborhood or τ. Any element u' of this neighborhood defines on the first level an elementary granule (indiscernibility class) $\{x \in U : Inf_A(x) = u'\}$. Hence, any neighborhood $\tau(u)$ defines on the first level a family of elementary granules corresponding to signatures from the neighborhood. Now, one can consider as a quality measure for the similarity τ a function assigning to τ a degree to which the union of the elementary granules mentioned above is included into a given concept.

In the second example, we assume that the information system on the first level has a bit more general structure. Namely, on any attribute value set V_a there is defined a relational structure $\mathcal{R}_a$ and a language $\mathcal{L}_a$ of formulas for expressing properties over V_a. For example, one can consider an attribute *time* with values in the set $\mathcal{N}$ of natural numbers, i.e., $V_a \subseteq \mathcal{N}$. The value $time(x)$ is interpreted as a time at which the object x was perceived. The relational structure $\mathcal{R}_{time}$ is defined by (V_a, S), where S is the successor relation in $\mathcal{N}$, i.e., xSy if and only if $y = x + 1$. Then relational structures on the second layer can correspond to windows of a given length T, i.e., structures of the form $(\{u_1, \ldots, u_T\}, S)$ where for some $x_1, \ldots, x_T$ we have $u_i = Inf_A(x_i)$ and $time(x_{i+1}) = time(x_i) + 1$ for $i = 1, \ldots, T$. Hence, the attributes on the second layer of modeling correspond to properties of windows while attributes on the third level could correspond to clusters of windows. Again in looking for relevant clusters we should consider links of the higher levels with lower levels. Another possibility will be to consider some relational structures on the attributes values sets on the second layer. They could allow us to model relations between windows such as overlapping, earlier than. Then, attributes on this level could describe properties of sequences of windows. Such attributes can correspond to some models of processes. Yet another possibility is to use additionally some spatial relations (e.g., nearness) between the successive elements of windows.

For structural objects, it is often used a decomposition method for modeling relational structures on the second level. The object signatures are decomposed into parts and some relations between such parts are considered which are defined over relational structures with the universe $\times_{a \in A} V_a$. One of the methods is based on searching for (i) a decomposition of the object signatures; (ii) tolerance relations defined on parts of object signatures received by decomposition; and (iii) relations over tolerance classes of such tolerance relations (e.g., expressing closeness of classes of parts corresponding to tolerance classes). This method aims to discover relational structures such that it is possible to define over such structures relevant clusters (granules, patterns) of objects for the considered task (e.g., approximation of concepts). The relations over tolerance classes are used for filtering relevant compositions of parts of object signatures defined by tolerance classes. This approach is closely related to constrained sums of information systems [16]. For example, any object of the constrained sum $+_R(IS_1, IS_2)$ of information systems IS_1, IS_2 consists of pairs (x_1, x_2) of objects from IS_1 and IS_2 satisfying some constraints described by $R \subseteq U_1 \times U_2$, i.e., $U = R \cap (U_1 \times U_2)$. The attributes of $+(IS_1, IS_2)$ consist of the attributes of IS_1 and IS_2, except that if there are any attributes in common, then we make their distinct copies, to avoid confusion.

It is worthwhile mentioning that in searching under uncertainty for relevant granules it is also necessary to use methods for estimation if the discovered patterns on (training) samples of objects are relevant on the whole universe of objects.

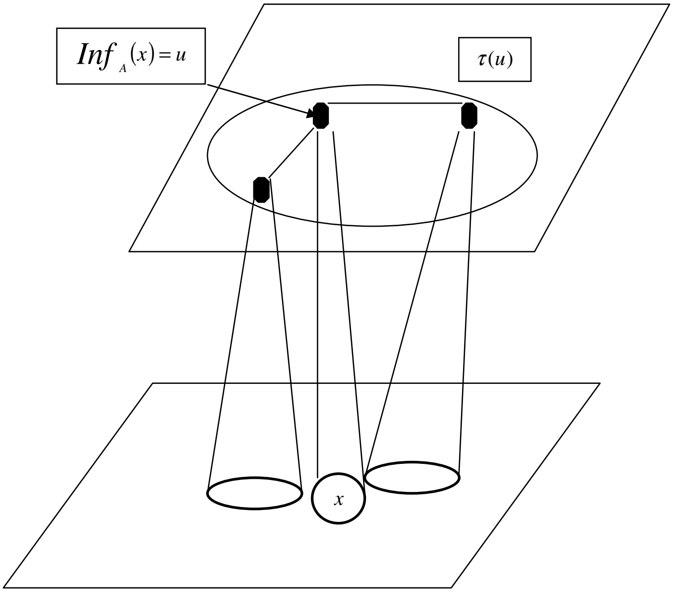


Fig. 1. Modeling of granules

The above examples are typical for granular computing where for a given task it is necessary to search for granules in a given granular system which are satisfying some optimization criteria. The discussed methods are used in spatio-temporal reasoning (see, e.g, [17]), in behavioral pattern identification and planning (see, e.g., [4,3]). There are some other basic concepts which should be considered in granular computing. One of them is related to risk. In the following section we present some remarks about risk in construction of granules.

4 Risk and Utility Functions in Construction of Granules

There is a large literature on relationships between decision making and risk. In this section, we discuss some problems related to risk in granular computing. An example of risk analysis (based on rough sets) for medical data the reader can find in [5].

First we recall the definition of *granule system*. Any such system GS consists of a set of granules G. Moreover, a family of relations with the intended meaning *to be a part to a degree* between granules is distinguished. The degree structure is described by a relation *to be an exact part*. More formally, a granule system is any tuple

$$GS = (G, H, <, \{\nu_p\}_{p \in H}, size) \tag{1}$$

where G is a non-empty set of granules. H is a non-empty set of granule inclusion degrees with a binary relation $<$ (usually a strict partial order) which defines on H a structure used to compare the degrees. $\nu_p \subseteq G \times G$ is a binary relation *to be a part to a degree at least p* between granules from G, called *rough inclusion*. $size : G \longrightarrow R_+$ is the granule size function, where R_+ is the set of nonnegative reals.

In constructing of granule systems it is necessary to give a constructive definition of all their components. In particular, one should specify how more compound granules are defined from already defined granules or given elementary granules. Usually, the set of granules is defined as the least set generated from distinguished elementary granules (e.g., defined by indiscernibility classes) by some operations on the granules. These operations are making it possible to fuse elementary granules for obtaining new granules relevant for the task to be solved. In the literature many different operations on granules are reported (see, e.g., [15]) from those defined by boolean combination of descriptors to compound classifiers or networks of classifiers.

Let us consider, a task of searching in the set of granules of a granule system GS for a granule g satisfying a given constraint to a satisfactory degree, e.g., $\nu_{tr}(g, g_0)$, where $\nu : G \times G \longrightarrow [0, 1]$ is the inclusion function, $\nu_{tr}(g, g_0)$ means that $\nu(g, g_0) \geq tr$, g_0 is a given granule and tr is a given threshold. Let g^* be a solution, i.e., g^* satisfies the condition

$$\nu(g^*, g_0) > tr. \tag{2}$$

Risk analysis is a well established notion in decision theory [6]. We would like illustrate the importance of risk analysis in GC.

A typical risk analysis task in GC can be described as follows. For a granule g^* is constructed a granule $N(g^*)$, i.e. representing a cluster of granules defined by g^* received by changing some parameters of g^* such as attribute values used in the g^* description. We would like to estimate how this changes influence the condition (2).

First, let us assume that $\nu(g^*, g_0) = \nu_{SRI}(\|g^*\|, \|g_0\|)$, where $\|\cdot\|$ denotes the semantic of granule, i.e., a function $\|\cdot\| : G \longrightarrow P(U)$ for a given universe of objects U and ν_{SRI} is the standard rough inclusion function. Then, one can take $\delta^* = \arg\min_{\delta \in [0,tr]}(\nu(N(g^*), g_0) \geq tr - \delta)$. The value δ^* can be treated as a *risk degree* of changing the inclusion degree in g_0 when the granule g^* is substituted by $N(g^*)$.

One can consider a hierarchy of granules over g^* defined by an ascending sequence $N_1(g^*), \ldots, N_k(g^*)$, i.e., $\|N_1(g^*)\| \subseteq \ldots \subseteq \|N_k(g^*)\|$ and corresponding risk degrees $\delta_1^* \leq \ldots \delta_k^*$. For example, if δ_1^* is sufficiently small than g^* is called *robust* with respect to deviations caused by taking $N_1(g^*)$ instead of g^*. However, when i is increasing then taking $N_i(g^*)$ instead of g^* gradually increases the risk degree. The above example illustrates importance of risk analysis in GC. Information maps introduced in [18] can be used for risk analysis.

Let us now move to the concept of *utility function* over granules. The concept of utility function has been intensively studied in decision theory or game theory [8,7]. We would like to present an illustrative example showing that such functions are important for granule systems.

We assume two granule systems GS and GS_0 with granule sets G and G_0 are given. We consider two properties of granules in this systems, i.e., $P \subseteq G$ and $P_0 \subseteq G_0$ Moreover, we assume that checking the membersip for P is much simpler than for P_0 (e.g., because granules from G_0 are much simpler than granules from G). This means that there are given algorithms $\mathcal{A}$, $\mathcal{A}_0$ for checking the membership in P and P_0, respectively, and the complexity of $\mathcal{A}_0$ is much lower than the complexity of the algorithm $\mathcal{A}$. Under the above assumptions it is useful to search for a *utility function* $Utility : G \longrightarrow G_0$ reducing the membership problem for P to the membership problem for P_0, i.e., a function with the following property: $g \in P$ if and only if $Utility(g) \in P_0$. Construction of the utility function satisfying the above condition may be not feasible. However, it becomes often feasible when we relax the binary membership relation $\in$ to the membership at least to a given degree (see, e.g., [20]). This example illustrates, the important property of utility functions. Usually, G_0 is a set of scalar values or it is assumed that some preference relation over G_0 is given.

Finally, we would like to add that in GC it is necessary to develop methods searching for approximation of risk degrees and utility function from data and domain knowledge analogously to approximation of complex concepts (see, e.g., [4]).

5 Conclusions

We have discussed the role of approximation spaces in construction of granules satisfying criteria expressed by the minimal length principle. The role of risk measures and utility functions in GC was illustrated. In our system searching for adaptive approximation of complex concepts, we plan to implement strategies based on the minimal length principle in GC, risk measures in GC, and utility functions in GC. This will also require developing methods for approximation of risk measures and utility functions.

Acknowledgements. The research of Andrzej Skowron was supported by the grant from Ministry of Scientific Research and Information Technology of the Republic of Poland. The research of Jaroslaw Stepaniuk was supported by the Białystok University of Technology Rector's Grant.

References

1. Alpigini, J.J., Peters, J.F., Skowron, A., Zhong, N. (eds.): RSCTC 2002. LNCS (LNAI), vol. 2475, pp. 14–16. Springer, Heidelberg (2002)
2. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Dordrecht (2003)

3. Bazan, J.G., Kruczek, P., Bazan-Socha, S., Skowron, A., Pietrzyk, J.J.: Automatic planning of treatment of infants with respiratory failure through rough set modeling. In: RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 418–427. Springer, Heidelberg (2006)
4. Bazan, J., Peters, J.F., Skowron, A.: Behavioral pattern identification through rough set modelling. In: Ślęzak, D., Wang, G., Szczuka, M., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 688–697. Springer, Heidelberg (2005)
5. Bazan, J., Osmólski, A., Skowron, A., Ślęzak, D., Szczuka, M., Wróblewski, J.: Rough set approach to the survival analysis. In: [1], pp. 522–529
6. Byrd, D.M., Cothern, C.R.: Introduction to Risk Analysis: A Systematic Approach to Science-Based Decision Making. ABS Group, Rockville, MD (2000)
7. Fishburn, P.C.: Utility Theory for Decision Making, Robert E. Krieger Publishing Co, Huntington, NY (1970)
8. Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, Princeton, NJ, 1944 sec.ed (1947)
9. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1) 3–27 (2007), Rough sets: Some extensions. Information Sciences 177(1) 28–40 (2007), Rough sets and Boolean reasoning. Information Sciences 177(1) 41–73 (2007)
10. Pal, S.K., Polkowski, L., Skowron, A. (eds.): Rough-Neural Computing: Techniques for Computing with Words. Springer, Berlin (2004)
11. Pawlak, Z.: Rough Sets. In: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht (1991)
12. Rissanen, J.: Minimum-description-length principle. In: Kotz, S., Johnson, N. (eds.) Encyclopedia of Statistical Sciences, pp. 523–527. John Wiley & Sons, New York (1985)
13. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
14. Skowron, A., Stepaniuk, J.: Information granules: Towards foundations of granular omputing. International Journal of Intelligent Systems 16(1), 57–86 (2001)
15. Skowron, A., Stepaniuk, J.: Information granules and rough-neural computing. In: [15], pp. 43–84.
16. Skowron, A., Stepaniuk, J.: Constrained sums of information systems. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 300–309. Springer, Heidelberg (2004)
17. Skowron, A., Synak, P.: Complex patterns. Fundamenta Informaticae 60(1-4), 351–366 (2004)
18. Skowron, A., Synak, P.: Reasoning in information maps. Fundamenta Informaticae 59(2-3), 241–259 (2004)
19. Stepaniuk, J.: Knowledge discovery by application of rough set models. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.) Rough Set Methods and Applications. New Developments in Knowledge Discovery in Information Systems, pp. 137–233. Physica–Verlag, Heidelberg (2000)
20. Stepaniuk, J., Skowron, A., Peters, J., Swiniarski, R.: Calculi of approximation spaces. Fundamenta Informaticae 72(1-3), 363–378 (2006)
21. Ślęzak, D.: Approximate entropy reducts. Fundamenta Informaticae 53(3-4), 365–390 (2002)
22. Zadeh, L.A.: A new direction in AI: Toward a computational theory of perceptions. AI Magazine 22(1), 73–84 (2001)

Attribute Core Computation Based on Divide and Conquer Method*

Feng Hu[1,2], Guoyin Wang[1,2], and Ying Xia[1,2]

[1] School of Information Science and Technology,
Southwest Jiaotong University,
Chengdu 600031, P.R. China
[2] Institute of Computer Science and Technology,
Chongqing University of Posts and Telecommunications,
Chongqing 400065, P.R. China
{hufeng, wanggy, xiaying}@cqupt.edu.cn

Abstract. The idea of divide and conquer method is used in developing algorithms of rough set theory. In this paper, according to the partitions of equivalence relations on attributes of decision tables, two novel algorithms for computing attribute core based on divide and conquer method are proposed. Firstly, a new algorithm for computing the positive region of a decision table is proposed, and its time complexity is $O(|U| \times |C|)$, where, $|U|$ is the size of the set of objects and C is the size of the set of attributes. Secondly, a new algorithm for computing the attribute core of a decision table is developed, and its time complexity is $O(|U| \times |C|^2)$. Both these two algorithms are linear with $|U|$. Simulation experiment results show that the algorithm of computing attribute core is not only efficient, but also adapt to huge data sets.

Keywords: Rough set, divide and conquer, positive region, attribute core.

1 Introduction

Rough set (RS) is a valid mathematical theory to deal with imprecise, uncertain, and vague information [1]. It has been applied in many fields such as machine learning, data mining, intelligent data analyzing and control algorithm acquiring successfully since it was proposed by Pawlak in 1982 [2].

In divide and conquer method, a problem which is hard to be solved directly is divided into many sub-problems and conquered respectively. The structures of the sub-problems are similar to the one of the original problem except their sizes are smaller. The divide and conquer method divide a problem into simpler

* This paper is partially supported by National Natural Science Foundation of China under Grants No.60373111 and No.60573068, Program for New Century Excellent Talents in University (NCET), Natural Science Foundation of Chongqing under Grant No.2005BA2003, Science & Technology Research Program of Chongqing Education Commission under Grant No.KJ060517.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 310–319, 2007.

sub-problems iteratively in this way and the sizes of the sub-problems will be reduced to be easy enough to be processed directly [3,4].

In the study of rough set theory, the computation of positive region and attribute core are two basic operations, and their efficiencies will affect the efficiencies of further attribute reduction and value reduction. Many attribute reduction algorithms have already been proposed [5-16]. However, few of them can deal with huge data sets well. Combining the idea of divide and conquer method and partition of equivalence relation in a decision table, a huge data set can be divided into many small ones that can be processed directly easily. In addition, the complexity of the original problem could be reduced. According to the above analysis, a new algorithm for computing positive region based on divide and conquer method is proposed, and its time complexity is $O(|U| \times |C|)$. Furthermore, a new algorithm for computing attribute core based on divide and conquer is also proposed, and its time complexity is $O(|U| \times |C|^2)$.

The rest of this paper is organized as follows. In section 2, some basic notions about rough set theory are introduced. A new algorithm for computing positive region is proposed in section 3. In section 4, a novel algorithm for computing attribute core based on divide and conquer is proposed. In section 5, some experiment results are discussed. We draw some conclusions in section 6.

2 Basic Notions of Rough Set Theory

For the convenience of illustration, some basic notions of rough set theory are introduced here at first.

Def. 1 (decision table [2]) A decision table is defined as $S =< U, A, V, f >$, where U is a non-empty finite set of objects, called universe, R is a non-empty finite set of attributes, $A = C \cup D$, where C is the set of condition attributes and D is the set of decision attributes, $D \neq \emptyset$. $V = \bigcup\limits_{p \in R} V_p$, and V_p is the domain of the attribute p. $f : U \times A \rightarrow V$ is a total function such that $f(x_i, A) \in V_p$ for every $p \in A, x_i \in U$.

Def. 2 (indiscernibility relation [2]) Given a decision table $S =< U, A = C \cup D, V, f >$, each subset $B \subseteq C$ of attribute determines an indiscernibility relation *IND(B)* as follows: $IND(B) = \{(x, y)|(x, y) \in U \times U, \forall b \in B(b(x) = b(y))\}$.

Def. 3 (lower-approximation, upper-approximation and border region [2]) Given a decision table $S =< U, C \cup D, V, f >$, for any subset $X \subseteq U$and the indiscernibility relation $IND(B)$, the B lower-approximation, upper-approximation and border region of X are defined as: $B\_(X) = \bigcup\limits_{Y_i \in U/IND(B) \wedge Y_i \subseteq X} Y_i$, $B^-(X) = \bigcup\limits_{Y_i \in U/IND(B) \wedge Y_i \cap X \neq \Phi} Y_i$, $BN(X) = B^-(X) - B_-(X)$.

Def. 4 (positive region [2]) Given a decision table $S =< U, A, V, f >$. $P \subseteq A$ and $Q \subseteq A$, the P positive region of Q is defined as: $Pos_P(Q) = \bigcup\limits_{X \in U/Q} P\_(X)$.

Def. 5 (relative core [2]) Given a decision table $S =< U, A, V, f >$, $P \subseteq A$, $Q \subseteq A$, and $r \in P$. r is unnecessary in P with reference to Q if and only if

$Pos_P(Q) = Pos_{P-\{r\}}(Q)$, otherwise r is unnecessary in P with reference to Q. The core of P with reference to Q is defined as: $CORE_Q(P) = \{r|r \in P$, r is necessary in P with reference to $Q\}$. Attribute r is necessary in P with reference to Q can be written as r is relative necessary, too.

Def. 6 [2] Given a decision table $S =< U, A, V, f >$, $P \subseteq A$, $Q \subseteq A$. $\forall r \in P$, if r is necessary in P with reference to Q, we call P is independent with reference to Q.

Def. 7 (relative reduction [2]) Given a decision table $S =< U, A, V, f >$, $P \subseteq A$, $Q \subseteq A$. $Red \subset P$, if Red is independent with reference to Q and $Pos_{Red}(Q) = Pos_P(Q)$, we call Red is a reduction of P with reference to Q.

In this paper, $f(x, c)(x \in U \wedge c \in C)$ is noted as $c(x)$, and $f(x, d)(x \in U \wedge D = \{d\})$ is noted as $d(x)$.

3 Algorithm for Computing Positive Region Based on Divide and Conquer Method

In [17], a method for computing positive region is proposed by partitioning the universe of a decision table. In this paper, by reducing condition attributes and partitioning the universe of a decision table, the original decision table could be divided into many new decision tables with different attribute spaces. The method is as follows.

Theorem 1. Given a decision table $S =< U, A = C \cup D, V, f >$. $\forall c(c \in C)$, $U/\{c\}$ is a partition of S, that is, S is divided into $k(k = |IND(U/\{c\})|)$ sub-decision tables S_1, S_2,..., S_k, where, $S_k =< U_k, (C-\{c\}) \cup D, V_k, f_k >$, satisfying $_{\forall x \in U_i \forall y \in U_i} c(x) = c(y)(1 \leq i \leq k)$ and $_{\forall x \in U_i \forall z \in U_j} c(x) \neq c(z)(1 \leq i < j \leq k)$. Let $R = C - \{c\}$, $Pos^i_R(D)(1 \leq i \leq k)$ be the positive region of a sub decision table S_i, $Pos_C(D)$ be the positive region of the original decision table S. Then, $Pos_C(D) = \bigcup\limits_{1 \leq i \leq k} Pos^i_R(D)$.

Proof: firstly, prove $Pos_C(D) \subseteq \bigcup\limits_{1 \leq i \leq k} Pos^i_R(D)$.

$\forall x \in Pos_C(D)$, suppose x is assigned to sub-decision table S_k, that is, $x \in U_k$. Now, we need to prove $x \in Pos^i_R(D)$. Prove to the reverse.

Suppose $x \notin Pos^i_R(D)$, then $\exists y \in U_j(_{\forall a \in C-\{c\}}(a(x) = a(y)) \wedge (d(x) \neq d(y)))$. Since $c(x) = c(y)$, so $(_{\forall a \in C}(a(x) = a(y)) \wedge (d(x) \neq d(y)))$, that is, $x \notin Pos_C(D)$, which is conflict with the premise $x \in Pos_C(D)$. Therefore, $x \in Pos^i_R(D)$, then $Pos_C(D) \subseteq \bigcup\limits_{1 \leq i \leq k} Pos^i_R(D)$. That's to say, $Pos_C(D) \subseteq \bigcup\limits_{1 \leq i \leq k} Pos^i_R(D)$.

Then, prove $\bigcup\limits_{1 \leq i \leq k} Pos^i_R(D) \subseteq Pos_C(D)$.

$\forall x \in Pos^i_{C-\{c\}}(D)(1 \leq i \leq k)$, $\forall y \in U$, if $y \notin U_i$, there is $c(x) \neq c(y)$. So, $x \in Pos_C(D)$. That's to say, $\bigcup\limits_{1 \leq i \leq k} Pos^i_R(D) \subseteq Pos_C(D)$.

Therefore, $Pos_C(D) = \bigcup\limits_{1 \leq i \leq k} Pos^i_R(D)$. Theorem 1 holds.

With Theorem 1, we could develop an algorithm for computing positive region based on divide and conquer.

```
Algorithm 1. Computing Positive Region Based on Divide and Conquer Method
Input:  A decision table $S =< U, C \cup D, V, f >$
Output: Positive region $Pos_C(D)$
Step1: (Initiative) $Pos_C(D) = \phi$;
Step2: (Compute positive region by invoking recursive function)
    $Get\_Positive(U, 1)$;
Step3: (Return) return $Pos_C(D)$
Recursive Function $Get\_Positive$(Set $OSet$, int $k$)
  if $(k < 1)$ or $(|OSet| < 1)$ then return; end if
  if $(|OSet| = 1)$ then
    $Pos_C(D) = Pos_C(D) \cup OSet$; return;
  end if
  if $(k > |C|)$ then
    if $_{\forall x \in OSet \forall y \in OSet} d(x) = d(y)$ then $Pos_C(D) = Pos_C(D) \cup OSet$; end if
    return;
  end if
  Let $c = c_k$, $V^c = \phi$;
  for $i = 1$ to $|OSet|$ do
    $V^c = V^c \cup f(x_i, c)$;
  end for
  for $i = 1$ to $|V^c|$ do
    $OSet_j^c = \phi$;
  end for
  construct a mapping function $f^{'} : V^c \rightarrow N(N = 1, 2, ..., |V^c|)$, satisfying:
  $_{\forall x \in V^c \forall y \in V^c} (f^{'}(x) = f^{'}(y)) \Leftrightarrow (x = y)$.
  for $i = 1$ to $|OSet|$ do
    let $j = f^{'}(f(x_i, c))$; $OSet_j^c = OSet_j^c \cup \{x_i\}$;
  end for
  for $j = 1$ to $|V^c|$ do
    recursive invoking: $Get\_Positive(OSet_j^c, k + 1)$
  end for
End Function
```

Let's analyze the time complexity and space complexity of Algorithm 1 now.

Suppose $n = |U|$, $m = |C|$, $p = max(|V_i|)(1 \leq i \leq |C|)$. Because calculating all values of k-th attribute in the set of objects $OSet$ can be performed in the time $O(n)$, the time complexity of Algorithm 1 could be approximated by the following recursive equation:

$$T(n,m) = \begin{cases} 1. \quad (n = 1) \\ n. \quad (m = 0) \\ 2n + p_1 + T(n_1, m-1) + T(n_2, m-1) + ... + T(n_k, m-1). \\ \qquad (n_1 + n_2 + ... + n_k = n, n > 1, m > 0, p_1 \leq min(p, n)) \\ 0. \quad (else) \end{cases} \quad (1)$$

According to the iterative method and solution of recursive equation [3], we can find:

$$\begin{aligned}
T(n,m) &\leq (2n+n) + T(n_1,m-1) + T(n_2,m-1) + ... + T(n_k,m-1)\\
&\leq 3n + T(n_1,m-1) + T(n_2,m-1) + ... + T(n_k,m-1)\\
&\leq 3n + \ (3n_1 + T(n_1^1,m-2) + T({}^1_2,m-2) + ... + T({}^1_{t_1},m-2))\\
&\quad +(3n_2 + T(n_1^2,m-2) + T({}^2_2,m-2) + ... + T({}^2_{t_2},m-2))\\
&\quad +...\\
&\quad +(3n_k + T(n_1^k,m-2) + T({}^k_2,m-2) + ... + T({}^k_{t_k},m-2))\\
&\leq 3n+3n_1+3n_2+...+3n_k+ \ (T(n_1^1,m-2)+T({}^1_2,m-2)+...+T({}^1_{t_1},m-2))\\
&\quad +(T(n_1^2,m-2) + T({}^2_2,m-2) + ... + T({}^2_{t_2},m-2))\\
&\quad +...\\
&\quad +(T(n_1^k,m-2) + T({}^k_2,m-2) + ... + T({}^k_{t_k},m-2))\\
&\leq 3n + 3n+ \ (T(n_1^1,m-2) + T({}^1_2,m-2) + ... + T({}^1_{t_1},m-2))\\
&\quad +(T(n_1^2,m-2) + T({}^2_2,m-2) + ... + T({}^2_{t_2},m-2))\\
&\quad +...\\
&\quad +(T(n_1^k,m-2) + T({}^k_2,m-2) + ... + T({}^k_{t_k},m-2))\\
&\leq 3n + 3n + ... + 3n + n\\
&\leq 3 \times m \times n + n
\end{aligned}$$

That is, $T(n,m) = O(n \times m)$.

Suppose $n = |U|$, $m = |C|$, $p = max(|V_i|)(1 \leq i \leq |C|)$. Then, the space complexity of Algorithm 1 is: $O(n + p \times m)$.

4 Algorithm for Computing Attribute Core Based on Divide and Conquer Method

Lemma 1. Given a decision table $S =< U, A = C \cup D, V, f >$. $\forall c(c \in C)$, $U/\{c\}$ is a partition of S, that is, S is divided into $k(k = |IND(U/\{c\})|)$ sub-decision tables S_1, S_2,..., S_k. Where, $S_k =< U_k, (C-\{c\}) \cup D, V_k, f_k >$, satisfying $_{\forall x \in U_i \forall y \in U_i} c(x) = c(y)(1 \leq i \leq k)$ and $_{\forall x \in U_i \forall z \in U_j} c(x) \neq c(z)(1 \leq i < j \leq k)$. Suppose $Core_i(1 \leq i \leq k)$ be the attribute core of the sub-decision table S_i, and $Core$ be the attribute core of the decision table S. Then, $_{\forall a \in Core_i}\ a \in Core$.

Lemma 2. Given a decision table $S =< U, A = C \cup D, V, f >$. $\forall c(c \in C)$, which is unnecessary in C with reference to D, that is, $Pos_{C-\{c\}}(D) = Pos_C(D)$. $U/\{c\}$ is a partition of S, that is, S is divided into $k(k = |IND(U/\{c\})|)$ sub-decision tables S_1, S_2,..., S_k. Where, $S_k =< U_k, (C-\{c\}) \cup D, V_k, f_k >$, satisfying $_{\forall x \in U_i \forall y \in U_i} c(x) = c(y)(1 \leq i \leq k)$ and $_{\forall x \in U_i \forall z \in U_j} c(x) \neq c(z)(1 \leq i < j \leq k)$. Suppose $Core_i(1 \leq i \leq k)$ be the attribute core of the sub-decision table S_i, and $Core$ be the attribute core of the decision table S. Suppose $red_i(1 \leq i \leq k)$ be an attribute reduction of the sub-decision table S_i. Let $R = \bigcup\limits_{1 \leq i \leq k} red_i$. Then, there are two conclusions:

(1) $Core = \bigcup\limits_{1 \leq i \leq k} Core_i$. (2)In the decision table S, $Pos_R(D) = Pos_C(D)$.

Lemma 3. Given a decision table $S =< U, A = C \cup D, V, f >$. Let $Core(Core \neq \phi)$ be the attribute core of S. $\forall c(c \in Core)$, which is a core attribute (necessary attribute) of S, that is, $Pos_{C-\{c\}}(D) \neq Pos_C(D)$. $U/\{c\}$ is a partition of S, that is, S is divided into $k(k = |IND(U/\{c\})|)$ sub decision tables S_1, S_2,..., S_k. Where, $S_k =< U_k, (C - \{c\}) \cup D, V_k, f_k >$, satisfying $\forall_{x \in U_i} \forall_{y \in U_i} c(x) = c(y)(1 \leq i \leq k)$ and $\forall_{x \in U_i} \forall_{z \in U_j} c(x) \neq c(z)(1 \leq i < j \leq k)$. Suppose $Core_i(1 \leq i \leq k)$ be the attribute core of the sub-decision table S_i, and $red_i(1 \leq i \leq k)$ be an attribute reduction of the sub-decision table S_i. Let $R = \{c\} \cup \bigcup_{1 \leq i \leq k} red_i$. Then, there are two conclusions:

(1) $Core = \{c\} \cup \bigcup_{1 \leq i \leq k} Core_i$. (2)In the decision table S, $Pos_R(D) = Pos_C(D)$.

Theorem 2. Given a decision table $S =< U, A = C \cup D, V, f >$. $\forall c(c \in C)$, according to $U/\{c\}$, S is divided into $k(k = |IND(U/\{c\})|)$ sub-decision tables S_1, S_2,..., S_k. Where, $S_k =< U_k, (C - \{c\}) \cup D, V_k, f_k >$, satisfying $\forall_{x \in U_i} \forall_{y \in U_i} c(x) = c(y)(1 \leq i \leq k)$ and $\forall_{x \in U_i} \forall_{z \in U_j} c(x) \neq c(z)(1 \leq i < j \leq k)$. Suppose $Core_i(1 \leq i \leq k)$ be the attribute core of the sub decision table S_i, and $Core$ be the attribute core of the decision table S. Then, $\bigcup_{1 \leq i \leq k} Core_i \subseteq Core \subseteq \{c\} \cup \bigcup_{1 \leq i \leq k} Core_i$.

Proof: Obviously, Lemma 1, Lemma 2, Lemma 3 and Theorem 2 could be proved using basic concerts of rough set theory. We omit their proofs here due to page limits.

According to Theorem 2, an algorithm for computing attribute core based on divide and conquer could be developed.

Algorithm 2. Computing Attribute Core Based on Divide and Conquer Method

```
Input:   A decision table $S =< U, C \cup D, V, f >$
Output: Attribute Core ($Core$) of $S$
Step1: (*Initiative*) $Core = \phi$;
Step2: (*Compute Attribute Core using recursive function*)
    $Get\_Core(U, 1)$;
Step3: (*Return*) return $Core$
Recursive Function $Get\_Core$(Set $OSet$, int $k$)
  if ($k < 1$) or ($|OSet| < 1$) then return; end if
  if ($c_k \in Core$) then return;
  else
    Suppose $C^k = c_k \cup c_{k+1} \cup ... \cup c_{|C|}$;
    For decision table $S' =< OSet, C^k \cup D, V^k, f^k >$, compute positive
    region $Pos_{C^k-\{c_k\}}(D)$ using Algorithm 1;
    $Pos_{C^k}(D) = \phi$;
  end if
  Let $c = c_k$, $V^c = \phi$;
  for  $i = 1$  to  $|OSet|$  do
```

```
        $V^c = V^c \cup f(x_i, c)$;
    end for
    for $i = 1$ to $|V^c|$ do
        $OSet_j^c = \phi$;
    end for
    construct a mapping function $f^{'} : V^c \rightarrow N(N = 1, 2, ..., |V^c|)$, satisfying:
    $\forall x \in V^c \forall y \in V^c \; (f^{'}(x) = f^{'}(y)) \Leftrightarrow (x = y)$.
    for $i = 1$ to $|OSet|$ do
        let $j = f^{'}(f(x_i, c))$; $OSet_j^c = OSet_j^c \cup \{x_i\}$;
    end for
    for $j = 1$ to $|V^c|$ do
        $Pos_{C^k}(D) = Pos_{C^k}(D) \cup$ $Get\_Positive(OSet_j^c, k+1)$;
        $Get\_Core(OSet_j^c, k+1)$;
    end for
    if $(Pos_{C^k - \{c\}}(D) < Pos_{C^k}(D))$ then $Core = Core \cup \{c\}$; end if
End Function
```

Now, let's analyze the time complexity and space complexity of Algorithm 2.

Suppose $n = |U|$, $m = |C|$. Then, the time complexity of Algorithm 2 could be approximated by the following recursive equation:

$$T(n,m) = \begin{cases} O(n \times m) + T(n_1, m-1) + T(n_2, m-1) + ... + T(n_k, m-1). \\ \qquad (n_1 + n_2 + ... + n_k = n, n > 1, m > 0) \\ 0. \quad (else) \end{cases} \tag{2}$$

According to the iterative method and solution of recursive equation [3], we can have: $T(n,m) = O(n \times m) \times m = O(n \times m^2)$.

Suppose $n = |U|$, $m = |C|$, $p = max(|V_i|)(1 \leq i \leq |C|)$. Then, the space complexity of Algorithm 2 is: $O(n + p \times m)$.

5 Experiment Results

Firstly, some data sets from *UCI* database are used to test Algorithms 2. Secondly, data sets KDDCUP99 are used to test the efficiency of Algorithm 2(Data sets KDDCUP99 can be downloaded at *http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html*).

5.1 Experiment Results in *UCI* Database

Data sets $Heart\_c\_ls$, $Pima\_India$, rx_bq_ls, $Liver\_disorder$ and $Abalone$ from *UCI* database (These data sets can be downloaded at http://www.ics.uci.edu) are used as test data sets. In order to compare our algorithms with existed algorithms, the algorithm in [5,7,18] and the algorithm in [19] are chosen, called Algorithm a and Algorithm b respectively. The experiment results are shown in Table 1. Where, T is running time(in second) of algorithms, and N is the cardinality of core attribute. The configuration of the PC here is P4 2.60G CPU, 256M RAM, Windows XP.

We can find from Table 1 that results of Algorithm a, Algorithm b and Algorithm 2 are valid. However, the Algorithm 2 could save some time.

Table 1. Experiment results on *UCI* database

Data Sets	*Number of Attribute*	*Number of Records*	*Algorithm a*		*Algorithm b*		*Algorithm* 2	
			T	*N*	*T*	*N*	*T*	*N*
Glass	9	214	0.016	9	0.003	9	0.001	9
Heart_c_ls	13	303	0.047	9	0.006	9	0.003	9
Australian_Credit	14	660	0.141	8	0.023	8	0.005	8
Pima_India	8	738	0.156	5	0.025	5	0.003	5
Liver_disorder	6	1260	0.063	5	0.009	5	0.005	5
Abalone	8	4177	8.031	6	1.147	6	0.041	6

5.2 Experiment Results on Data Sets KDDCUP99

In order to test the efficiency of Algorithm 2 on really huge data sets, 20 KDDCUP99 data sets are downloaded. The number of records of these data sets are 1×10^5, 2×10^5, 3×10^5,..., 20×10^5 respectively. The number of condition attributes is 41. The experiment results are shown in Fig.1. The configuration of the PC here is also P4 2.60G CPU, 256M RAM, windows XP.

We can find from Fig.1 that the efficiency of Algorithm 2 is very high on huge data sets. Besides, the time cost of our algorithm is almost linear with the number of objects. In the meantime, we test the minimum data set of Fig.1 with Algorithm *a* and Algorithm *b*, their running time are both more than 1 hour.

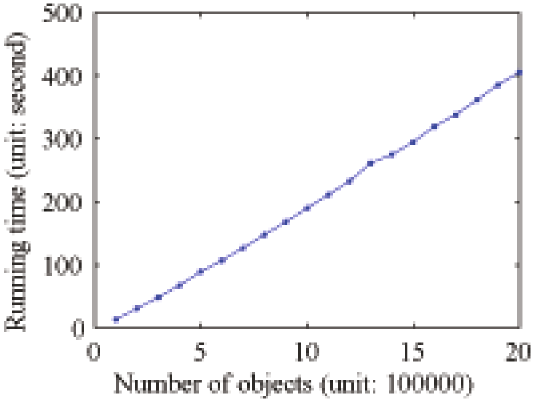


Fig. 1. Experiment results on KDD data sets

6 Conclusion

Though rough set theory is becoming more and more mature, its application in industry is still limited. An important reason is that the efficiency of many algorithms of rough set theory is too low to meet to the need of industry in huge data set environments. In this paper, the idea of divide and conquer method is used in the rough set theory, and an algorithm for computing positive region and an algorithm for computing attribute core are proposed. Experiment results show that the proposed algorithms are not only efficient, but also can deal with huge data sets. Studying on algorithms of attribute reduction and value reduction based on divide and conquer method will be our further work.

References

1. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
2. Wang, G.Y.: Rough set theory and knowledge acquisition (in Chinese). Xi'an Jiaotong University Press, Xi'an (2001)
3. Fu, Q.X, Wang, X.D.: Algorithms and data structure (in Chinese). Publishing House of Electronics Industry, Beijing (2003)
4. Yu, X.X, Cui, G.H, Zhou, H.M.: Fundmental of Computer Algorithms (in Chinese). Huazhong University Press, Wuhan (2001)
5. Skowron, A., Rauszer, C.: The discernibility functions matrics and functions in information systems. In: Slowinski, R. (ed.) Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory, pp. 331–362. Kluwer Academic Publisher, Dordrecht (1992)
6. Bazan, J., Skowron, A., Synak, P.: Dynamic reductions as tool for exacting laws for decision table. Lecture Note in Artificial Intelligence, vol. 869, pp. 346–355. Springer, Berlin (1994)
7. Hu, X.H., Cercone, N.: Learning in relational database: A rough set approach. International Journal of Computional Intelligence 11(2), 323–338 (1995)
8. Jelonek, J., et al.: Rough set reduction of attributes and their domains for neural networks. Computional Intelligence 11(2), 338–347 (1995)
9. Ziarko, W., Shan, N.: Data-based acquisition and incremental modification classfication rules. Computational Intelligence 11(2), 357–370 (1995)
10. Nguyen, H.S, Nguyen, S.H.: Some efficient algorithms for rough set methods. In: Proceedings of the Sixth International Conference, Information Procesing and Management of Uncertainty in Knowledge-Based Systems(IPMU"96), July 1-5, Granada, Spain, vol. 2, pp. 1451–1456 (1996)
11. Susmaga, R.: Experiments in incremental computation of reducts. In: Skowron, A., Olkowski, A. (eds.) Rough Sets in Data Mining and Knowledge Discovery, Springer, Berlin (1998)
12. Bazan, B.J H.S., Nguyen, S.H., Nguyen, P.: Rough set algorithms in classification problem. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.) Rough Set Methods and Applications, pp. 49–88. Physica-Verlag, Heidelberg (2000)
13. Ye, D.Y., et al.: An improvement to Jelonek's attribute reduction algorithm (in Chinese). Acta. Electronica Sinica 28(12), 81–82 (2000)
14. Wang, J., Wang, J.: Reduction algorithms based on discernibility matrix: the ordered attributed method. Journal of Computer Science and Technology 11(6), 489–504 (2001)
15. Liu, S.H., Sheng, Q.J., Wu, B., et al.: Research on efficient algorithms for Rough set methods (in Chinese). Chinese Journal of Computers 30(7), 1086–1088 (2002)
16. Wang, G.Y., Yu, H., Yang, D.C.: Decision table reduction based on conditional information entropy. Chinese Journal of computer (in Chinese) 25(7), 759–766 (2002)

17. Liu, S.H., Cheng, Q.J., Shi, Z.Z.: A new method for fast computing positve region. Journal of Computer Research and Development (in Chinese) 40(5), 637–642 (2003)
18. Ye, D.Y., Chen, Z.J.: A new discernibility matrix and the computation of a core. Acta. Electronica Sinica (in Chinese) 30(7), 1086–1088 (2002)
19. Wang, G.Y.: The computation method of core attribute in decision table. Chinese Journal of Computer (in Chinese) 26(5), 611–615 (2003)

Fast Discovery of Minimal Sets of Attributes Functionally Determining a Decision Attribute*

Marzena Kryszkiewicz and Piotr Lasek

Institute of Computer Science, Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
{mkr,p.lasek}@ii.pw.edu.pl

Abstract. In our paper, we offer an efficient *Fun* algorithm for discovering minimal sets of conditional attributes functionally determining a given dependent attribute, and in particular, for discovering Rough Sets certain, generalized decision, and membership distribution reducts. *Fun* can operate either on partitions or alternatively on stripped partitions that do not store singleton groups. It is capable of using functional dependencies occurring among conditional attributes for pruning candidate dependencies. The experimental results show that all variants of *Fun* have similar performance. They also prove that *Fun* is much faster than the Rosetta toolkit's algorithms computing all reducts and faster than *TANE*, which is one of the most efficient algorithms computing all minimal functional dependencies.

1 Introduction

The determination of minimal functional dependencies is a standard task in the area of relational databases. TANE [5] or Dep-Miner [11] are example efficient algorithms for discovering minimal functional dependencies from relational databases. A variant of the task, which consists in discovering minimal sets of conditional attributes that functionally or approximately determine a given decision attribute, is one of the topics of Artificial Intelligence and Data Mining. Such sets of conditional attributes can be used, for instance, for building classifiers. In the terms of Rough Sets, such minimal conditional attributes are called reducts [13]. One can distinguish a number of types of reducts. Generalized decision reducts (or equivalently, possible/approximate reducts [7]), membership distribution reducts (or equivalently, membership reducts [7]), and certain decision reducts belong to most popular Rough Sets reducts. In general, these types of reducts do not determine the decision attribute functionally. However, it was shown in [8] that these types of reducts are minimal sets of conditional attributes functionally determining appropriate modifications of the decision attribute. Thus, the task of searching such reducts is equivalent to looking for minimal sets of attributes functionally determining a given attribute. In this

* Research has been supported by grant No 3 T11C 002 29 received from Polish Ministry of Education and Science.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 320–331, 2007.

paper, we focus on finding all such minimal sets of attributes. To this end, one might consider applying either methods for discovering Rough Sets reducts, or discovering all minimal functional dependencies and then selecting such that determine a requested attribute.

A number of methods for discovering reducts have already been proposed in the literature. e.g. [3-4],[6],[9-10],[12-20]. The most popular methods are based on discernibility matrices [15]. Unfortunately, the existing methods for discovering all reducts are not scalable. The recently offered algorithms for finding all minimal functional dependencies are definitely faster. In this paper, we focus on direct discovery of all minimal functional dependencies with a given dependent attribute, and expect this process to be faster than the discovery of all minimal functional dependencies. Here, we offer an efficient *Fun* algorithm for discovering minimal functional dependencies with a given dependent attribute, and, in particular, for discovering three above mentioned types of reducts. *Fun* can operate either on partitions or alternatively on stripped partitions that do not store singleton groups. It is capable of using functional dependencies occurring among conditional attributes, which are found as a sideeffect, for pruning candidate dependencies.

The layout of the paper is as follows: Basic notions of information systems, functional dependencies, decision tables and reducts are recalled in Section 2. In Section 3, we offer the *Fun* algorithm. The experimental results are reported in Section 4. We conclude our results in Section 5.

2 Basic Notions

2.1 Information Systems

An *information system* is a pair $S = (O, AT)$, where O is a non-empty finite set of *objects* and AT is a non-empty finite set of *attributes* of these objects. In the sequel, $a(x)$, $a \in AT$ and $x \in O$, denotes the value of attribute a for object x, and V_a denotes the *domain* of a. Each subset of attributes $A \subseteq AT$ determines a binary A-*indiscernibility* relation $IND(A)$ consisting of pairs of objects indiscernible wrt. attributes A; that is, $IND(A) = \{(x, y) \in O \times O | \forall_{a \in A}\, a(x) = a(y)\}$. $IND(A)$ is an equivalence relation and determines a partition of O, which is denoted by π_A. The set of objects indiscernible with an object x with respect to A in S is denoted by $I_A(x)$ and is called A-*indiscernibility class*; that is, $I_A(x) = \{y \in O | (x, y) \in IND(A)\}$. Clearly, $\pi_A = \{I_A(x) | x \in O\}$.

2.2 Functional Dependencies

Functional dependencies are of high importance in designing relational databases. We recall this notion after [2]. Let $S = (O, AT)$ and $A, B \subseteq AT$. $A \to B$ is defined a *functional dependency* (or A is defined to *determine* B *functionally*), if $\forall_{x \in O}\, I_A(x) \subseteq I_B(x)$. A functional dependency $A \to B$ is called *minimal*, if $\forall_{C \in A}\, C \to B$ is not functional.

Table 1. Sample DT extended with d^{N}_{AT}, ∂_{AT}, μ_d^{AT}

oid	a	b	c	e	f	d	d^{N}_{AT}	∂_{AT}	$\mu_d^{AT} :< \mu_1^{AT}, \mu_2^{AT}, \mu_3^{AT} >$
1	1	0	0	1	1	1	1	$\{1\}$	$< 1, 0, 0 >$
2	1	1	1	1	2	1	1	$\{1\}$	$< 1, 0, 0 >$
3	0	1	1	0	3	1	N	$\{1, 2\}$	$< 1/2, 1/2, 0 >$
4	0	1	1	0	3	2	N	$\{1, 2\}$	$< 1/2, 1/2, 0 >$
5	0	1	1	2	2	2	2	$\{2\}$	$< 0, 1, 0 >$
6	1	1	0	2	2	2	N	$\{2, 3\}$	$< 0, 1/3, 2/3 >$
7	1	1	0	2	2	3	N	$\{2, 3\}$	$< 0, 1/3, 2/3 >$
8	1	1	0	2	2	3	N	$\{2, 3\}$	$< 0, 1/3, 2/3 >$
9	1	1	0	3	2	3	3	$\{3\}$	$< 0, 0, 1 >$
10	1	0	0	3	2	3	3	$\{3\}$	$< 0, 0, 1 >$

Example 2.2.1. Let us consider the information system in Table 1. $\{ce\} \rightarrow \{a\}$ is a functional dependency, nevertheless, $\{c\} \rightarrow \{a\}$, $\{e\} \rightarrow \{a\}$, and $\emptyset \rightarrow \{a\}$ are not. Hence, $\{ce\} \rightarrow \{a\}$ is a minimal functional dependency. □

Property 2.2.1. Let $A, B, C \subseteq AT$.

a) If $A \rightarrow B$ is a functional dependency, then $\forall_{C \supset A}\ C \rightarrow B$ is functional.
b) If $A \rightarrow B$ is not a functional dependency, then $\forall_{C \subset A}\ C \rightarrow B$ is not a functional dependency.
c) If $A \rightarrow B$ is a functional dependency, then $\forall_{C \supset A}\ C \rightarrow B$ is not a minimal functional dependency.
d) If $A \rightarrow B$ and $B \rightarrow C$ are functional dependencies, then $A \rightarrow C$ is functional.
e) If $A \subset B$, $A \rightarrow B$ is a functional dependency, and $B \cap C = \emptyset$, then $B \rightarrow C$ is not a minimal functional dependency.

Functional dependencies can be calculated by means of partitions [5] as follows:

Property 2.2.2. Let $A, B \subseteq AT$. $A \rightarrow B$ is a functional dependency iff $\pi_A = \pi_{AB}$ iff $|\pi_A| = |\pi_{AB}|$.

Example 2.2.2. Let us consider the information system in Table 1. We observe that $\pi_{\{ce\}} = \pi_{\{cea\}} = \{\{1\}, \{2\}, \{3, 4\}, \{5\}, \{6, 7, 8\}, \{9, 10\}\}$. The equality of $\pi_{\{ce\}}$ and $\pi_{\{cea\}}$ (or their cardinalities) is sufficient to conclude that $\{ce\} \rightarrow \{a\}$ is a functional dependency. □

The next property recalls a method of calculating a partition with respect to an attribute set C by intersecting partitions with respect to subsets of C. Let $A, B \subseteq AT$. The *product of partitions* π_A and π_B, denoted by $\pi_A \cap \pi_B$, is defined as $\pi_A \cap \pi_B = \{Y \cap Z | Y \in \pi_A \text{ and } Z \in \pi_B\}$.

Property 2.2.3. Let $A, B, C \subseteq AT$ and $C = A \cup B$. Then, $\pi_C = \pi_A \cap \pi_B$.

2.3 Decision Tables, Reducts and Functional Dependencies

A *decision table* is an information system $DT = (O, AT \cup \{d\})$, where $d \notin AT$ is a distinguished attribute called the *decision*, and the elements of AT are called *conditions*. A *decision class* is defined as the set of all objects with the same decision value. By X_{d_i} we will denote the decision class consisting of objects the decision value of which equals d_i, where $d_i \in V_d$. Clearly, for any object x in O,

$I_d(x)$ is a decision class. It is often of interest to find minimal subsets of AT (or *strict reducts*) that functionally determine d. It may happen, nevertheless, that such minimal sets of conditional attributes do not exist.

Example 2.3.1. Table 1 describes a sample decision table $DT = (O, AT \cup \{d\})$, where $AT = \{a, b, c, e, f\}$. Partition $\pi_{AT} = \{\{1\}, \{2\}, \{3, 4\}, \{5\}, \{6, 7, 8\}, \{9\}, \{10\}\}$ contains all AT-indiscernibility classes, whereas $\pi_{\{d\}} = \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9, 10\}\}$ contains all decision classes. There is no functional dependency between AT and d, since there is no decision class in $\pi_{\{d\}}$ containing AT-indiscernibility class $\{3, 4\}$ (or $\{6, 7, 8\}$). As $AT \rightarrow d$ is not functional, then $C \rightarrow d$, where $C \subseteq AT$, is not functional either. □

Rough Sets theory deals with the problem of non-existence of strict reducts by means of other types of reducts, which always exist, irrespectively if $AT \rightarrow d$ is a functional dependency, or not. We will now recall such three types of reducts, namely certain decision reducts, generalized decision reducts, and membership distribution reducts.

Certain decision reducts. Certain decision reducts are defined based on the notion of a *positive region* of DT, thus we start with introducing this notion. A *positive region* of DT, denoted as POS, is the set-theoretical union of all AT-indiscernibility classes, each of which is contained in a decision class of DT; that is, $POS = \bigcup\{X \in \pi_{AT} | X \subseteq Y, Y \in \pi_d\} = \{x \in O | I_{AT}(x) \subseteq I_d(x)\}$. A set of attributes $A \subseteq AT$ is called a *certain decision reduct* of DT, if A is a minimal set, such that $\forall_{x \in POS}\ I_A(x) \subseteq I_d(x)$ [13]. Now, we will introduce a *derivable decision attribute* for an object $x \in O$ as a modification of the decision attribute d, which we will denote by $d^{\mathrm{N}}_{AT}(x)$ and define as follows: $d^{\mathrm{N}}_{AT}(x) = d(x)$ if $x \in POS$, and $d^{\mathrm{N}}_{AT}(x) = \mathrm{N}$, otherwise (see Table 1 for illustration). Clearly, all objects with values of d^{N}_{AT} that are different from N belong to POS.

Property 2.3.1 [8]. Let $A \subseteq AT$. A is a certain decision reduct iff $A \rightarrow \{d^{\mathrm{N}}_{AT}\}$ is a minimal functional dependency.

Generalized decision reducts. Generalized decision reducts are defined based on a *generalized decision.* Let us thus start with introducing this notion. An A-*generalized decision* for object x in DT (denoted by $\partial_A(x)$), $A \subseteq AT$, is defined as the set of all decision values of all objects indiscernible with x wrt. A; i.e., $\partial_A(x) = \{d(y) | y \in I_A(x)\}$ [15]. For $A = AT$, an A-*generalized decision* is also called a *generalized decision* (see Table 1 for illustration). $A \subseteq AT$ is defined a *generalized decision reduct* of DT, if A is a minimal set such that $\forall_{x \in O}$ $\partial_A(x) = \partial_{AT}(x)$.

Property 2.3.2 [8]. Let $A \subseteq AT$. Attribute set A is a generalized decision reduct iff $A \rightarrow \{\partial_{AT}\}$ is a minimal functional dependency.

μ-Decision Reducts. The generalized decision informs on decision classes to which an object may belong, but does not inform on the degree of the membership to these classes, which could be also of interest. A *membership distribution function)* $\mu^A_d : O \rightarrow [0, 1]^n, A \subseteq AT, n = |V_d|$, is defined as follows [7],[16-17]:

$$\mu^A_d(x) = (\mu^A_{d_1}(x), \ldots, \mu^A_{d_n}(x)), \text{ where}$$

$$\{d_1, \ldots, d_n\} = V_d \text{ and } \mu^A_{d_i}(x) = \frac{|I_A(x) \cap X_{d_i}|}{|I_A(x)|}.$$

Please, see Table 1 for illustration of μ^{AT}_d. $A \subseteq AT$ is a called a *μ-decision reduct* (or *membership distribution reduct*) of DT, if A is a minimal set such that $\forall_{x \in O}\ \mu^A_d(x) = \mu^{AT}_d(x)$.

Property 2.3.3 [8]. Let $A \subseteq AT$. A is a μ-decision reduct iff $A \to \{\mu^{AT}_d\}$ is a minimal functional dependency.

3 Computing Minimal Sets of Attributes Functionally Determining Given Dependent Attribute with Fun

In this section, we offer the *Fun* algorithm for computing all minimal subsets of conditional attributes AT that functionally determine a given dependent attribute ∂. Clearly, *Fun* shall return certain decision reducts for $\partial = \partial_{AT}$, generalized decision for $\partial = d^{\mathrm{N}}_{AT}$, and μ-decision reducts for $\partial = \mu^{AT}_d$. For brevity, a minimal subset of AT that functionally determines a given dependent attribute ∂ will be called a *∂-reduct.*

3.1 Main Algorithm

The *Fun* algorithm takes two arguments: a set of conditional attributes AT and a functionally dependent attribute ∂. As a result, it returns all ∂-reducts. *Fun* starts with creating singleton candidates $\mathcal{C}_1$ for ∂-reducts from each attribute in AT. Then, the partitions (π) and their cardinalities (*groupNo*) wrt. ∂ and all attributes in $\mathcal{C}_1$ are determined.

Notation for *Fun*	
• $\mathcal{C}_k$	candidate k attribute sets (potential ∂-reducts);
• $\mathcal{R}_k$	k attribute ∂-reducts;
• $C.\pi$	the representation of the partition π_C of the candidate attribute set C; it is stored as the list of groups of objects identifiers (*oids*);
• $C.groupNo$	the number of groups in the partion of the candidate attribute set C; that is, $\lvert\pi_C\rvert$;
• $\partial.T$	an array representation of π_∂;

```
Algorithm $Fun$(attribute set $AT$, dependent attribute $\partial$);
$\mathcal{C}_1 = \{\{a\} | a \in AT\}$;        // create singleton candidates from conditional attributes in $AT$
forall $C$ in $\mathcal{C}_1 \cup \{\partial\}$ do begin
  $C.\pi = \pi_C$;
  $C.groupNo = |\pi_C|$
endfor;
/* calculate an array representation of $\pi_\partial$ for later multiple use in the $Holds$ function */
$\partial.T = PartitionArrayRepresentation(\partial)$;
for ($k = 1$; $\mathcal{C}_k \neq \emptyset$; $k++$) do begin                                  // Main loop
  $\mathcal{R}_k = \{\}$;
  forall candidates $C \in \mathcal{C}_k$ do begin
    if $Holds(C \to \{\partial\})$ then              // Is $C \to \{\partial\}$ a functional dependency?
      remove $C$ from $\mathcal{C}_k$ to $\mathcal{R}_k$;                // store $C$ as a $k$ attribute $\partial$-reduct
    endif
  endfor;
  /* create $(k+1)$ attribute candidates for $\partial$-reducts from $k$ attribute non-$\partial$-reducts */
  $\mathcal{C}_{k+1} = FunGen(\mathcal{C}_k)$;
endfor;
return $\bigcup_k \mathcal{R}_k$;
```

Next, the *PartitionArrayRepresentation* function (see Section 3.3) is called to create an array representation of π_∂. This representation shall be used multiple times in the *Holds* function, called later in the algorithm, for efficient checking whether candidate attribute sets determine ∂ functionally. Now, the main loop starts. In each k-th iteration, the following is performed:

- The *Holds* function (see Section 3.3) is called to check if k attribute candidates $\mathcal{C}_k$ determine ∂ functionally. The candidates that do are removed from the set of k attribute candidates to the set of ∂-reducts R_k.
- The *FunGen* function (see Section 3.2) is called to create $(k + 1)$ attribute candidates $\mathcal{C}_{k+1}$ from the k attribute candidates that remained in $\mathcal{C}_k$.

The algorithm stops when the set of candidates becomes empty.

3.2 Generating Candidates for ∂-Reducts

The *FunGen* function creates $(k + 1)$ attribute candidates $\mathcal{C}_{k+1}$ by merging k attribute candidates $\mathcal{C}_k$, which are not ∂-reducts. The algorithm adopts the manner of creating and pruning of candidates introduced in [1] (here: candidate sets of attributes instead of candidates for frequent itemsets). There are merged only those pairs of k attribute candidates $\mathcal{C}_k$ that differ merely on their last attributes (see [1] for justification that this method is lossless and non-redundant). For each new candidate C, π_C is calculated as the product of the partitions wrt. the merged k attribute sets (see Section 3.3 for the *Product* function). The cardinality (*groupNo*) of π_C is also calculated. Now, it is checked for each new $(k+1)$ attribute candidate C, if there is its k attribute subset A not present in $\mathcal{C}_k$. If

```
function $FunGen(\mathcal{C}_k)$;
/* Merging */
forall $A, B \in \mathcal{C}_k$ do
  if $A[1] = B[1] \wedge \ldots \wedge A[k-1] = A[k-1] \wedge A[k] < B[k]$ then begin
    $C = A[1] \cdot A[2] \cdot \ldots \cdot A[k] \cdot B[k]$;
    /* compute partition $C.\pi$ as a product of $A.\pi$ and $B.\pi$, and the number of groups in $C.\pi$ */
    $C.groupNo = Product(A.\pi, B.\pi, C.\pi)$;
    add $C$ to $\mathcal{C}_{k+1}$
  endif;
endfor;
/* Pruning */
forall $C \in \mathcal{C}_{k+1}$ do
  forall $k$ attribute set $A$, such that $A \subset C$ do
    if $A \notin \mathcal{C}_k$ then
      /* $A \subset C$ and $\exists B \subseteq A$ such that $B \rightarrow \{\partial\}$ holds, so $C \rightarrow \partial$ holds, but is not minimal */
      begin delete $C$ from $\mathcal{C}_{k+1}$; break
      end
    elseif $A.groupNo = C.groupNo$ then                    // optional pruning step
      /* $A \rightarrow C$ holds, so $C \rightarrow \{\partial\}$ is not a minimal functional dependency */
      begin delete $C$ from $\mathcal{C}_{k+1}$; break
      end
    endif
  endfor
endfor;
return $\mathcal{C}_{k+1}$;
```

∂-reduct, and hence C is deleted from the set $\mathcal{C}_{k+1}$. Optionally, for each tested k attribute subset A that is present in $\mathcal{C}_k$, it is checked, if $|\pi_A|$ equals $|\pi_C|$. If so, then $A \rightarrow C$ holds (by Property 2.2.2). Hence, $A \rightarrow \{\partial\}$ is not a minimal functional dependency (by Property 2.2.1e), and thus C is deleted from $\mathcal{C}_{k+1}$.

3.3 Using Partitions in Fun

Computing Array Representation of Partition. The *PartitionArrayRepresentation* function returns an array T of the length equal to the number of objects O in DT. For a given attribute C, each element of T is assigned the index of the group in $C.\pi$ to which the index of the element belongs. As a result, j-th element of T informs to which group in $C.\pi$ j-th object in DT belongs, $j = 1..|O|$.

```
function PartitionArrayRepresentation(attribute set $C$);
/* assert: $T$ is an array[1 ... |O|] */
$i = 1$;
for $i$-th group $G$ in partition $C.\pi$ do begin
   for each $oid$ $G$ do $T[oid] = i$ endfor;
   $i = i + 1$
endfor
return $T$;
```

Verifying Candidate Dependency. The *Holds* function checks, if there is a functional dependency between the set of attributes C and an attribute ∂. It is checked for successive groups G in $C.\pi$, if there is an *oid* in G that belongs to a group in $\partial.\pi$ different from the group in $\partial.\pi$ to which the first *oid* in G belongs (for the purpose of efficiency, the pre-calculated $\partial.T$ representation of the partition for ∂ is applied instead of $\partial.\pi$). If so, this means that G is not contained in one group of $\partial.\pi$ and thus $C \rightarrow \{\partial\}$ is not a functional dependency. In such a case, the function stops returning **false** as a result. Otherwise, if no such group G is found, the function returns **true**, which means that $C \rightarrow \{\partial\}$ is a functional dependency.

```
function Holds($C \rightarrow \{\partial\}$);
/* assert: $\partial.T$ is an array representation of $\partial.\pi$ */
for each group $G$ in partition $C.\pi$ do begin
   $oid$ = first element in group $G$;
   $\partial$-firstGroup = $\partial.T[oid]$;          // the identifier of the group in $\partial.\pi$ to which $oid$ belongs
   for each next element $oid \in G$ do begin
      $\partial$-nextGroup = $\partial.T[oid]$;
      if $\partial$-firstGroup $\neq$ $\partial$-nextGroup then
         /* there are oids in $G$ that identify objects indiscernible wrt. $C$, but discernible wrt. $\partial$ */
         return false                                  // hence, $C \rightarrow \{\partial\}$ does not hold
      endif
   endfor;
endfor;
return true;                                                         // $C \rightarrow \{\partial\}$ holds
```

Computing Product of Partitions. The *Product* function computes the partition wrt. the attribute set C and its cardinality from the partitions wrt. the attribute sets A and B. The function examines successive groups wrt. the partition for B. The objects in a given group G in $B.\pi$ are split into maximal subgroups in such a way that the objects in each resultant subgroup are contained in a same group in $A.\pi$. The obtained set of subgroups equals $\{G \cap Y | Y \in A.\pi\}$. Product $C.\pi$ is calculated as the set of all subgroups obtained from all groups in $B.\pi$; i.e., $C.\pi = \bigcup_{G \in B.\pi} \{G \cap Y | Y \in A.\pi\} = \{G \cap Y | Y \in A.\pi \text{ and } G \in B.\pi\} = B.\pi \cap A.\pi$.

In order to calculate the product of the partitions efficiently (with time complexity linear wrt. the number of objects in DT), we follow the idea presented in [5], and use two static arrays T and S: T is used to store an array representation of the partition wrt. A; S is used to store subgroups obtained from a given group G in $B.\pi$.

```
function $Product(A.\pi, B.\pi;$ var $C.\pi)$;
/* assert: $T[1..|O|]$ is a static array */
/* assert: $S[1..|O|]$ is a static array with all elements initially equal to $\emptyset$ */
$C.\pi = \{\}$; $groupNo = 0$;
/* calculate an array representation of $A.\pi$ for later multiple use in the $Product$ function */
$T = PartitionArrayRepresentation(A)$; $i = 1$;
for $i$-th group $G$ in partition $B.\pi$ do begin
   $A\text{-}GroupIds = \emptyset$;
   for each element $oid \in G$ do begin
      $j = T[oid]$;                    // the identifier of the group in $A.\pi$ to which $oid$ belongs
      insert $oid$ into $S[j]$; insert $j$ into $A\text{-}GroupIds$
   endfor;
   for each $j \in A\text{-}GroupIds$ do begin
      insert $S[j]$ into $C.\pi$;
      $groupNo = groupNo + 1$; $S[j] = \emptyset$
   endfor;
   $i = i + 1$
endfor;
return $groupNo$;
```

3.4 Using Stripped Partitions in Fun

The representation of partitions that requires storing objects identifiers (*oids*) of all objects in DT may be too memory consuming. In order to alleviate this problem, it was proposed in [5] to store *oids* only for objects belonging to non-singleton groups in a partition representation. Such a representation of a partition is called a *stripped* one. Clearly, the stripped representation is lossless.

```
function $StrippedHolds(C \rightarrow \{\partial\})$;
$i = 1$;
for $i$-th group $G$ in partition $C.\pi$ do begin
   $oid$ = first element in group $G$;
   $\partial\text{-}firstGroup = \partial.T[oid]$;          // the identifier of the group in $\partial.\pi$ to which $oid$ belongs
   if $\partial\text{-}firstGroup$ = null then return false endif;
   /* $\partial.T[oid]$ = null indicates that $oid$ constitutes a singleton group in the partition for $\partial$. */
   /* Hence, no next object in $G$ belongs to this group in $\partial.\pi$ , so $C \rightarrow \{\partial\}$ does not hold.   */
   for each next element $oid \in G$ do begin
      $\partial\text{-}nextGroup = \partial.T[oid]$;
      if $\partial\text{-}firstGroup \neq \partial\text{-}nextGroup$ then
         /* there are $oids$ in $G$ that identify objects indiscernible wrt. $C$, but discernible wrt. $\partial$ */
         return false                                  // hence, $C \rightarrow \{\partial\}$ does not hold
      endif
   endfor;
   $i = i + 1$
endfor;
return true;                                               // $C \rightarrow \{\partial\}$ holds
```

Example 3.4.1. In Table 1, the partition wrt. $\{ce\}$ equals $\{\{1\}, \{2\}, \{3, 4\}, \{5\}, \{6, 7, 8\}, \{9, 10\}\}$, whereas the stripped partition wrt. $\{ce\}$ equals $\{\{3, 4\}, \{6, 7, 8\}, \{9, 10\}\}$. □

When applying stripped partitions in our *Fun* algorithm instead of usual partitions, one should call the *StrippedHolds* function instead of *Holds*, and the

StrippedProduct function instead of *Product*. The modified parts of the functions have been shadowed in the code below. We note, however, that the *groupNo* field still stores the number of groups in an unstripped partition (singleton groups are not stored, but are counted!).

```
function StrippedProduct($A.\pi$, $B.\pi$; var $C.\pi$);
$C.\pi = \{\}$; $groupNo = B.groupNo$;
$T = PartitionArrayRepresentation(A)$; $i = 1$;
for $i$-th group $G$ in partition $B.\pi$ do begin
  $A - GroupIds = \emptyset$;
  for each element $oid \in G$ do begin
    $j = T[oid]$;          // the identifier of the group in $A.\pi$ to which $oid$ belongs
    if $j$ = null then $groupNo = groupNo + 1$;          // respect singleton subgroups
    else begin insert $oid$ into $S[j]$; insert $j$ into $A$-$GroupIds$ endif
  endfor;
  for each $j \in A - GroupIds$ do begin
    if $|S[j]| > 1$ then
      insert $S[j]$ into $C.\pi$          // store only non-singleton groups
    endif ;
    $groupNo = groupNo + 1$; $S[j] = \emptyset$          // but count all groups, including singleton ones
  endfor;
  $groupNo = groupNo - 1$;
  $i = i + 1$
endfor;
/* Clearing of array $T$ for later use */
for $i$-th group $G$ in partition $A.\pi$ do
  for each element $oid \in G$ do $T[oid]$ = null endfor
endfor;
return $groupNo$;
```

4 Experimental Results

We have performed a number of experiments on a few data sets available in UCI Repository datasets (http://www.ics.uci.edu/˜mlearn/MLRepository.html) and other used by the Rough Sets community. We have reported the times of discovering reducts by four variants of *Fun*, as well as, the *TANE*, *SAVGeneticReducer* and *RSESExhaustiveReducer* algorithms. We used the implementation of *TANE* provided by its authors. *SAVGeneticReducer* and *RSESExhaustiveReducer*, used for experiments, come from the Rosetta toolkit. Because of Rosetta limitations, we did not perform experiments with *RSESExhaustiveReducer* on datasets larger than 500 records.

As follows from Table 2, *Fun* is faster than *TANE* and much faster than the both algorithms from Rosetta. The performance of the four variants of Fun is similar. In Figure 1, we plotted times of the performance of *Fun*, *TANE* and *SAVGeneticReducer* for the nursery dataset in a logarithmic scale. The time performance of *Fun* and *TANE* is linear wrt. the number of objects in the dataset. The time performance of *SAVGeneticReducer* is 2 to 3 orders of magnitude greater and is non-linear wrt. the number of objects. In Figures 2-4, we presented the time performance of *Fun* and *TANE* in a linear scale for the nursery,

Table 2. Comparison of *Fun*, *TANE*, *SAVGeneticReducer* and *RSESExhaustive-Reducer*. Time is given in milliseconds ($^+$ - originally, time measured in seconds); * - a data set does not contain an object id; F - # of min. functional dependencies; P - applied optional pruning step in *FunGen*; S - applied stripped partitions

Data set $DT = (O, AT \cup \{d\})$			Fun	Fun	Fun	Fun	TANE	SAV Genetic	RSES Exhaustive	
Name	$\lvert O \rvert$	$\lvert AT \rvert$	-	P	S	PS	S	Reducer	Reducer	F
diabetic.33	33	12	10	10	10	10	30	<500 (or 0 sec)$^+$	<500 (or 0 sec)$^+$	2
diabetic.33*	33	11	20	10	10	10	20	<500 (or 0 sec)$^+$	<500 (or 0 sec)$^+$	10
diabetic	107	12	50	30	30	30	40	<500 (or 0 sec)$^+$	<500 (or 0 sec)$^+$	9
diabetic*	107	11	40	40	20	20	30	<500 (or 0 sec)$^+$	<500 (or 0 sec)$^+$	14
nursery.500	500	9	20	10	10	10	10	<500 (or 0 sec)$^+$	18000 (or 18 sec)$^+$	8
nursery.500*	500	8	20	10	10	10	10	<500 (or 0 sec)$^+$	17000 (or 17 sec)$^+$	2
nursery	12960	9	451	539	471	481	681	274000 (or 274 sec)$^+$	not available$^+$	1
nursery*	12960	8	441	450	451	481	701	247000 (or 247 sec)$^+$	not available$^+$	2
krkopt	8056	6	250	251	260	250	420	1296000(or 1296 sec)$^+$	not available$^+$	1

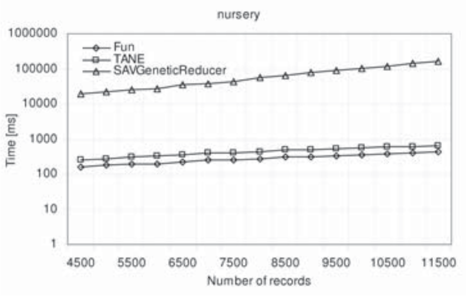


Fig. 1. nursery - logarithmic scale

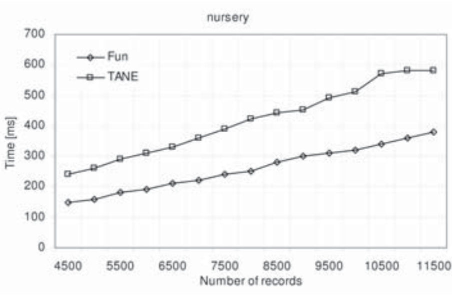


Fig. 2. nursery - linear scale

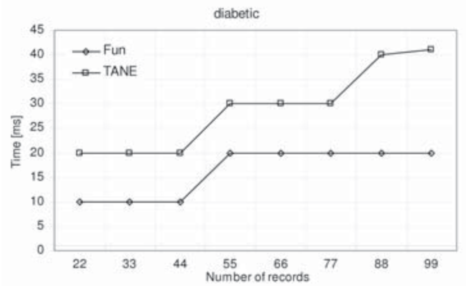


Fig. 3. diabetic - linear scale

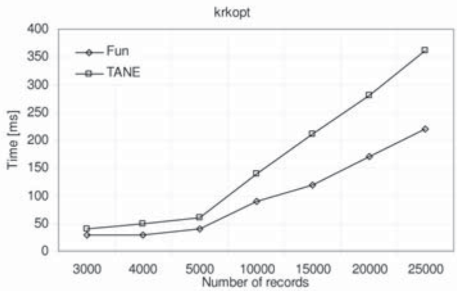


Fig. 4. krkopt - linear scale

diabetic and krkopt datasets, respectively. On average, *TANE* is approximately by 60%, 80%, and 60% slower than *Fun* for the respective datasets.

5 Conclusions and Future Work

We have proposed the *Fun* algorithm for discovering minimal sets of conditional attributes functionally determining a decision attribute, and in particular for computing certain, generalized decision, and μ-distribution reducts. *Fun* is consistently faster than *TANE*, which computes all minimal functional dependencies, and is orders of magnitude faster than *SAVGeneticReducer* and *RSES-ExhaustiveReducer* from Rosetta. The four variants of *Fun*, we have implemented and tested, show similar performance. We are going to continue testing their performance on a diverse large datasets. We intend to specify categories of datasets and appropriate (fastest) variants of *Fun* for them.

References

[1] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast Discovery of Association Rules. In: Advances in KDD. AAAI, Menlo Park, California, pp. 307–328 (1996)

[2] Armstrong, W.W.: Dependency Structures of Data Based Relationships. In: Proc. of IFIP Congress. Geneva, Switzerland, pp. 580–583 (1974)

[3] Bazan, J., Skowron, A., Synak, P.: Dynamic Reducts as a Tool for Extracting Laws from Decision Tables. In: Raś, Z.W., Zemankova, M. (eds.) ISMIS 1994. LNCS, vol. 869, pp. 346–355. Springer, Heidelberg (1994)

[4] Bazan, J., Nguyen, H.S., Nguyen, S.H., Synak, P., Wroblewski, J.: Rough Set Algorithms in Classification Problem. In: Rough Set Methods and Applications, pp. 49–88. Physica- Verlag, Heidelberg (2000)

[5] Huhtala, Y., Karkkainen, J., Porkka, P., Toivonen, H.: TANE: An Efficient Algorithm for Discovering Functional and Approximate Dependencies. The. Computer Journal 42(2), 100–111 (1999)

[6] Jelonek, J., Krawiec, K., Stefanowski, J.: Comparative study of feature subset selection techniques for machine learning tasks. In: Proc. of IIS, Malbork, Poland, pp. 68–77 (1998)

[7] Kryszkiewicz, M.: Comparative Study of Alternative Types of Knowledge Reduction in Inconsistent Systems. Intl. Journal of Intelligent Systems 16(1), 105–120 (2001)

[8] Kryszkiewicz, M.: Certain, Generalized Decision, and Membership Distribution Reducts versus Functional Dependencies in Incomplete Systems. RSEISP, LNAI (2007)

[9] Kryszkiewicz, M., Cichon, K.: Towards Scalable Algorithms for Discovering Rough Set Reducts. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.., Świniarski, R.W., Szczuka, M. (eds.) Transactions on Rough Sets I. Journal Subline, LNCS, vol. 3100, pp. 120–143. Springer, Heidelberg (2004)

[10] Lin, T.Y.: Rough Set Theory in Very Large Databases. In: Proc. of CESA IMACS, Lille, France, vol. 2, pp. 936–941 (1996)

[11] Lopes, S., Petit, J.-M., Lakhal, L.: Efficient Discovery of Functional Dependencies and Armstrong Relations. In: Proc. of EDBT, pp. 350–364 (2000)

[12] Nguyen, S.H., Skowron, A., Synak, P., Wroblewski, J.: Knowledge Discovery in Databases: Rough Set Approach. In: Proc. of IFSA, vol. 2, pp. 204–209, Prague, (1997)

[13] Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data, vol. 9. Kluwer Academic Publishers, Boston (1991)

[14] Shan, N., Ziarko, W., Hamilton, H.J., Cercone, N.: Discovering Classification Knowledge in Databases Using Rough Sets. In: Proc. of: KDD, pp. 271–274 (1996)

[15] Skowron, A.: Boolean Reasoning for Decision Rules Generation. ISMIS, 295–305 (1993)

[16] Slezak, D.: Approximate Reducts in Decision Tables. In: Proc. of IPMU, Granada, Spain, vol. 3, pp. 1159–1164 (1996)

[17] Slezak, D.: Searching for Frequential Reducts in Decision Tables with Uncertain Objects. In: Proc. of RSCTC, Warsaw, 1998, pp. 52–59. Springer, Heidelberg (1998)

[18] Slowinski, R. (ed.): Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory, vol. 11. Kluwer Academic Publishers, Boston (1992)

[19] Stepaniuk, J.: Approximation Spaces, Reducts and Representatives. In: Rough Sets in Data Mining and Knowledge Discovery, Springer, Berlin (1998)

[20] Wroblewski, J.: Finding Minimal Reducts Using Genetic Algorithms. In: Proc. of JCIS, September/October 1995, Wrightsville Beach, NC, pp. 186–189 (1995)

A Simple Reduction Analysis and Algorithm Using Rough Sets*

Ning Xu[1], Yun Zhang[2], and Yongquan Yu[2]

[1] Dept. of Computer Science and Information Engineering,
Shanghai Institute of Technology,
[2] Automation Faculty, Guangdong University of Technology
xun@sit.edu.cn,yz@gdut.edu.cn,yyq@gdut.edu.cn

Abstract. Indiscernibility relation and attribute reduction criteria are the important concepts in rough sets, also are the important base for further researching on attribute reduction. By analyzing the set theory background of indiscernibility and the reduction theory, it can be seen that an information system has the similar characteristics relative to a relation database table and can be analyzed using its data table structure. Combining the structure information of the system with the rough sets reduction theory, a simple reduction analysis can be completed and get useful reduction information. Such as whether a system has redundant attributes or not and how many attributes are need by the system to maintaining its classes, etc. The analysis is realized by a simple algorithm: PARA, the algorithm together with an effective heuristics algorithm can decide an area of the minimum reduct. It greatly reduces the searching area of finding minimum reduct and can play some role in high-dimensionality reduction. A given example shows the algorithm.

Keywords: rough sets, attribute reduction, set theory, data table structure, algorithm.

1 Introduction

Attribute reduction, also called dimensionality reduction or feature selection, has important meaning in data mining, pattern recognition, machine learning, artificial intelligence, and so on. It is one of key techniques in data pre-processing and data compression. Its research has going on for thirty years and obtained plentiful results. Rough sets, proposed by Poland mathematic professor Zdzislaw Pawlak[1,2] in 1982, is one of the most important results.

Rough sets has been very famous in dealing with imprecise or fuzzy data, and also famous in attribute reduction as a new mathematic tool. It sets up reduction theory and reduction criteria which can be applied to general structured data.

* The project is supported by Guangdong Natural Science Foundation (China) (No.04009480, No.06301299) and Appropriative Researching Fund for Professors and Doctors, Guangdong Institute of Education ARF(GDEI).

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 332–339, 2007.

The theory changes the attribute reduction situation and is widely used to many practical fields. Today, rough sets is attracting great attention from the all world.

Rough sets puts forward the reduction theory of holding classified characteristics of a datable. Because always there are several subsets meet the criteria, people seek the way of getting the attribute subsets as possible as quickly, correctly and smallest. Among the subsets, the minimum reduct, with least attributes, has the most research value. Although experts had proved that to get the minimum reduct is still a NP-hard problem[3] to high-dimensionality datasets by rough sets (the hugeness of attribute subset amount makes the load to find the optimum subset cannot be accepted.), but by virtue of rough sets, the further research is going on and entering a new phase.

This paper discusses the set theory background of rough sets reduction theory and information system structure features. Comparing with relation database table, a reduction analysis of an information system and the algorithm PARA are gained. The algorithm gives the lower limit to search the minimum reduct, if it's together with an efficient heuristic algorithm, a certain area to find the minimum reduct can be decided, which is greatly reduced relative to search in the all subsets.

The paper is arranged as below: in the second part is rough sets reduction theory, the third part is its set theory meaning on partition; the fourth part is the discussion of relation datable and information system structure characteristics including the application in reduction analysis; the fifth part is the algorithm and examples. The last part gives conclusion.

2 Rough Sets Reduction Theory

A data table is called an information system[1] in rough sets when it is processed discretely and is described as: $S = \{U, A, V, f\}$. U-the universal, $U = \{x_1, x_2, ..., x_n\}$; A-the set of all attributes; V-the set of all values of attributes; f-the map function, $f : U \times A \to V$. Generally, $A = C \cup D$, C is condition attribute set and D is decision attribute set.

The attribute reduction of rough sets is set up on objects' classification.

For any $P \subsetneqq A$, the $\cap P$ gives an equivalence relation, denoted with ind(P), called indiscernibility relation:

$\mathrm{ind}(P) = \{(x, y) \in U^2 | a \in P, a(x) = a(y)\}$

Ind(P) generates a partition on U, or a group of equivalent classes. They are denoted by $U/\mathrm{ind}(P)$. Set $X \subseteq U$and a $R \in A$, rough sets defines lower approximation of X in R is:

$R_{-}(X) = \cup\{Y \in U/R|Y \subseteq X\}$

And upper approximation of X in R is:

$R^{-}(X) = \cup\{Y \in U/R|Y \cap X \neq \emptyset\}$

The lower approximation is also expressed by $\mathrm{pos}_R(U, X)$, $\mathrm{pos}_R(U, X) = R_{-}(X)$, called positive region. If $U/\mathrm{ind}(D) = \{Y_1, Y_2, ..., Y_k\}$ is a equivalent relation given by decision attributes, $P \subseteq C$, then the positive region of P in C with respect to D is:

$$pos_{ind(P)}(D) = \bigcup_{i=1}^{k} pos_{ind(p)}(Y_i) \tag{1}$$

If $c \in C$, and

$$pos_{ind(C-c)}(D) = pos_{ind(C)}(D) \tag{2}$$

then c is dispensable (can be reduced or reducted) with respect to D; else c is necessary. Rough sets reduction is defined as: $P \subsetneqq C$if every c in P is necessary with respect to D, then P is considered independent with respect to D. If P is independent about D, and

$$pos_{ind(P)}(D) = pos_{ind(C)}(D) \tag{3}$$

then call P is a reduct in C with respect to D denoted by $\text{red}_D(C)$. Generally, there are several reducts meet (3) in an information system, the intersection of all these reducts is called core, denoted by $\text{core}_D(C) = \cap\ \text{red}_D(C)$.

Form (3) shows that: some redundant attributes can be reduced from datable, only the information system S maintains the positive region unchanged.

3 Set Theory and Partition

Every indiscernibility relation on U, such as ind(C), ind(P), ind(D) or R, is an equivalence relation, and gives a partition on U. A reduction analysis can be developed on set theory partition.

*Define*3.1 Presume: $\pi_1 = \{A_1, A_2, ..., A_n\}, \pi_2 = \{B_1, B_2, ..., B_m\}$ are two partitions of set U, if every A_i is a subset of some B_j, then π_1 is considered as a refinement of π_2. It is denoted by: $\pi_1 \preccurlyeq \pi_2$. If one of A_i is a proper subset of some B_j at least, then π_1 is considered as a proper refinement of π_2. It is denoted by: $\pi_1 \prec \pi_2$.

*Define*3.2 Presume π_1 and π_2 are two partitions of set U, call $\{A_i \cap B_j | A_i \cap B_j \neq \emptyset, i = 1, 2, ..., n; j = 1, 2, ..., m\}$ as product of partitions of π_1 and π_2, denoted by: $\pi_1 \cdot \pi_2$.

Bellow lemmas can be proved easily.

*Lemma*3.1[5] If π_1 and π_2 are two partitions of set U, R_1 and R_2 are the equivalence relations of π_1 and π_2 correspondingly, then product of partitions: $\pi_1 \cdot \pi_2$ is the corresponding partition of equivalence relation $R_1 \cap R_2$.

*Lemma*3.2 If π_1 and π_2 are two partitions of set U, then the product of partitions $\pi(\pi_1 \cdot \pi_2)$ meets: $\pi \preccurlyeq \pi_1$. $\pi \preccurlyeq \pi_2$. If $\pi_1 \neq \pi_2$, the product of partitions π will be a proper refinement of one of the two partitions of π_1 and π_2 at least.

It can be proved as bellow.

Prove : Suppose π_1 and π_2 are two partitions of set U, and $\pi = \{A_i \cap B_j | A_i \cap B_j \neq \emptyset, i = 1, 2, ..., n; j = 1, 2, ..., m\}$. $\because A_i \cap B_j \subsetneqq A_i, A_i \cap B_j \subsetneqq B_j$ by define 3.1, $\therefore \pi \preccurlyeq \pi_1, \pi \preccurlyeq \pi_2$.

Suppose: $\pi = \{C_1, C_2, ..., C_l\}, \pi_1 \neq \pi_2, \exists i, j$, that: $A_i \neq B_j$ and $A_i \cap B_j \neq \emptyset$.

There are three kinds of situations:

(1) $B_j \subset A_i$, then $A_i \cap B_j = \{C_k\} = B_j, \because C_k \subset A_i, \therefore \pi \prec \pi_1$;
(2) $A_i \subset B_j$, then $A_i \cap B_j = \{C_k\} = A_i, \because C_k \cap B_j, \therefore \pi \prec \pi_2$;
(3) $A_i \not\subseteq B_j, B_j \not\subseteq A_i, A_i \cap B_j \neq \emptyset$, If $A_i \cap B_j = \{C_k\}$ $(C_{k-1} + C_k = A_i$ and $C_k + C_{k+1} = B_j$, $\because C_k \subset A_i$ and $C_k \subset B_j$, by define 3.1, $\therefore \pi \prec \pi_1$ and $\pi \prec \pi_2$.

So, when two partitions are unequal, their product of partitions is a proper refinement of one of them at least. End

Among all the partitions, there are two extraordinary partitions, which are: π_S and π_G.

$\pi_S = \{U\}$ is called minimum partition, the cardinality of its equivalence relation R_S (the number of equivalence class) is 1.

$\pi_G = \{\{x_1\}, \{x_2\}, ..., \{x_n\}\}$ is called maximum partition, the cardinality of its equivalence relation R_G is n.

For any equivalence relation R on U, its partition: π_R=U/R=$\{X_1, X_2, ..., X_R\}$. Its cardinality: card(R) = $|U/R|$ always meets: $1 \leqslant$ card(R) $\leqslant n$.

For two equivalence relations R_1 and R_2, their indiscernibility relation $R = R_1 \cap R_2$, the cardinality of R, by lemma 3.2, always meets: card(R)$\geqslant$card(R_1), card(R)$\geqslant$card(R_2).

So, when increasing equivalence relation to a set of equivalence relations, the cardinality of the new equivalence relation will be increased, or does not decrease, the new partition will be finer.

An information system S, $P \subseteq C$, when new condition attribute is added to P, the indiscernibility relation ind(P) will be more and finer then before, $c \in C - P$, the cardinality of ind(P) meets: card($P \cap c$)$\geqslant$ card(P).

In the S, there are $|C|$ condition attributes, $P \subseteq C$, it will be always true that: card(C)$\geqslant$ card(P).

4 Data Table Structure and Its Characteristics

Information system is a kind of relation database tables. To a relation data table, its number of attributes and the number of every attribute values decide how many different objects can be described by them. The attributes and their values decide the structure of the table.

According to the definition of rough sets, $P \subseteq C$, the indiscernibility relation ind(P):

ind(P) = $\{(x, y) \in U^2 | \forall c \in P, c(x) = c(y)\}$.

If$|U/\text{ind}(C)| \neq n$, it means there are some objects in the same equivalence class of U/ind(C), they can not be identified by ind(C), they are the same objects in it. These objects are considered having the same classical knowledge, they are repeated objects.

The numbers of same objects can support the rules expression in reasoning analysis, but no more meaning in reduction analysis; they can be reduced and does not affect the attribute reduction analysis. After reducing the redundant objects, the partition of ind(C) always arrives maximum partition π_G.

For simplifying the discussion, suppose ind(C) always gives the maximum partition on U(Every data table can realize it). It means that between any two objects at least there is one attribute, in which the two objects have different values:

If $U = \{x_1, x_2, ..., x_n\}$, and $C = \{c_1, c_2, ..., c_m\}, \exists k$ (at least one) makes it true:

$c_k(x_i) \neq c_k(x_j), i \neq j, k \in [1, 2, ..., m]$.

The structure of a database table can decide the number of different objects. Similar to database table, the structure of an information system also can decide the number of different objects. Vice versa, if the structure and the object number have been known, whether the attributes are redundant or not can be understudied.

According to the number of attributes and the number of the attribute's values, how many different objects are described by them can be computed. And if the result is greatly greater than the objects in the system, that says the attributes of the system are enough or redundant to needed to describe the objects differences, so the reduction is possible and efficient. If the result is not greater greatly than the objects in system, the reduction may not be efficient.

The detail computing and analysis can be carried out as following:

An information system S, if there are m condition attributes, and among them, m_1 attributes have t_1 different values, m_2 attributes have t_2 different values, m_3 attributes have t_3 different values,...,and m_r attributes have t_r different values, then the system can describe n_N different objects:

$$n_N = \prod_{i=1}^{r} t_i^{m_i}, 2 \leqslant t_i \leqslant |U|, \sum_{i=1}^{r} m_i = m. \tag{4}$$

$|U| = n$, if $n_N >> n$, the system has enough or redundant attributes.

$\forall c, c \in C, |U/c| = s$, arraying the s from big to small:$s_1 \geqslant s_2 \geqslant s_3 \geqslant ... \geqslant s_m$, multiplying them one by one, then $\exists p_0, p_0 \in I, I = \{1, 2, 3, ..., m\}$, makes the two formulas to be true:

$$\prod_{i=1}^{p_0-1} s_i \leqslant n, \prod_{i=1}^{p_0} s_i \geqslant n \tag{5}$$

The formulas show: at least p_0 attributes are needed to describe the different objects in the system. If less than p_0, the system certainly does not arrive maximum partition π_G.

Based on rough set reduction theory, if the partition is π_G by ind(P), $P \subseteq C$, then to any decision attribute ind(D), always has: $pos_{ind(P)}(D) = \{U\}$, of course has: $pos_{ind(C)}(D) = \{U\}$.

Because a reduct must meet (3), so the attribute number in a reduct generally has: $|red_D(C)| \approx p_0$, or $|red_D(C)| \geqslant p_0$.

The result displays the lowest limit-p_0 to search minimum reducts, it is the least attributes to maintain the system classes.

5 Algorithm and Examples

The n_N of a system can be get by (4), if $n_N \gg n$, it shows the datable has enough or redundant attributes, else the system may not have redundant attribute. The p_0 of the system can be get by (5), $|C| - p_0$ is the maximal attribute number which may be reduced from the system.

An unchanged positive region $pos_{ind(C)}(D) = pos_{ind(P)}(D)$ of a system is a criterion of attribute can be reduced in it. And only when $pos_{ind(P)}(D) = pos_{ind(C)}(D) = \{U\}$, the attribute subset P is a reduct by (3).

Following is the algorithm of reduction analysis PARAPre-Analysis of attribute Reduction Algorithm to an information system S

①. Deciding t_i, m_i of C, computing n_N using formula (4);

②. List s_i of every c_i, compute p_0 using (5);
③. IF $|C| - p_0 = 0$ THEN stop and exit, ELSE;
④. Computing ind(C) of the system;
⑤. Computing pos$_{ind(C)}(D)$ of the system;
⑥. $\forall c \in C$IF pos$_{ind(C-c)}(D) \neq$ pos$_{ind(C)}(D)$ THEN core(C)= core$(C) \cup \{c\}$;
⑦. $C' = C-$core(C)
⑧. Output $n_N, n, p_0, |C| - p_0$, core(C), $p_0 -$ |core$(C)|, C'$.

This is a classical CTR(Car Test Result) table [6], and is classified, as in Table 1, by the reduction analysis PARA, as follows:

①. Deciding $t_1 = 2, m_1 = 7; t_2 = 3, m_2 = 2$; by formula(4):$n_N = 1152$;
②. The s_i list: 3,3,2,2,2,2,2,2,2, use formula (5): $p_0 = 4$;
③. $|C| - p_0 = 5$, turn ④;
④. $U/\text{ind}(C) = \{\{1\}, \{2\}, \{3\}, \{4\}, ..., \{21\}\}$ (maximum partition);
⑤. pos$_{ind(C)}(D) = \{1, 2, 3, 4, ..., 21\} = \{U\}$;
⑥. pos$_{ind(C-d)}(D) \neq \{U\}$ and pos$_{ind(C-i)}(D) \neq \{U\}$, so core$_D(C) = \{d, i\}$;
⑦. $C' = C-$core$_D(C) = \{a, b, c, e, f, g, h\}$;
⑧. Output: $n_N = 1152, n = 21, p_0 = 4, |C| - p_0 = 5$,core$_D(C) = \{d, i\}$,
$p_0 -$ |core$(C)| = 2, |C| -$ |core$_D(C)| = 7$.

Because $n_N >> n(1152 >> 21)$, so the system has enough condition attributesand may half of them are redundant. Because $p_0 = 4$, so four attribute could become a reduct. As the core has two attributes, other two attributes are needed to realize a reduct. Now, the area to find a reduct among all subsets of 7 attributes is reduced to the subsets which only has 2 attributes, the number of attribute subsets is reduced from 511 to $C_7^2 = 21$. A minimum reduct is: $red_D(C) = \{d, i, a, e\}$.

The other example is from reference [7]. Its dataset comes from records of medical treatment. There are 20 inspective attributes, and 568 cases. Five exports divided the cases to 5 classes. By use of the reduction analysis algorithm PARA, the out put is: $n_N \approx 8.0 \times 10^{14}$, that says $n_N >> n$, and $p_0 = 4$, so the system, from its structure, can at most be reduced 16 attributes, and only 4 attributes can describe the difference between every two objectors to the 568

Table 1. Classified CTR Dataset

U	a	b	c	d	e	f	g	h	i	D
1	0	1	1	1	1	1	1	1	0	1
2	0	1	0	1	1	0	1	0	0	1
3	0	1	0	1	1	1	1	0	0	1
4	0	0	1	1	1	1	1	0	1	2
5	0	1	0	1	1	0	0	0	0	1
6	0	1	0	0	1	0	0	1	2	0
7	0	1	0	1	1	0	1	0	2	0
8	1	0	0	0	0	1	2	0	1	2
9	0	0	0	0	0	1	2	0	0	1
10	0	0	0	0	0	1	0	1	0	1
11	1	0	0	1	0	1	2	0	1	2
12	1	0	0	1	1	0	0	0	0	2
13	0	0	0	0	1	0	0	0	0	1
14	1	0	1	1	0	1	1	0	0	2
15	1	0	0	0	0	0	2	0	0	2
16	0	0	1	1	1	0	1	0	0	1
17	0	1	0	1	1	0	1	1	0	1
18	0	0	0	1	1	0	1	1	0	1
19	1	0	0	1	0	1	0	0	0	2
20	0	0	0	1	0	1	0	0	0	2
21	0	0	0	0	0	1	0	0	0	1

cases. There is one attribute in its core; the area of finding minimum reducts can only check attribute subsets, not 20! attribute subsets. The two numbers show that the searching area is reduced greatly.

6 Conclusion

Attribute reduction, especially in high-dimensionality, has many important meanings. This article discusses a reduct analysis algorithm from indiscernibility relation and the set characteristics. After combining with dataset structure, a reduction pre-analysis gives much information and determined the lower limit for searching reducts. The examples show the algorithm is efficient and easy to understand and use. It provides a useful analysis before dimensionality reduction. It could play some role in heuristic reduction and finding minimum reducts. The research tries to give a new and reference way on dimensionality reduction.

References

1. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
2. Pawlak, Z.: Rough Sets and Their Applications. Microcomputer Applications 13(2), 71–75 (1994)

3. Wong, S.K.M., Ziarko, W.: On optimal decision rules in decision tables. Bulletin of Polish Academy of Science 33, 693–696 (1995)
4. Ning, X.: The Theory and Technique Research of Attribute Reduction in Data Mining Based on Rough Sets, PhD dissertation, Guangdong University of Technology (2005)
5. Ziwei, N., Jingqiu, C.: Discrete Mathematics Science Publishes 2002.10.
6. Wenxiu, Z., Weizhi, W., Jiye, L., Deyu, L.: Theory and Method of Rough Sets, Science Publishes, 2001.7
7. Jia-Yuarn, G.: Rough set-based approach to data mining, PhD dissertation, Department of Electrical Engineering and Computer Science, Case Wester University, USA (January 2003)
8. Xiaohua, H.: Knowledge discovery in database: An Attribute-oriented rough set approach (Rules, Decision Matrices), PhD dissertation, The University of Regina (Canada) (1995)
9. Jue, W., Duoqian, M.: Analysis on Attribute Reduction Strategies of Rough Set. Journal of Computer Science & Technology 13(2), 189–193 (1998)
10. Zhongzhi, S.: Knowledge Discovery. Tsinghua University Press, Beijing 2002.1

Mining Mass Spectrometry Database Search Results—A Rough Set Approach

Jianwen Fang[1,*] and Jerzy W. Grzymala-Busse[2]

[1] Bioinformatics Core Facility
and
Information and Telecommunication Technology Center
University of Kansas, Lawrence, KS 66045, USA
jwfang@ku.edu

[2] Department of Electrical Engineering and Computer Science, University of Kansas,
Lawrence, KS 66045, USA
and
Institute of Computer Science Polish Academy of Sciences, 01-237 Warsaw, Poland
jerzy@ku.edu
http://lightning.eecs.ku.edu/index.html

Abstract. This paper reports results of experiments on mass spectrometry database search results produced by Keller *et al.* This data set describes human proteins. Data mining was conducted using the LERS system. First, the data set was discretized by a cluster analysis algorithm based on agglomerative approach. Then the basic rule set was induced by the LEM2 algorithm. Finally, the rule set was refined using changing rule strength methodology and truncation of the rule set. Our results reach the level of sensitivity and specificity of competing methods. However, our results are explainable since they are in a form of rules and, additionally, we can interpret the role of important features.

1 Introduction

With the advance of soft ionization technologies of electrospray (ES) and matrix-assisted laser desorption ionization (MALDI), tandem mass spectrometry (MS/-MS) with database search has emerged as the method of choice for the identification of proteins in high-throughput proteomics studies. Such an approach usually starts with protein separation using 2D-gel or other technologies. The isolated proteins are then digested to peptides using proteases such as trypsin. The resulting peptides are fragmented and ionized using either ES or MALDI technology. The recorded mass spectra are compared to theoretical ones computed from all possible peptides obtained from a protein sequence database using database search software such as SEQUEST [16], Mascot [14], ProteinProspector [3] and X!Tandem [4]. The spectra are then assigned to peptides that best match theoretical spectra. Most of these programs use scores to rank the candidate peptides

[*] This research has been partially supported by the K-INBRE Bioinformatics Core, NIH grant P20 RR016475.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 340–349, 2007.

that indicate the degree of agreement between spectra and assigned peptides. A validation procedure is generally required to discriminate false positives in the assigned peptides due to the imperfect nature of these search algorithms. This can be done by manual inspection of an expert or by applying empirical filtering criteria based on database search scores and properties of the assigned peptides, such as the number of tryptic termini. However, the manual validation is prohibitively time-consuming when the database is large and the filtering criteria are not reliable and may miss a large number of true positives. We have found that it is common to miss 30–50% of true positives in the tests as presented in Table 1.

Table 1. The performance of conventional filtering approaches, where *charge* denotes peptide charge

Filtering method	Sensitivity	Specificity
XCorr ≥ 2, ΔCn ≥ 0.1, SpRank ≤ 50, NTT $= 2$	0.567	0.99844
XCorr ≥ 2, ΔCn ≥ 0.1, SpRank ≤ 50, NTT ≥ 1	0.732	0.99290
charge $= +1$, XCorr ≥ 1.5, NTT $= 2$ OR charge $= +2$ OR charge $= +3$, XCorr ≥ 2.0, NTT $= 2$	0.572	0.99796
ΔCn > 0.1 AND (charge $= +1$, XCorr ≥ 1.9, NTT $= 2$ OR (charge $= +2$ AND (XCorr ≥ 3 OR $2.2 \leq$ XCorr ≤ 3.0, NTT ≥ 1)) OR charge $= +3$: XCorr ≥ 3.75, NTT ≥ 1)	0.641	0.99514
ΔCn ≥ 0.08 AND (charge $= +1$, XCorr ≥ 1.8 OR charge $= +2$, XCorr ≥ 2.5 OR charge $= +3$, XCorr ≥ 3.5)	0.555	0.99718
ΔCn ≥ 0.1 AND (charge $= +1$, XCorr ≥ 1.9, NTT $= 2$ OR charge $= +2$, XCorr ≥ 2.2, NTT $= 1$ OR charge $= +3$, XCorr ≥ 3.75, NTT $= 1$)	0.567	0.99825
ΔCn ≥ 0.1, SpRank ≤ 50, NTT ≥ 1, AND (charge $= +1$ not included OR charge $= +2$, XCorr ≥ 2.0 OR charge $= +3$, XCorr ≥ 2.5)	0.712	0.99494

In the past several years there have been several attempts to develop software tools using statistical and machine learning algorithms to validate database search hits and consequently improve the results [1,11,15]. Keller *et al.* were among the first to use these approaches to classify the results of SEQUEST searches [11]. They formulated a new metric based on SEQUEST scores that takes into consideration the length of peptide and penalizes lower ranker and poor mass accuracy. Anderson *et al.* used Support Vector Machine (SVM), a powerful machine learning algorithm, to classify SEQUEST peptide assignment as correct and incorrect, also based on SEQUEST scores [1]. They found that SVM yielded fewer false positives and false negatives comparing to conventional cutoff approaches. Very recently, Ulintz *et al.* used SVM, boosting and Random Forest (RF) to classify MS/MS database search results using SEQUEST and Spectrum Mill, a search engine based on ProteinProspector algorithms [15]. All three algorithms improved sensitivity and specificity considerably over conventional cutoff approaches. While all these approaches delivered better performance than conventional filtering approaches, they failed to provide details how the improvements were achieved, as all methods used in previous studies belong to "black-box" approaches. In this study, we sought to develop interpretable classifiers based on rough set theory. The classifiers resulted in rules that can be readily examined by biomedical researchers to further improve database search engines.

2 Data Set

The original experimental dataset was generated by Keller *et al.* as described in [11]. This dataset was also used by Ulintz *et al.* in their data validation studies [15]. In brief, these data were generated in a ThermoFinnigan ion trap mass spectrometer from twenty-two different LC/MS/MS runs on mixtures of eighteen proteins mixed in varying concentrations. Overall 37044 spectra were generated in the experiments. These spectra were then searched by SEQUEST against a protein database that was composed from human protein database with eighteen additional known proteins. Only top-scoring peptides were retained in the database search. Peptides matching the known eighteen proteins were considered as true positives and the remaining top hits were negatives. For direct comparison, we retained the same division of the dataset into training and test datasets as in [15]. We also used the fifteen descriptive features as in [15], see Table 2.

Usually, in the medical field, the problem is to diagnose a specific disease, where all cases affected by the disease are defined as elements of the primary class. Any subset of the set of all cases, defined by the same value of the decision is called a *class* (or *concept*). All remaining cases are defined as elements of a secondary class (healthy patients). Diagnosis is characterized by *sensitivity* (the conditional probability of the set of correctly diagnosed cases from the primary class given the primary class) and by *specificity* (the conditional probability of the set of correctly diagnosed cases from the secondary class given the secondary class). Thus the sensitivity is the ratio of the number of true positives to the sum of the numbers of true positives and false negatives, while specificity is the ratio of the number of true negatives to the sum of the numbers of true negatives and false positives.

Table 2. Descriptive features used in the study

Feature name	Description
Delta	Parent ion mass error
Charge	Parent ion charge
Intensity	Normalized intensity of the peaks
Length	Length of the peptide
Matching peptide	The number of peptides matching the parent ion mass within the mass tolerance
Sp	Preliminary score
SpRank	Rank based on Sp
ΔCn	Difference in normalized correlation scores between next-best and best hits
XCorr	Cross-correlation score
ratio	Fraction of experimental ions matched with the theoretical ions
N_pro	Number of prolines
N_arg	Number of arginines
C_term	C-terminal residue
NTT	Number of tryptic termini
PMF	Proton mobility factor

Our training data set contained 25931 cases, with 1930 cases being the primary class and remaining 24001 cases being the secondary class. The testing data set contained 11113 cases, distributed into 827 cases from the primary class and 10286 cases from the secondary class.

3 Discretization, Rule Induction and Classification

All numerical attributes were discretized before rule induction, i.e., numerical values of these attributes were converted into symbolic. For our experiments we selected a discretization based on cluster analysis. First clusters were formed, using bottom-up (agglomerative) approach. The process was continued until each elementary set, defined by all attributes, was contained in some concept or all attributes defined the same indiscernibility relation as for the original data set. Both ideas, of the *elementary set* and *indiscernibility relation*, are taken from rough set theory [12, 13]. Then the clusters were projected on numerical attributes and initial intervals were created. Finally, these intervals were merged together using the same criterion to stop as in the process of forming clusters.

For rule induction, classification, and validation we used the data mining system LERS (Learning from Examples based on Rough Sets) [5, 6]. After discretization, in the next step of processing the input data file, LERS checks if the input data file is consistent. If the input data file is inconsistent, LERS computes lower and upper approximations of all classes. The ideas of *lower* and *upper approximations* are fundamental for rough set theory [12,13].

In general, LERS uses two different approaches to rule induction: one is used in machine learning, the other in knowledge acquisition. In machine learning, or more specifically, in learning from cases (examples), the usual task is to learn the smallest set of minimal rules, describing the class. To accomplish this goal LERS uses two algorithms: LEM1 and LEM2 (LEM1 and LEM2 stand for Learning from Examples Module, version 1 and 2, respectively). In our experiments we used only LEM2 algorithm since, in general, LEM2 induces simpler and more accurate rule sets.

The classification system of LERS is a modification of the *bucket brigade algorithm* [2,10]. The decision to which concept a case belongs is made on the basis of two factors: strength and support. They are defined as follows: *strength* is the total number of cases correctly classified by the rule during training. The second factor, *support*, is defined as the sum of strengths for all matching rules from the concept. The concept C for which the support, i.e., the following expression

$$\sum_{matching\ rules\ R\ describing\ C} Strength(R)$$

is the largest is the winner and the case is classified as being a member of C.

In the classification system of LERS, if complete matching is impossible, all partially matching rules are identified. These are rules with at least one attribute-value pair matching the corresponding attribute-value pair of a case. For any partially matching rule R, the additional factor, called *Matching _ factor* (R), is computed. Matching_factor (R) is defined as the ratio of the number of matched attribute-value pairs of R with a case to the total number of attribute-value pairs of R. In partial matching, the concept C for which the following expression is the largest

$$\sum_{\substack{partially\ matching\\ rules\ R\ describing\ C}} Matching\_factor(R) * Strength(R)$$

is the winner and the case is classified as being a member of C.

Every rule induced by LERS is preceded by three numbers: the total number of attribute-value pairs on the left-hand side of the rule, strength, and the rule domain size, i.e., the total number of training cases matching the left-hand side of the rule.

4 Postprocessing of Rules

Once rule sets were induced we used two different postprocessing techniques applied to these rule sets. The first technique was called *increasing rule strengths* [7, 8]. This technique is used for imbalanced data sets, that is, data sets with different class sizes. Our data set was imbalanced, the total size of primary class was much smaller than the total size of secondary class. In such data, during classification of unseen cases, rules matching a case and voting for the primary classes are outvoted by rules voting for the bigger, secondary classes. Thus the diagnosis of a primary classes is poor and the resulting classification system would be rejected by diagnosticians.

Therefore it is necessary to decrease the error rates for the primary class. Since the data set is imbalanced, the simplest idea is to add cases to the primary class in the data set, e.g., by adding duplicates of the available cases. The total number of training cases will increase, hence the total running time of the rule induction system will also increase. Adding duplicates will not change the knowledge hidden in the original data set, but it may create a balanced data set so that the average rule set strength for both classes will be approximately equal. The same effect may be accomplished by increasing the average rule strength for the primary class. In our research we selected the optimal rule set by multiplying the rule strength for all rules describing the primary class by the same real number called a *rule strength multiplier*. In general, the error rates for the primary classes decrease with the increase of the rule strength multiplier. At the same time, the error rates for the secondary classes increase.

The second mechanism to increase the conditional probabilities for primary class was *rule truncation*, a method of reducing the rule set by deleting weak rules, describing a few training cases, by removing rules with strengths not exceeding some cutoff. The truncation algorithm was already used for diagnosis of melanoma, see, e.g., [8]. By removing weak rules the total number of rules describing the class is reduced. This may result in rules that may not match the cases completely as they would have before the truncation process. However, the LERS classification system is equipped with partial matching. A case may still be very closely related to the correct class and thus may be correctly recognized.

5 Experiments

Our experiments were performed on the training data set (with 25931 cases) discretized by the agglomerative cluster analysis algorithm. A basic rule set was induced from the discretized data set by the LEM2 algorithm. Then we incrementally increased the rule strength multiplier for all rules describing the primary classes, see Table 3. Sensitivity and specificity presented in Table 3 were computed using the testing data set (with 11113 cases). During these experiments the truncation cutoff was not used (all rules participated in classification). Then, with the rule strength multipler equal to 1000, we gradually increased the truncation cutoff, up to 100, for the rule set describing the secondary class. The size of the rule set describing the secondary class decreased from 282 (the original rule set) to 141 (the rule set corresponding to the truncation cutoff equal to 100). During all of our experiments the size of the rule set describing the primary class was always equal to 244. The ROC (Receiver Operating Characteristic) graph, illustrating our experiments, is presented in Figure 1.

Table 3. Performance of rough set models

Strength multiplier	Truncation cutoff	Sensitivity	Specificity
1	0	0.8440	0.99543
20	0	0.8839	0.99217
100	0	0.9117	0.98493
500	0	0.9178	0.98260
1000	0	0.9190	0.98085
1000	5	0.9202	0.97968
1000	20	0.9287	0.97832
1000	50	0.9323	0.97579
1000	100	0.9383	0.96850

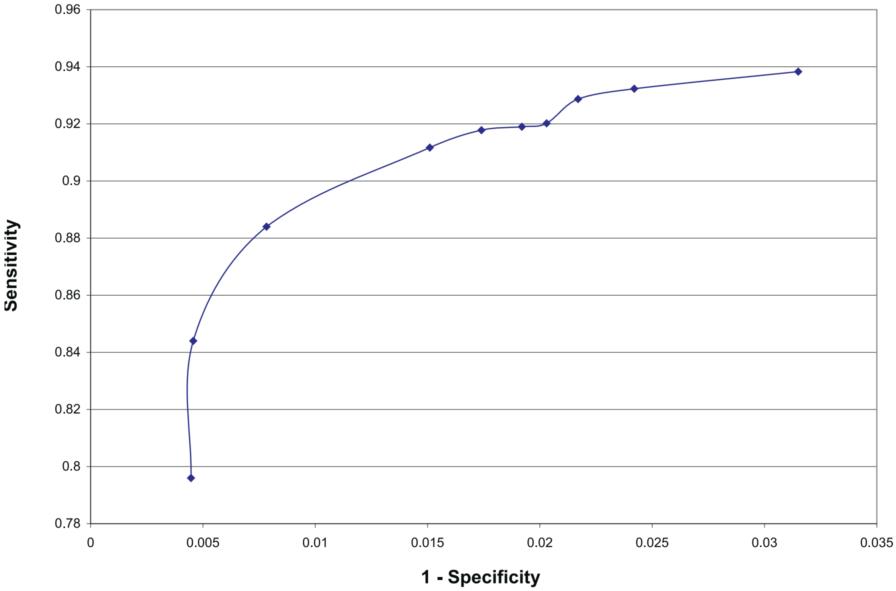


Fig. 1. ROC graph

6 Results and Comparison with Other Approaches

The dataset that was the subject of our experiments was previously analyzed in other studies using various machine learning algorithms [11,15]. For example, Ulintz *et al.* reported that approaches using boosting and random forest achieved a sensitivity of 0.99, PeptideProphet and SVM delivered 97 – 98% sensitivity at a false positive rate of roughly 0.05 [15]. Thus the performance of our approaches is comparable to Ulintz's results as we achieved better false positive rates but poorer sensitivities. Keller *et al.* reported a sensitivity of 89% with an error of

2.5% [11]. Although a direct comparison to this study is not applicable because Keller *et al.* used different division of training and test datasets, it appears that our model is competitive.

7 Interpretation of the Decision Rules

An advantage of white-box approaches such as rough set theory over "black-box" methods is that the detailed knowledge of the classification process is available for better understanding the problem under study. In the present study, the classification rules discovered by our classifiers reveal several important observations leading to better understanding the chemistry underlying the molecule fragmentation and ionization. For example, the mobile proton factor (MPF) was discovered as a very useful indicator. A single rule involving only two features can eliminate approximately 40% of true negatives without error:

(PMF, 0.699..5.5) & (C_term, others) −> (label, -1)

The PMF is calculated as:

$$\frac{R + 0.8 * K + 0.5 * H}{charge}$$

where R is the number of arginine, K is the number of lysine, and H stands for the number of histidine. Charge means the charge on the parent peptide. Although it was known that a smaller value of PMF indicates higher protein mobility [15], it was unclear the degree that PMF would affect the peptide detection using MS/MS technologies. From our results, it seems that PMF is particularly useful to eliminate peptides with a terminal residue other than arginine and lysine. It is worth to note that the rule does not use any SEQUEST score.

NTT (the number of tryptic terminals) is important since the peptides are the products from tryptic digestions. It measures whether the peptide is fully tryptic (NTT = 2), partially tryptic (NTT = 1), or non-tryptic (NTT = 0). However, the NTT of a fully tryptic terminal peptide can be equal to one. NTT was found as the most important attribute in Ulintz's study [15]. A higher NTT is a strong indication of a true positive; however, the NTT of a small portion of true positives is either 0 or 1. For example, this type of peptides accounts for about one quarter of the true positives in our dataset. Thus improvement in this type of peptide identification will significantly increase the sensitivity and specificity. We found that the following single rule correctly classifies approximately 40% of these partially tryptic and non-tryptic peptides. Thus peptides with lower NTT but higher XCorr and ΔCn are likely true positives.

(XCorr, 3.4218..7.2792) & (ΔCn, 0.2362..0.5565) & (NTT, 0..1.5) −> (label, 1)

Most of the rules discovered in our study involve one or more features that are not SEQUEST scores. These features are either peptide physicochemical properties (e.g., MPF, Length, etc.) or protein sequence environment (e.g., NTT). The results further confirm the conclusion in our recent study that these properties can be used to improve data validation models [5].

8 Conclusions

We have proposed a rough set based approach to validate MS/MS database search results. The performance of our approach is comparable to competing methods. However, some important rules discovered in this study may lead to better understanding of the chemistry underlying the molecule fragmenttion and ionization. In addition, these rules may be used in the development of novel mass spectrometry database search engines.

References

1. Anderson, D.C., Li, W., Payan, D.G., W.S., N.: A new algorithm for the evaluation of shotgun peptide sequencing in proteomics: support vector machine classification of peptide MS/MS spectra and SEQUEST scores. J. Proteome. Res. 2, 137–146 (2003)
2. Booker, L.B., Goldberg, D.E., Holland, J.F.: Classifier systems and genetic algorithms. In: Carbonell, J.G. (ed.) Machine Learning. Paradigms and Methods, pp. 235–282. MIT Press, Menlo Park, CA (1990)
3. Clauser, K.R., Baker, P.R., Burlingame, A.L.: Role of accurate mass measurement (+/− 10 ppm) in protein identification strategies employing MS or MS/MS and database searching. Anal. Chem. 71, 2871–2882 (1999)
4. Craig, R., Ronald, C., Beavis, R.C.: TANDEM: matching proteins with mass spectra. Bioinformatics 20, 1466–1467 (2004)
5. Fang, J.W., Dong, Y.H., Williams, T.D., Lushington, G.H.: Classification of MS/MS Peptide Identifications and Its Application in Data Validation (Submitted)
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.: MLEM2: A new algorithm for rule induction from imperfect data. In: Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2002, Annecy, France, July 1–5, 2002, 243–250 (2002)
8. Grzymala-Busse, J.W., Hippe, Z.S.: Postprocessing of rule sets induced from a melanoma data set. In: Proceedings of the COMPSAC 2002, 26th Annual International Conference on Computer Software and Applications, Oxford, England, August 26–29, 2002, pp. 1146–1151 (2002)
9. Grzymala-Busse, J.W., Goodwin, L.K., Grzymala-Busse, W.J., Zheng, X.: An approach to imbalanced data sets based on changing rule strength. In: Learning from Imblanced Data Sets, AAI Workshop at the 17th Conference on AI, AAAI-2000, Austin, TX, July 30–31, 2000, pp. 69–74 (2000)
10. Holland, J.H., Holyoak, K.J., Nisbett, R.E.: Induction. Processes of Inference, Learning, and Discovery. MIT Press, Menlo Park, CA (1986)
11. Keller, A., Nesvizhskii, A.I., Kolker, E., Aebersold, R.: Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. Anal Chem. 74, 5383–5392 (2002)
12. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
13. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)

14. Perkins, D.N., Pappin, D.J., Creasy, D.M., Cottrell, J.S.: Probability-based protein identification by searching sequence databases using mass spectrometry data. Electrophoresis 20, 3551–3567 (1999)
15. Ulintz, P.J., Zhu, J., Qin, Z.S., Andrews, P.C.: Improved classification of mass spectrometry database Search results using newer machine learning approaches. Mol. Cell. Proteomics 5, 497–509 (2006)
16. Yates III, J.R., Eng, J.K., McCormack, A.L.: An approach to correlate tandem mass spectral data of peptides with amino acid sequences in the protein database. J. Am. Soc. Mass. Spectrom. 5, 976–989 (1994)

Rough Set Approach to Spam Filter Learning

Mawuena Glymin and Wojciech Ziarko

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
ziarko@cs.uregina.ca, moe.glymin@uregina.ca

Abstract. This article presents an elementary overview of techniques employed for spam detection via probabilistic decision table-based predictive data modelling. The focus here is to present a solution that combines simple algorithms together with some heuristics to construct generalized rough approximations of spam and legitimate e-mails using the variable precision rough set (VPRSM) approach. Experiments were conducted to explore the application of VPRSM for designing an intelligent agent for spam filtering.

1 Introduction

Spamming is the abuse of electronic messaging systems to send unsolicited, undesired bulk messages. A person engaged in spamming is known as a spammer. E-mail spam is the most common form of spam - a term which is applicable to similar abuses in other media: instant messaging spam, mobile phone messaging spam, web search engine spam,etc.

E-mail spam, otherwise known as unsolicited bulk (or commercial) e-mail or junk mail may be defined as one or more nearly identical messages sent out to a large group of recipients who have not requested it and have no use of the information being relayed in the message. E-mail spam targets individual users with direct mail messages. Because of the very low cost associated with sending e-mails, spammers are able to send millions of e-mails daily over the internet. E-mail addresses on spam lists are often created by scanning Usenet postings, stealing Internet mailing lists, dictionary attacks or searching the web for addresses, amongst others.

Spam filtering is an automatic analysis and classification of incoming e-mail messages, to allow the disqualification of spam from inclusion in a group of legitimate messages for a particular user. Generations of spam filters have emerged over the years to deal with the spam issue. Most of these filters succeeded to some point in discriminating between spam and legitimate e-mails, however they require manual intervention. For example content based methods require human efforts to build lists of characteristics and their scores. Over the last five years, statistical filters have gained more attention as they are able to tweak themselves; getting better and better with less manual intervention. The most popular statistical approach is the Bayesian filter, which assigns probability estimates to e-mails. Even such filters have their limitations as spammers still manage to

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 350–359, 2007.

evade them by using various exploiting techniques [2]. Consequently, novel approaches are desired to deal with ever-increasing flood of spam and the persistent attempts by spammers to break the existing anti-spam barriers.

In this article, we report our research and experiments on the application of a Rough Set-based analysis in conjunction with the probabilistic framework of the variable precision rough set model (VPRSM) [3] [6] [9] for personalized adaptive spam filtering. In our approach, the learned spam detection rules are structured into hierarchies of probabilistic decision tables [8], which offer the advantage of uniform rule size and the ability to control the progress of learning. This decision table-based approach has the potential to lead to more powerful application technology for a solution to the spam problem by broadening content-based filtering with a system that has the real advantage to combine personalized and robust spam combating heuristics and the ability to learn. Other advantages include taking the whole message into account in the rule derivation, as generated from corpora of categorized messages rather than involving direct human effort in rule development and tuning.

We discuss our experiences with a system based on findings from research and experiments into ways of using VPRSM for detecting spam. In what follows, we present the rough set fundamentals of the system's operation and the results of our experiments with system's application to spam detection.

2 Information Representation

We assume that observations about objects of interest (e.g. e-mail messages in our case) are expressed through values of functions, referred to as *attributes*, belonging to the union of two disjoint sets $C \cup D$. The functions belonging to the set C are called *condition attributes*, whereas functions in D are referred to as *decision attributes*. In the context of our application, the condition attributes are predefined e-mail properties which contribute to spam detection. We will assume that there is only one binary-valued decision attribute, that is $D = \{d\}$ representing the classification of a message as spam or legitimate e-mail. Technically, each attribute a belonging to $C \cup D$ is a mapping $a : U \rightarrow V_a$, where V_a is a finite set of values called the *domain* of the attribute a. In our case, the domains of condition attributes are also binary, reflecting the presence or absence of a property. Some binary condition attributes are obtained through a discretization process, in which the range of numeric measurements is divided into a number of subranges by suitably selected cut points. The cut points are then used as two valued attributes representing the ranges of values below or above the cut.

Each subset of attributes $B \subseteq C \cup D$ also defines a mapping, denoted as $\mathbf{B} : U \rightarrow \mathbf{B}(U) \subseteq \otimes_{a \in B} V_a$, where $\otimes$ denotes Cartesian product operator of all domains of attributes in B. For a tuple $z \in \mathbf{C} \cup \mathbf{D}(U)$ and a subset of attributes $B \subseteq C \cup D$, the projection $z.B$ corresponds to a set of objects $\mathbf{B}^{-1}(z) = \{e \in U : \mathbf{B}(e) = z\}$ whose values of attributes in B match $z.B$. The family of sets $\mathbf{B}^{-1}(z)$ (for different tuples z) forms a partition of the universe U. The partition will be denoted as U/B and its classes will be called *B-elementary sets*. In particular, the $C \cup D$-elementary sets, denoted as $G \in U/C \cup D$, will be referred to as *atoms.* The pair $(U, U/C)$ will be called an *approximation space.* The C-elementary sets $E \in U/C$ will be referred to as *elementary sets* and the D-elementary sets $F \in U/D$ will be called *decision categories.* In the application discussed in this article, there are only two decision categories corresponding to the target set X of spam e-mails, and its complement $\neg X$ of legitimate e-mails.

3 Variable Precision Rough Sets

We assume here that the universe of interest U (the collection of all possible e-mail messages) can be partitioned into finite collection of atoms in terms of condition and decision attributes. In addition, it is assumed that all subsets $X \subseteq U$ under consideration are measurable with a probability measure $0 < P(X) < 1$. This *prior probability* of a set X, $P(X)$ can be estimated from data by calculating its frequency in a sample in the standard way. We also assume the existence of conditional probabilities $P(X|E)$ representing the likelihood of occurrence of decision category X, or $\neg X$, relative to the occurrence of an elementary set E.

The VPRSM extends upon original rough set ideas introduced by Pawlak [5] by allowing parametric definitions of rough approximation regions (i.e. positive, negative and boundary regions of a decision category X of interest in the probabilistic framework.

Informally, the VPRSM defines the positive region of a set X as an area where the certainty degree of an object's membership in the decision category X is relatively high, the negative region as an area where the certainty is relatively low, and the boundary area where the certainty of an object's membership in the target set is not sufficiently high and not sufficiently low. The defining criteria in the VPRSM are expressed in terms of conditional probabilities $P(X|E)$ and of the prior probability $P(X)$ of the decision category X. The *precision control* parameters define the approximation regions as follows.

The *lower limit* l parameter, satisfying the constraint $0 \leq l < P(X) < 1$, represents the highest acceptable degree of the conditional probability $P(X|E)$ to include the elementary set E in the *negative region* of the set X. That is, the *l-negative region* of the set X, denoted as $NEG_l(X)$ is defined by $NEG_l(X) = \cup\{E : P(X|E) \leq l\}$.

The second parameter, referred to as the *upper limit* u, satisfying the constraint $0 < P(X) < u \leq 1$, defines the *u-positive region* of the set X. The upper limit reflects the least acceptable degree of the conditional probability $P(X|E)$ to include the elementary set E in the positive region, or *u-lower approximation* of the set X. The u-positive region of the set X, $POS_u(X)$ is defined as $POS_u(X) = \cup\{E : P(X|E) \geq u\}$.

The objects which are not classified as being in the u-positive region nor in the l-negative region belong to the (l, u)-boundary region of the decision category X, denoted as $BNR_{l,u}(X) = \cup\{E : l < P(X|E) < u\}$.

The boundary is a specification of objects about which it is known that their associated probability of belonging, or not belonging to the decision category X, is not significantly different from the prior probability of the decision category $P(X)$.

4 Probabilistic Decision Table Hierarchies

The probabilistic decision tables approximately represent probabilistic relations between condition and decision attributes [8]. They arise from the idea of decision table acquired from data introduced by [5]. For the given decision category $X \in U/D$ and the given lower and upper limit parameters l and u, we define the *probabilistic decision table* $DT^{C,D}_{l,u}$ as a mapping $C(U) \rightarrow \{POS, NEG, BND\}$. The mapping is assigning each tuple of values of condition attributes $t \in \mathbf{C}(U)$ its unique designation of one of VPRSM approximation regions $POS_u(X)$, $NEG_l(X)$ or $BND_{l,u}(X)$, the corresponding elementary set E_t is included in, the elementary set E_t probability $P(E_t)$ and conditional probability $P(X|E_t)$:

$$DT^{C,D}_U(t) = \begin{cases} (P(E_t), P(X|E_t), POS) \Leftrightarrow E_t \subseteq POS_u(X) \\ (P(E_t), P(X|E_t), NEG) \Leftrightarrow E_t \subseteq NEG_l(X) \\ (P(E_t), P(X|E_t), BND) \Leftrightarrow E_t \subseteq BND_{l,u}(X) \end{cases} \quad (1)$$

An example probabilistic decision table is shown in Table 1, given $l = 0.1$ and $u = 0.8$.

Table 1. Partial Probabilistic Decision Table

E	C_1 flag spam exploits	C_2 flag reply exists	C_3 longest word length	C_4 No. recip-ients	No. e-mails	No. spam	No. legit	$P(E_t)$	$P(X\|E)$ X=*spam*	VPRSM Region
E_1	0	1	1	1	24	16	8	0.3	0.67	BND
E_2	0	0	0	1	24	1	23	0.3	0.04	NEG
E_3	1	0	1	0	32	29	3	0.4	0.9	POS
...	...	...	...	...	...	...	...	...	...	...

To represent the degree of relative coverage of the universe by lower approximations of decision categories, Pawlak introduced the notion of attribute dependency, referred here as γ-dependency [4][5]. The $\gamma(D|C)$-dependency provides a useful measure for evaluating the quality of decision tables learned from data. The $\gamma(D|C)$-dependency can be extended to the variable precision model of rough sets and used to represent the relative size of the boundary area of a decision table. In the extended version it is defined as the combined probability of the u-positive and l-negative regions:

$$\gamma_{l,u}(X|C) = P(POS_u(X) \cup NEG_l(X)). \quad (2)$$

The probabilistic decision tables can be structured into hierarchies by treating the boundary region $BND_{l,u}(X)$ of a decision table as a sub-universe of U, denoted as $U' = BND_{l,u}(X)$, based on which another decision table can be constructed using some condition attributes C', disjoint from C. As in the universe U, in the approximation space $(U', U'/C')$, the decision table for the subset $X' \subseteq X$ (the target decision category in U'), $X' = X \cap BND_{l,u}(X)$ can be constructed according to the formula (1). By repeating this step recursively, a linear hierarchy of probabilistic decision tables can be grown until either boundary area disappears in one of the constructed decision tables, or no attributes can be identified to produce non-boundary decision table at the final step. The algorithmic details of the decision table hierarchy construction for the spam detection application are described in Section 6.

The hierarchy of decision tables can be used as an approximate classifier, or predictor, classifying a new object as a member, or non-member of the decision category X (i.e. as spam or legitimate message in our application). For objects falling into positive or negative regions of the decision category, the classifications produced by the properly constructed and validated hierarchy of decision tables will result in significantly higher decision certainty, or significantly lower misclassification rate as compared to decisions guided only by prior probabilities $P(X)$ and $\neg P(X)$.

5 Spam Detection System Development

The spam detection system operation is divided into two phases: a training phase and a classification phase. In the training phase, rough set-based machine learning is used on the collected sets of pre-classified e-mails (hypertext source codes) for decision table hierarchy-based model construction. In the classification phase, learnt decision table hierarchy is used to predict the decision category of incoming e-mails.

We considered e-mails collected from hotmail accounts to train and test the decision model. The process involves manual retrieval and categorical labelling of hypertext documents as a complete representation of actual e-mails. Each training corpus contained at least 4200 e-mails consisting of approximately 30% of legitimate and 70% of spam messages to reflect actual inflow of e-mail. Spam e-mails were collected from multi-user inboxes however, for better results and also to reduce impacts of misclassification. Legitimate e-mails were gathered from a single highly active user account. The training corpora were collected over a 2 year period (2004/2005).

Prior to VPRSM-based modelling we formalized an extraction protocol for governing the standard representation of e-mails. Data objects (e-mails) were presented as feature vectors for training and testing. Queries were implemented as functions for the extraction of interesting information *features* from the messages to construct condition attributes[2,1]. We identified a total of 58 features to be extracted from each e-mail. These were used to construct feature vectors, representing the universe of collected e-mails. Tools used to facilitate representation were SQL, Visual Basic and Perl.

In the process of feature extraction, we scanned all messages to breakdown e-mails according to predefined areas, creating representational views for message header and body analysis. To capture the structure of an e-mail we divided it into small components called *tokens* to be further analyzed. Database tables were deployed to hold individual tokens for thorough token and global structural analysis. Some common tokens included punctuation symbols, sequences of characters, spaces, tabs, carriage returns, line feeds, and entities such as e-mail addresses, images, symbols, html tags, attachments and URLs. For each token, several varying characteristic attributes were collected such as it's position, size, length, case (upper, lower, mixed or proper) for word tokens, character (alpha, numeric or alphanumeric)and alphabetical composition (vowels/consonants) for sentence tokens, and more.

Next, we performed analysis on each identified area to gather useful information obtained by posing queries about some defined concepts (corresponding to condition attributes). We employed heuristics to analyze both exploits by spammers and indicators of spam or non-spam. The analysis performed at this stage included:

- Identification of possible characteristic attributes of individual tokens or token groups, to be used to develop queries that provide a summary or descriptive view of some defined concepts.
- Investigation of structure, meaning, positioning, and proportional relations between categories of certain token types given by definitions from a formal grammar. For example we performed a ratio analysis of distinct cases (upper, lower, mixed or proper) as presented by alpha word tokens for a globalized conceptual view on casing.
- Feature aggregation was employed for simpler concept description to enhance the discriminative power of rules generated. This process attempted to present more useful features by constructing compound features using constructive operators on multi elementary features. For example, consider two common exploits in spam: URL encoding (queries will look for hexadecimal URL, hiding target URL with an @ sign, etc.) and character encoding (&#109;ortgage renders into Mortgage), as features. Although both elementary features are investigated separately using distinct feature queries, we may perform an *OR* operation on the boolean output of both queries to generate a new compound feature, represented as the conditional attribute.

No feature ranking was employed at this stage since the hierarchy structure using VPRSM picks the best attributes per hierarchy level. A few examples of some queries used to defined feature concepts are provided below:

- **Spammy**
  Investigate if e-mail contains spam words, phrases, or the like e.g dear friend, click here, debt financing, viagra, etc.

- **Exploits in Spam**
Check body of e-mail for exploits such as URL obscuring, word obscuring, domain spoofing, white-out, token breaking, etc.
- **String Analysis (subject and body)**
Length of field, length of longest word in a field, number of words in field, number of sentences in field, number of images in body, etc.
- **Header Analysis**
Does top-level domain (eg .ca) of e-mail addresses exist in the full list of identified Internet top-level domains? Do usernames or hostnames of e-mail addresses conform to that only permissible when creating an e-mail account? Is sender the same as recipient? Attachments with no extension, e.g. file1? Is recipients username found in other fields? Does Reply-to field exist? Is there a subject? Does subject contains defined illegal symbols?
- **Counts & Attachments**
Count number of e-mails addresses in To/From fields, number of comma separated values in cc and bcc fields, number of attachments, total size of all attachments, number of distinct filenames, number of distinct file extensions.
- **Ratio Analysis**
Check case category (proper , mixed, lower, upper, or any combinations) ratios for all word tokens. Calculate character category (alpha, numeric, symbolic, or any combinations) ratio for all characters in a given string. Calculate vowel-consonant ratio for all alphabets in a given string.

Lastly, we applied a functional mapping to convert derived quantitative attributes to qualitative binary attributes for decision table construction. In this process, the range of each numeric attributes was divided into a predefined number of intervals according to required representation resolution level (e.g. 100 intervals) using evenly spaced cut points. Subsequently, each cut point was treated as an attribute by itself to provide a pool of binary attributes to be used to grow the hierarchy of decision tables. In addition, we used some qualitative binary attributes, such as the presence or absence of images in the e-mail, which supplemented the converted quantitative attributes.

6 Learning Hierarchy of Probabilistic Decision Tables

Given a training data set, we used the VPRSM-based hierarchy structured decision tables to represent probabilistic knowledge learned from data. This derived knowledge was applied to construct generalized rough approximations of the target set, the spam. Elementary sets represent general patterns that are automatically discovered from hierarchies of probabilistic decision tables and are used to formulate rules for predicting new incoming mail. The objective here was to classify e-mail objects as *spam* or *legitimate*. The major stages of the algorithm [8] adapted for the generation of the hierarchy of decision tables are presented below.

Algorithm HDTL

```
1. $\mathbb{U} \longleftarrow U$, $\mathbb{C} \longleftarrow C$, $\mathbb{D} \longleftarrow D$
2. Define root layer  decision table $DT_{\mathbb{U}}^{\mathbb{C},\mathbb{D})}$
3. Compute $\gamma$-dependency of current layer  $\gamma(DT_{\mathbb{U}}^{\mathbb{C},\mathbb{D})})$
4. Repeat
   {
5. If  $(\gamma(DT_{\mathbb{U}}^{\mathbb{C},\mathbb{D})}) = 0$ Then
     {
     Output current layer decision table  $DT_{\mathbb{U}}^{\mathbb{C},\mathbb{D}}$
     Stop
     }
6. Output current layer region decision table  $DT_{\mathbb{U}}^{\mathbb{C},\mathbb{D}}$
7. Select new condition attributes on the boundary area $\mathbb{C} \longleftarrow$ select$(BND^{\mathbb{C},\mathbb{D}}(\mathbb{U}))$

8. If  $(\mathbb{C} = \phi)$ Then Stop
9. Define new universe, initiate new layer,  $\mathbb{U} \longleftarrow BND^{\mathbb{C},\mathbb{D}}(\mathbb{U})$
10. Define current layer  decision table $DT_{\mathbb{U}}^{\mathbb{C},\mathbb{D})}$
   }
```

At each layer of the hierarchy, new binary condition attributes are added from the pre-computed pool of cut-based attributes. The number of the attributes added on each level is limited by a user-set parameter to avoid the exponential growth of the decision tables, and also to avoid the classical problem of ”over-fitting” the training data. The selection of the attributes is controlled by a hill-climbing heuristic algorithm attempting to maximize the degree of probabilistic dependency, referred to as normalized λ-dependency $\lambda(D|C)$ [9] between the selected attributes and the decision attribute:

$$\lambda(X|C) = \frac{\sum_{E \in U/C} P(E)|P(X|E) - P(X)|}{2P(X)(1 - P(X))} \tag{3}$$

where $P(X)$ is prior probability of the target set X in the universe U. In practice, the prior probability represents the frequency of spam among e-mail messages.

7 System Evaluation

Data analysis was based on validation and interpretation of the mined rules generated from VPRSM hierarchical modelling. Testing corpora were collected over different time periods preceding training data collection (2005 - 2007). Evaluation of the computational model was measured using accuracy and error indicators defined as follows:

$$accuracy = \frac{B + D}{A + B + C + D} \tag{4}$$

$$error = \frac{A + C}{A + B + C + D} \tag{5}$$

where A denotes the number of objects incorrectly rejected from category X (False Negatives); B is the number of objects correctly rejected from category X (True Negatives); C is the number of objects incorrectly assigned to category X (False Positives); and D is the number of objects correctly assigned to category X (True Positives).

Given different parameter settings for system learning, we present sample evaluations of the proposed system in the table below. In Table 2, n is the number of attributes added on each hierarchy level while constructing the structure of decision tables, and l and u are the VPRSM precision control parameters. Unclassified cases are objects (e-mails) whose attribute value combinations are not matched by any learned rules, using the trained hierarchical model.

Table 2. Results using Test Corpus A of 905 objects as input vector

n	l	u	Unclassified cases	Classified cases	A	B	C	D	Accuracy (%)	Error (%)
8	0.1	0.9	0	905	17	188	20	680	95.91	4.09
8	0.2	0.8	0	905	6	176	32	691	95.80	4.20
8	0.3	0.7	17	888	14	174	17	683	96.51	3.49
10	0.4	0.6	98	807	9	132	8	658	97.89	2.11
8	0.4	0.8	5	900	6	171	32	691	95.77	4.23
8	0	1	0	905	9	175	33	688	95.36	4.64

Overall, the results were encouraging as accuracy measured across the different testing data sets was relatively high (over *95%*) and can be considered acceptable. For almost all tests, we observed low rates of unclassifiable cases, relative to total observations to be classified. The only exception was when the number of attributes per table level was increased to 10. This lead to the increase of unclassified cases to about 10%, which was expected due to related exponential increase in the complexity of learning [7], or the upper bound on the maximum size of the learned decision tables. Experiments show that with consistent retraining and better choice of attributes, higher system accuracy would be attained.

The extraction of rules obtained from boundary regions had the overall impact of increased false classifications. Various tests were performed with or without retraining the model, and with varying precision control parameter settings, different cut points, different elementary set size threshold values and the number of attributes per hierarchy level. The normalized λ-dependency measure for the whole hierarchy structure of decision tables [10] was used to a priori evaluate the resulting classifiers. It was observed that the λ-dependency is the lowest when $u = 0$ and $l = 1$, regardless of other parameter settings. This choice of certainty control parameters also generates the most number of hierarchy levels during the modelling stage. However, the optimization of parameters to provide the highest improvement in prediction remains a topic for further research.

8 Conclusion

We have performed experimental analysis of feasibility of using the hierarchy of VPSRM probabilistic decision tables at heart of a spam detection filter. The investigation involved content analysis of e-mails to construct appropriate attribute-value representation and experimentation with different hierarchy decision table-based e-mail classifiers. The results indicate that the proposed system has a good potential as a spam detection tool due to its demonstrated relatively high accuracy, the ability to learn and efficiency. The future work will involve more experimental evaluation and comparison with existing approaches.

Acknowledgments. The research reported in the article was partially supported by a research grant awarded to the second author by the Natural Sciences and Engineering Research Council of Canada.

References

1. Graham, J.: The Spammers Compendium. 1996-2006 http://www.jgc.org/tsc/
2. Hulten, G., Penta, A., Seshadrinathan, G., Mishra, M.: Trends in spam products and methods. In: Proc. of the First Conference on E-mail and Anti-Spam (CEAS) (2004) http://www.ceas.cc/papers-2004/165.pdf
3. Yao, Y.Y., Wong, S.K.M.: A decision theoretic framework for approximating concepts. International Journal of Man.-Machine Studies 37, 793–809 (1992)
4. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
5. Pawlak, Z.: Rough Sets - Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, Dordrecht (1991)
6. Ziarko, W.: Variable precision rough sets model. Journal of Computer and Systems Sciences 46(1), 39–59 (1993)
7. Ziarko, W.: On learnability of decision tables. In: Proc. of the Third Intl. Conference on Rough Sets and Current Trends in Computing, Uppsala, Sveden, LNAI, vol. 3066, pp. 394–401, Springer, Heidelberg (2004)
8. Ziarko, W.: Incremental Learning and Evaluation of Structures of Rough Decision Tables. In: Transactions on Rough Sets IV. LNCS, vol. 3700, Springer, Heidelberg (2005)
9. Ziarko, W.: Probabilistic rough sets. In: Ślęzak, D., Wang, G., Szczuka, M., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 283–293. Springer, Heidelberg (2005)
10. Ziarko, W.: Partition dependencies in hierarchies of probabilistic decision tables. In: Proc. of the RSKT 2006. LNCS (LNAI), vol. 4062, pp. 42–49. Springer, Heidelberg (2006)

Web-Based Support Systems with Rough Set Analysis

JingTao Yao and Joseph P. Herbert

Department of Computer Science
University of Regina, Regina
Canada S4S 0A2
{jtyao,herbertj}@cs.uregina.ca

Abstract. Rough sets have been applied to many areas where multi-attribute data is needed to be analyzed to acquire knowledge for decision making. Web-based Support Systems (WSS) are a new research area that aims to support human activities and extend human physical limitations of information processing with Web technologies. The applications of rough set analysis for WSS is looked at in this article. In particular, our focus will be on Web-Based Medical Support Systems (WMSS). A WMSS is a support system that integrates medicine practices (diagnosis and surveillance) with computer science and Web technologies. We will explore some of the challenges of using rough sets in a WMSS and detail some of the applications of rough sets in analyzing medical data.

1 Introduction

Web-based Support Systems (WSS) are a completely new frontier for computerized support systems [16]. It can be understood as extensions of existing research in two dimensions. It can also be viewed as natural extensions of decision support systems with the use of the Web to support more activities. In the technology dimension, WSS use the Web as a new platform for the delivery of support with new advances in technology can lead to further innovations in support systems. Along the application dimension, the lessons and experiences from DSS can be easily applied to other domains.

Research on information retrieval support systems [17], research support systems [4,13], decision support systems [5,11], and medical support systems [1,12] are just some of the recent investigations for moving support systems to the Web platform [14,15].

Rough set theory is a way of representing and reasoning imprecision and uncertain information in data [9]. It deals with the approximation of sets constructed from descriptive data elements. This is most helpful when trying to discover decision rules, important features, and minimization of conditional attributes. The beauty of rough sets is how it creates three regions, namely, the positive, negative and boundary regions. The boundary regions are useful for a undeterminable cases.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 360–370, 2007.

Researchers have used rough sets for diagnosing cancer [8], brain disorders [3], lung disease [7], and others. These applications of rough sets to data analysis may be included in a Web-based Medical Support System (WMSS).

This paper will focus on the issues of migrating the rough set model for use in Web-based support systems. The organization of this paper is as follows. Section 2 will discuss rough set theory and an extended probabilistic model that incorporates risk. Section 3 will provide WSS applications with rough sets and introduce Web-based medical support systems with rough set functionality. Finally, we conclude this paper in Section 4.

2 Rough Set Models

2.1 Algebraic Rough Set Model

Approximation is used to characterize a set $A \subseteq U$ [9], where U is a finite, nonempty universe. It may be impossible to precisely describe A given a set relation B. Equivalence classes are simply objects in U in which we have information. Definitions of lower and upper approximations follow:

$$\begin{aligned} \underline{apr}(A) &= \{x \in U | [x] \subseteq A\}, \\ \overline{apr}(A) &= \{x \in U | [x] \cap A \neq \emptyset\}. \end{aligned} \tag{1}$$

The lower approximation of a set A, denoted $\underline{apr}(A)$, is the union of all elementary sets that are included (fully contained) in X. The upper approximation of a set A, denoted $\overline{apr}(A)$, is the union of all elementary sets that have a nonempty intersection with A. This allows us to approximate unknown sets with known objects. We can now define notions of positive, negative, and boundary regions [9] of A:

$$\begin{aligned} POS(A) &= \underline{apr}(A), \\ NEG(A) &= U - \overline{apr}(A), \\ BND(A) &= \overline{apr}(A) - \underline{apr}(A). \end{aligned} \tag{2}$$

2.2 Probabilistic Rough Set Model

The algebraic method has very little flexibility for determining the classification regions. It may not be useful or applicable when majority cases are undeterminable. More flexible models include some probabilistic approaches, namely, variable precision rough sets [20] and decision-theoretic rough sets [18,19].

The decision-theoretic approach may lend itself to a more Web-friendly application for two reasons. First, it calculates approximation parameters by obtaining easily understandable notions of risk or loss from the user [19]. This allows for simpler user involvement instead of having parameters being arbitrarily provided. This is important when users are not qualified to set the parameters and just wish to perform analysis. Second, many types of WSS could make use of cost or risk annotations. We present a slightly reformulated decision-theoretic rough set model in this section, as reported in [18,19].

The Bayesian decision procedure allows for minimum risk decision making based on observed evidence. Let $\mathcal{A} = \{a_1, \ldots, a_m\}$ be a finite set of m possible actions and let $\Omega = \{w_1, \ldots, w_s\}$ be a finite set of s states. Let $P(w_j|\mathbf{x})$ be the conditional probability of an object x being in state w_j given the object description $\mathbf{x}$. Let $\lambda(a_i|w_j)$ denote the loss, or cost, for performing action a_i when the state is w_j.

Object classification with approximation operators can be fitted into this framework. The set of actions is given by $\mathcal{A} = \{a_P, a_N, a_B\}$, where a_P, a_N, and a_B represent the three actions to classify an object into $POS(A)$, $NEG(A)$, and $BND(A)$ respectively. Let $\lambda(a_\diamond|A)$ denote the loss incurred for taking action $a_\diamond$ when an object belongs to A, and let $\lambda(a_\diamond|A^c)$ denote the loss incurred by taking the same action when the object belongs to A^c. This can be given as loss functions $\lambda_{\diamond 1} = \lambda(a_\diamond|A)$, $\lambda_{\diamond 2} = \lambda(a_\diamond|A^c)$, and $\diamond = P$, N, or B.

If we consider the loss function inequalities $\lambda_{P1} \leq \lambda_{B1} < \lambda_{N1}$,that is, the loss incurred by λ_{N1} (false-negative) is more than the losses incurred by both a correct classification (λ_{P1}) and an indeterminant classification (λ_{B1}) we can formulate decision rules based on this division of the universe. The corresponding inequalities $\lambda_{N2} \leq \lambda_{B2} < \lambda_{P2}$, that is, a false-positive (λ_{P2}) has a greater cost than a correct classification (λ_{N2}) and an indeterminant classification (λ_{B2}), can further tell us how the universe is divided. We can formulate the following decision rules (P)-(B) [18] based on the set of inequalities above:

$$\begin{array}{lll}
\text{(P)} & \text{If } P(A|[x]) \geq \gamma \text{ and } P(A|[x]) \geq \alpha, & \text{decide } POS(A), \\
\text{(N)} & \text{If } P(A|[x]) \leq \beta \text{ and } P(A|[x]) \leq \gamma, & \text{decide } NEG(A), \\
\text{(B)} & \text{If } \beta \leq P(A|[x]) \leq \alpha, & \text{decide } BND(A),
\end{array}$$

where,

$$\begin{aligned}
\alpha &= \frac{\lambda_{P2} - \lambda_{B2}}{(\lambda_{B1} - \lambda_{B2}) - (\lambda_{P1} - \lambda_{P2})}, \\
\gamma &= \frac{\lambda_{P2} - \lambda_{N2}}{(\lambda_{N1} - \lambda_{N2}) - (\lambda_{P1} - \lambda_{P2})}, \\
\beta &= \frac{\lambda_{B2} - \lambda_{N2}}{(\lambda_{N1} - \lambda_{N2}) - (\lambda_{B1} - \lambda_{B2})}.
\end{aligned} \tag{3}$$

The α, β, and γ parameters define our regions, giving us an associated risk for classifying an object. The α parameter can be considered the division point between the POS region and BND region. Likewise, the β parameter is the division point between the BND region and the NEG region. When $\alpha > \beta$, we get $\alpha > \gamma > \beta$ and can simplify the rules (P-B) into (P1-B1):

$$\begin{array}{lll}
\text{(P1)} & \text{If } P(A|[x]) \geq \alpha, & \text{decide } POS(A); \\
\text{(N1)} & \text{If } P(A|[x]) \leq \beta, & \text{decide } NEG(A); \\
\text{(B1)} & \text{If } \beta < P(A|[x]) < \alpha, & \text{decide } BND(A).
\end{array}$$

When $\alpha = \beta = \gamma$, we can simplify the rules (P-B) into (P2-B2) [18]:

$$\begin{array}{lll}
\text{(P2)} & \text{If } P(A|[x]) > \alpha, & \text{decide } POS(A); \\
\text{(N2)} & \text{If } P(A|[x]) < \alpha, & \text{decide } NEG(A); \\
\text{(B2)} & \text{If } P(A|[x]) = \alpha, & \text{decide } BND(A).
\end{array}$$

These minimum-risk decision rules offer us a basic foundation in which to build a rough set risk analysis component for a WSS. They give us the ability to not only collect decision rules from data, but also the calculated risk that is involved when discovering (or acting upon) those rules.

3 Web-Based Support Systems with a Rough Set Component

For our future purposes of using rough sets for a WSS, we will look at a particular probabilistic approach that allows us to calculate associated risk for a partitioning of the object universe. The decision-theoretic rough set model [19] allows us to enhance the traditional data mining component of a WSS by adding a risk element to the decision process. Using this risk element, users of a WSS can make more informed decisions based on the rule-based knowledge base. Based on the three regions (POS, BND, and NEG), there are two types of decisions or support that the rough set component can offer the user:

1. **Immediate Decisions** (Unambiguous) - These types of decisions are based upon classification within the POS and NEG regions. The user can interpret the findings as:
   (a) Classification in the POS region is a definitive "yes" answer, for instance, the symptoms or test results indicate a patient suffers breast cancer.
   (b) Classification in the NEG region is a definitive "no" answer, for instance, the symptoms indicate that a patient does $n$ot suffer breast cancer.
2. **Delayed Decisions** (Ambiguous) - These types of decisions is based upon classification within the BND region. Since there is some element of uncertainty in this region, the user of the WSS should proceed with a "wait-and-see" agenda. Rough set theory may be meaningless when the "wait-and-see" cases are too large and unambiguous rules are scarce. Two approaches may be applied to decrease ambiguity:
   (a) Obtain more information [9]. More lab tests will be conducted in order to diagnose whether a patient suffers a disease, i.e., introduce more attributes of the information table. Conduct further studies to gain knowledge in order to make an immediate decision from the limited data sets.
   (b) A decreased tolerance for acceptable loss [18,19,20]. The probabilistic aspects of the rough set component allows the user to modify the (acceptable) loss functions in order to increase certainty. However, this may also increase the risk of "false-positives" and "false-negatives". For instance, a doctor may diagnose a patient with a lung infection with a

simple cough symptom and prescribe an antibiotic for treatment. The risk to both patient and doctor of the wrong diagnosis is relatively low compared to a conclusion of lung cancer and treated with chemotherapy. The decision-theoretic rough set model is adapted to consider the risk factor and calculate the tolerance level for a WSS.

These types of decisions could greatly influence the effectiveness of the knowledge base derived from the rough set component. The risk element provided by the decision-theoretic rough set model provides the user with the ability to customize the knowledge base to suit their priorities.

3.1 Binding Rough Sets with Web-Based Support Systems

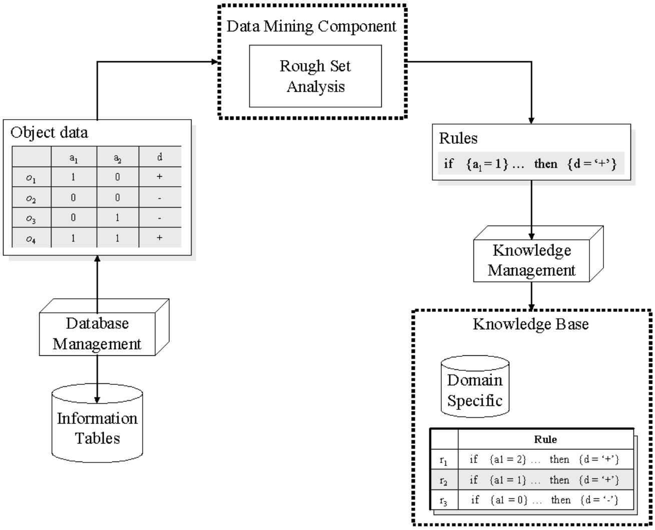
Both algebraic and probabilistic rough sets provide the user with methods to derive rules. These rules can then be used to support decision making. There are many types of WSS that support some form of decision making, including but not limited to Web-based decision support systems. Therefore, it follows that an important extension of rough sets should be the migration to the Web.

Looking at rough sets from a data mining perspective, it is one of many knowledge discovery methods that are available to the users. Given a depository of data, rough sets can be used to perform analysis of this data. The end result being a set of decision rules that can be used to describe, extend, or predict the domain in which the data was derived [10]. For example, a time-series data set describing stock index prices can be analyzed with rough sets in order to obtain decision rules that aid in forecasting the market [2].

The WSS framework utilizing rough sets would be connected to the components *knowledge base*, *database*, *interface*, as well as the other components [16]. This is shown in Fig. 1. Taking on the duties of the *data mining* component, rough sets would perform analysis on the data within the *database* component. It would derive decision rules based on this data. These rules would be captured by the *knowledge management* component, which would index it into the *domain specific knowledge base*.

Some derivations of WSS could make use of the cost or loss annotations provided by the decision-theoretic rough set models. This may include Web-based decision support systems where a decision made in conjunction with a decision rule could have some perceived implications portrayed by the loss functions. These λ_{P2} and λ_{N1} errors, or "false positive" and "false negative" errors respectively can be provided to the decision maker so that he or she can be better informed on which decision to make. Fig. 1 can be modified so that the *domain specific knowledge base* contains information regarding the $\lambda_{\diamond 1}$ and $\lambda_{\diamond 2}$ values corresponding to the decision rules used by the user.

The use of the decision-theoretic rough set model in WSS distances itself from the uses of the traditional rough sets. Rough set analysis is transformed into decision-theoretic rough analysis. Rules that are normally formed through rough set analysis are transformed into a "risk analysis" pair (decision rules with their respective costs). The decision making performed with the traditional



	a_1	a_2	d
o_1	1	0	+
o_2	0	0	-
o_3	0	1	-
o_4	1	1	+

	Rule
r_1	if {a1 = 2} … then {d = '+'}
r_2	if {a1 = 1} … then {d = '+'}
r_3	if {a1 = 0} … then {d = '-'}

Fig. 1. Sub-architecture with Rough Set Analysis as a data mining component

rule set can now be thought of a decision making with minimum cost tasks. For example, a set of rules governing the diagnosis of cancer would also have a set of risks that indicate the potential loss for a false positive or false negative.

3.2 Web-Based Medical Support Systems

In this section, we will describe a Web-based medical support system. A WMSS contains many components whose duties range from scheduling of appointments to maintaining a knowledge base of symptoms and diseases [12]. We focus on the decision support aspect [6] of a WMSS. This system will use rough sets to perform analysis on compatible medical data. A WMSS has a primary goal of supporting decisions of an expert (doctor, primary or secondary diagnostician).

For our purposes of using rough sets for a WMSS, we will look at a probabilistic approach that allows us to calculate associated risk for a partitioning of the object universe. The decision-theoretic rough set model allows us to enhance the traditional data mining component of a WSS by adding a risk element to the decision process. The architecture is shown in Fig 2. The individual components are described as follows:

Patient Database. The patient database contains data pertaining to patient symptoms. This is gathered by the users of the system by a number of questions and trials performed on the patients. The rough set component and information retrieval component access this database regularly.

Database Management System. The DBMS is a major component in any modern system. This is middleware that provides access to the patient database. The rough set component communicates with the DBMS for tuple data.

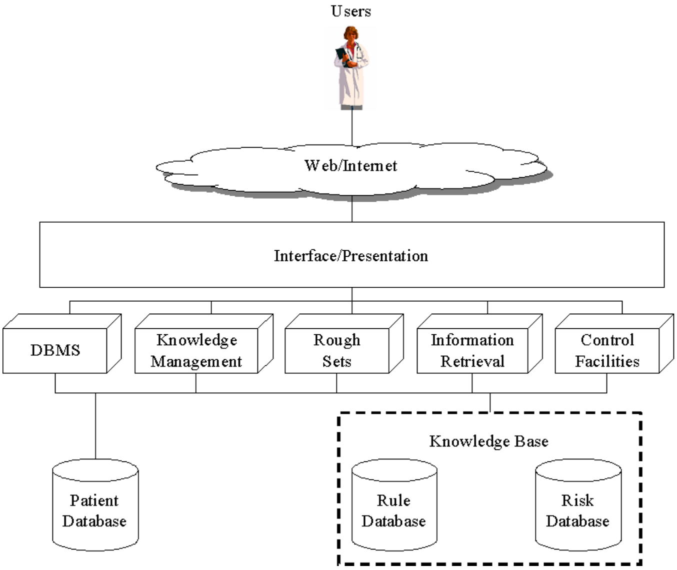


Fig. 2. A Partial Architecture of a Web-based Medical Support System

Knowledge Management. The knowledge management middleware component manages the knowledge base and provides access to the rule database and associated risk database. It acquires the risk analysis pairs from the rough set analysis component and indexes them accordingly.

Rough Set Component. The rough set component in this particular system makes use of the decision-theoretic rough set model to acquire knowledge (rules) and the associated risks of using that knowledge. It provides the users of the system with timely information to support their decision making.

Information Retrieval. The information retrieval component provides search and indexing functionality. Rough sets can play a role here [21]. This component is directly interfaced and has primary communication with the interface / presentation layer. Users of the system will access this component to retrieve patient data, information from knowledge base and other tasks.

Other Control Facilities. Other control facilities include a robust security and permission component. Since patient data is very sensitive and with the Web functionality of the entire system, security is a major concern.

Knowledge Base. The knowledge base component contains two major subcomponents: the rule database and associated risk database. The rule database is an index of the knowledge derived from the rough set analysis component. The associated risk database contains risk values for accepting a decision implied by the rule database.

Interface/Presentation. This component is an entire layer of user interfaces and server-side form request handlers. This layer presents the users of the system with a clean and efficient web interface for entering patient data, searching, and obtaining decision support.

Users. The users of the system include general practitioners, primary doctors, secondary diagnosticians, etc. The users access the WMSS via Interface / Presentation layer through the Internet.

The users may take the information provided and make an *unambiguous* decision. This represents a definitive "yes" or "no" diagnosis for a particular set of symptoms. This corresponds to those patients classified in either the POS or NEG regions. For those cases in the BND region, a "wait-and-see" decision is used. The support system would suggest that the users either decrease their tolerance (loss functions) or acquire additional data on the subject.

3.3 Web-Based Medical Support Systems with Risk Analysis

To see how a decision-theoretic rough set analysis component effects decisions in a WMSS, let us consider two diagnosis scenarios. The risk or cost is defined as consequences of the wrong diagnosis. Based on our common sense, the cost of wrong diagnosis of a flu is lower than that of a wrong diagnosis of cancer. The cancer diagnosis tolerance levels of either a false-negative or false-positive are very low. Patients may sacrifice their lives when a false-negative level is high as they may miss the best treatment time. They may suffer consequences of chemotherapy for non-existent cancer when a false-positive is high.

Using Table 1, let us form two hypothetical scenarios of patient diagnoses. First, a diagnosis of low severity with a low cost for a wrong diagnosis. This could be testing for a patient's minor allergies. An allergy test would be looking for positive indicators for symptoms $S = \{S_1, S_2, S_3\}$ and the diagnosis decision $D = \{\text{Decision}\}$. Below is a typical sample of loss functions for this situation:

$$\lambda_{P2} = \lambda_{N1} = 1u, \qquad \lambda_{P1} = \lambda_{N2} = \lambda_{B1} = \lambda_{B2} = 0, \tag{4}$$

where u is a unit cost determined by the individual administration. In this scenario, the administration has deemed that a false-positive (λ_{P2}) and false-negative (λ_{N1}) diagnosis has some form of cost whereas indeterminant diagnoses (λ_{B1}, λ_{B2}) and correct diagnoses (λ_{P1}, λ_{N2}) have no cost.

A diagnosis of high severity could have a high cost for a wrong diagnosis. This could be testing for whether a patient has a form of cancer. In Table 1, the cancer test would be looking for positive indicators for symptoms $S = \{S_1, S_4, S_5\}$ and the diagnosis decision $D = \{\text{Decision}\}$. Patient o_3 having symptoms $s_{3,1}$, $s_{3,2}$, and $s_{3,3}$ would give a decision $d_{3,1}$ for allergy tests. Below is a typical sample of loss functions for this situation:

$$\lambda_{P2} = \lambda_{N1} = 2u, \qquad \lambda_{B1} = \lambda_{B2} = 1u, \qquad \lambda_{P1} = \lambda_{N2} = 0, \tag{5}$$

Table 1. An Information Table

Patient	S_1	S_2	S_3	S_4	S_5	Decision
o_1	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$	$s_{1,4}$	$s_{1,5}$	d_1
o_2	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$	$s_{2,4}$	$s_{2,5}$	d_2
o_3	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$	$s_{3,4}$	$s_{3,5}$	d_3
o_4	$s_{4,1}$	$s_{4,2}$	$s_{4,3}$	$s_{4,4}$	$s_{4,5}$	d_4
o_5	$s_{5,1}$	$s_{5,2}$	$s_{5,3}$	$s_{5,4}$	$s_{5,5}$	d_5
o_6	$s_{6,1}$	$s_{6,2}$	$s_{6,3}$	$s_{6,4}$	$s_{6,5}$	d_6

where u is a unit cost determined by the individual administration. In this scenario, the administration has deemed that a false-positive (λ_{P2}) and false-negative (λ_{N1}) diagnoses is twice as costly as indeterminant diagnoses (λ_{B1} and λ_{B2}). Correct diagnoses (λ_{P1} and λ_{N2}) have no cost.

Using the loss functions in (4) and calculating the parameters using the formulas in (3), we obtain $\alpha = 1$, $\gamma = 0.5$, and $\beta = 0$. When $\alpha > \beta$, we get $\alpha > \gamma > \beta$. We use the the simplified decision rules (P1-B1) to obtain our lower and upper approximations:

$$\begin{aligned} \underline{apr}_{(1,0)}(A) &= \{x \in U | P(A|[x]) = 1\}, \\ \overline{apr}_{(1,0)}(A) &= \{x \in U | P(A|[x]) > 0\}. \end{aligned} \tag{6}$$

Using the loss functions in (5) and calculating the parameters using the formulas in (3), we obtain $\alpha = \beta = \gamma = 0.5$. When $\alpha = \beta = \gamma$, we use the simplified decision rules (P2-B2) to can obtain our new lower and upper approximations:

$$\begin{aligned} \underline{apr}_{(0.5,0.5)}(A) &= \{x \in U | P(A|[x]) > 0.5\}, \\ \overline{apr}_{(0.5,0.5)}(A) &= \{x \in U | P(A|[x]) \geq 0.5\}. \end{aligned} \tag{7}$$

The approximations in (6) mean that we can definitely class patient x into diagnosis class A if all similar patients are in diagnosis class A. The low loss functions (4) have indicated that users of the system can have high certainty when dealing with this class of patient. The approximations in (7) mean that we can definitely class patient x into diagnosis class A if strictly more than half of similar patients are in diagnosis class A. These examples use the loss functions to determine how high the level of certainty regarding a patient's symptoms needs to be in order to minimize cost.

4 Conclusion

We further explain the importance of Web-based support systems. A decision-theoretic rough set model can be used as the data mining component for a WSS. This extended model allows the component to provide additional decision support to the users. The two types of decision the users can make, immediate and delayed, are now fully supported by the rough set component. We reiterate this by detailing a Web-based medical support system framework that incorporates risk analysis through loss functions. The rough set component builds and maintains a risk database to assist the users in assessing the knowledge provided by the rule database.

References

1. Grzymala-Busse, J.W., Hippe, Z.S.: Data mining methods supporting diagnosis of melanoma. In: 18th IEEE Symposium on Computer-Based Medical Systems CBMS'05, pp. 371–373 (2005)
2. Herbert, J., Yao, J.T.: Time-series data analysis with rough sets. In: CIEF'04, pp. 908–911 (2005)
3. Hirano, S., Tsumoto, S.: Rough representation of a region of interest in medical images. International Journal of Approximate Reasoning 40, 23–34 (2005)
4. Ishikawa, T., Klaisubun, P., Honma, M., Qian, Z.: Remarkables: A web-based research collaboration support system using social bookmarking tools. In: WSS'06, Proceedings of WI-IAT 2006, Workshops, pp. 192–195 (2006)
5. Khosla, R., Lai, C.: Mediating human decision making with emotional attitudes in web based decision support systems. In: WSS'06, Proceedings of WI-IAT 2006, Workshops, pp. 204–207 (2006)
6. Krause, P., Fox, J., O'Neil, M., Glowinski, A.: Can we formally specify a medical decision-support system? IEEE Expert-Intelligent Systems & Their Applications 8, 56–61 (1993)
7. Kusiak, A., Kern, J.A., Kernstine, K.H., Tseng, B.T.L.: Autonomous decision-making: A data mining approach. IEEE Transactions on Information Technology In Biomedicine 4, 274–284 (2000)
8. Mitra, P., Mitra, S., Pal, S.K.: Staging of cervical cancer with soft computing. IEEE Transactions on Biomedical Engineering 47, 934–940 (2000)
9. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
10. Peters, J.F., Skowron, A.: A rough set approach to knowledge discovery. International Journal of Intelligent Systems 17, 109–112 (2002)
11. Power, D.J., Kaparthi, S.: Building web-based decision support systems. Studies in Informatics and Control 11, 291–302 (2002)
12. Stalidis, G., Prentza, A., Vlachos, I.N., Maglavera, S., Koutsouris, D.: Medical support system for continuation of care based on xml web technology. International Journal of Medical Informatics 64, 385–400 (2001)
13. Yao, J.T.: Supporting research with weblogs: A study on web-based research support systems. In: WSS'06, Proceedings of WI-IAT 2006 Workshops, pp. 161–164 (2006)
14. Yao, J.T., Lingras, P. (eds.): Proceedings of 2003 WI/IAT Workshop on Applications, Products and Services of Web-based Support Systems (WSS'03), Halifax, Canada (2003)
15. Yao, J.T., Raghvan, V.V., Wang, G.Y. (eds.): Proceedings of the Second International Workshop on Web-based Support System (WSS'04), Beijing, China (2004)
16. Yao, J.T., Yao, Y.Y.: Web-based support systems. In: [14], pp. 1–5 (2003)
17. Yao, Y.Y.: Information retrieval support systems. In: Proceedings of FUZZ-IEEE'02, pp. 773–778 (2002)

18. Yao, Y.Y.: Decision-theoretic rough set models. In: Proceedings of RSKT'07, LNAI. vol. 4481, pp. 1–12 (2007)
19. Yao, Y.Y., Wong, S.K.M.: A decision theoretic framework for approximating concepts. International Journal of Man.-machine Studies 37, 793–809 (1992)
20. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences 46, 39–59 (1993)
21. Ziarko, W., Fei, X.: VPRSM approach to web searching. In: RSCTC'02, LNAI, vol. 2475, pp. 514–521 (2002)

Interpreting Low and High Order Rules: A Granular Computing Approach

Yiyu Yao, Bing Zhou, and Yaohua Chen

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
{yyao,zhou200b,chen115y}@cs.uregina.ca

Abstract. The main objective of this paper is to provide a granular computing based interpretation of rules representing two levels of knowledge. This is done by adopting and adapting the decision logic language for granular computing. The language provides a formal method for describing and interpreting conditions in rules as granules and rules as relationships between granules. An information table is used to construct a concrete granular computing model. Two types of granules are constructed from an information table. They lead to two types of rules called low order and high order rules. As examples, we examine rules in the standard rough set analysis and dominance-based rough set analysis.

Keywords: Low order rules, high order rules, granular computing, dominance-based rough set analysis.

1 Introduction

Rules are a commonly used form for representing knowledge. Two levels of knowledge may be expressed in terms of low and high order rules, respectively [16]. A low order rule expresses connections between attribute values of the same object. Classification rules are a typical example of low order rules. For example, a classification rule may state that "if the Hair color is blond and Eye color is blue, then the Class is +." A high order rule expresses connections of different objects in terms of their attribute values. Functional dependencies are a typical example of high order rules. For example, a functional dependency rule may state that "if two persons have the same Hair color, then they will have the same Eye color." The notion of a high order rule is also related to relational learning, in which a k-ary predicate is used to define a relation between k objects [10]. For simplicity, in this paper we only consider binary relations.

In rough set analysis [6,7,8], a decision logic language *DL* is used to build conditions in low order rules and to interpret these conditions as subsets (i.e., granules) of objects. This language is referred to as $\mathcal{L}_0$ [16]. In order to describe high order rules, an extended language $\mathcal{L}_1$ is introduced [16]. In $\mathcal{L}_1$, conditions are interpreted in terms of a set of object pairs. The two languages share the same syntactic rules, with two semantic interpretations. That is, their main differences lie in the different interpretations of atomic formulas. It is therefore possible to introduce a common language.

In this paper, we propose a decision logic language $\mathcal{L}$ for granular computing. Instead of expressing the atomic formulas by a particular concrete type of conditions, we treat

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 371–380, 2007.

them as primitive notions that can be interpreted differently. This flexibility enables us to describe different types of rules. The language is interpreted in the Tarski's style through the notion of a model and satisfiability. The model is a non-empty domain consisting of a set of individuals. An individual satisfies a formula if the individual has the properties as specified by the formula. A concept is therefore jointly defined by a pair consisting of the intension of the concept, a formula of the language, and the extension of the concept, a subset of the model.

As illustrative examples to show the usefulness of the proposed language, we analyze rules in the standard rough set analysis [6,7,8] and dominance-based rough set analysis [2,3,4,11,15].

2 A Decision Logic Language for Granular Computing

By extracting the high-level similarities of the decision logic languages *DL*, $\mathcal{L}_0$, and $\mathcal{L}_1$, we propose a logic language $\mathcal{L}$ for granular computing.

The language $\mathcal{L}$ is constructed from a finite set of atomic formulas, denoted by $\mathcal{A} = \{p, q, ...\}$. Each atomic formula may be interpreted as representing one piece of basic knowledge. The physical meaning of atomic formulas becomes clearer in a particular application. In general, an atomic formula corresponds to one particular property of an individual under discussion. The construction of atomic formulas is an essential step of knowledge representation. The set of atomic formulas provides a basis on which more complex knowledge can be represented. Compound formulas can be built recursively from atomic formulas by using logic connectives. If ϕ and ψ are formulas, then so are $(\neg\phi)$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$, and $(\phi \leftrightarrow \psi)$.

The semantics of the language $\mathcal{L}$ can be defined in the Tarski's style through the notion of a model and satisfiability. The model is a nonempty domain consisting of a set of individuals, denoted by $M = \{x, y, ...\}$. The meaning of formulas can be given recursively. For an atomic formula p, we assume that an individual $x \in M$ either satisfies p or does not satisfy p, but not both. For an individual $x \in M$, if it satisfies an atomic formula p, we write $x \models p$, otherwise, we write $x \not\models p$. The satisfiability of an atomic formula by individuals of M is viewed to be the basic knowledge describable by the language $\mathcal{L}$. An individual satisfies a formula if the individual has the properties as specified by the formula. Let ϕ and ψ be two formulas, the satisfiability of compound formulas is defined as follows:

$$
\begin{array}{lll}
(1). & x \models \neg\phi & \text{iff} \;\; x \not\models \phi, \\
(2). & x \models \phi \wedge \psi & \text{iff} \;\; x \models \phi \text{ and } x \models \psi, \\
(3). & x \models \phi \vee \psi & \text{iff} \;\; x \models \phi \text{ or } x \models \psi, \\
(4). & x \models \phi \rightarrow \psi & \text{iff} \;\; x \models \neg\phi \vee \psi, \\
(5). & x \models \phi \leftrightarrow \psi & \text{iff} \;\; x \models \phi \rightarrow \psi \text{ and } x \models \psi \rightarrow \phi.
\end{array}
$$

In order to emphases the roles played by the set of atomic formulas $\mathcal{A}$ and the set of individuals M, we also rewrite the language $\mathcal{L}$ as $\mathcal{L}(\mathcal{A}, M)$.

The construction of the set of atomic formulas and the model M depends on a particular application. For modeling different problems, we may choose different sets of atomic formulas and models. The language $\mathcal{L}$ therefore is flexible and enables us to describe a variety of problems.

With the notion of satisfiability, one can introduce a set-theoretic interpretation of formulas of the language $\mathcal{L}$. If ϕ is a formula, the meaning of ϕ in the model M is the set of individuals defined by [7]:

$$m(\phi) = \{x \in M \mid x \models \phi\}. \tag{1}$$

That is, $m(\phi)$ is the set of individuals satisfying a formula ϕ. This enables us to establish a correspondence between logic connectives and set-theoretic operators. Specifically, the following properties hold:

$$\begin{array}{ll} \text{(C1).} & m(\neg\phi) = -m(\phi), \\ \text{(C2).} & m(\phi \wedge \psi) = m(\phi) \cap m(\psi), \\ \text{(C3).} & m(\phi \vee \psi) = m(\phi) \cup m(\psi), \\ \text{(C4).} & m(\phi \rightarrow \psi) = -m(\phi) \cup m(\psi), \\ \text{(C5).} & m(\phi \leftrightarrow \psi) = (m(\phi) \cap m(\psi)) \cup (-m(\phi) \cap -m(\psi)), \end{array}$$

where $-m(\phi) = M - m(\phi)$ denotes the set complement of $m(\phi)$.

In the study of concepts, many interpretations have been proposed and examined. The classical view regards a concept as a unit of thought consisting of two parts, i.e., the intension and extension of the concept [1,13]. By using the language $\mathcal{L}$, we obtain a simple and precise representation of a concept in terms of its intension and extension. That is, a concept is defined by a pair $(m(\phi), \phi)$. The formula ϕ is the description of properties shared by individuals in $m(\phi)$, and $m(\phi)$ is the set of individuals satisfying ϕ. A concept is thus described jointly by its intension and extension. This formulation enables us to study concepts in a logic setting in terms of intensions and in a set-theoretic setting in terms of extensions.

The language $\mathcal{L}$ provides a formal method for describing granules. Elements of a granule may be interpreted as instances of a concept, i.e., the extension of the concept. The formula is a formal description of the granule. In this way, the language $\mathcal{L}$ only enables us to define certain subsets of M. For an arbitrary subset of M, we may not be able to construct a formula for it. In other words, depending on the set of atomic formulas, the language $\mathcal{L}$ can only describe a restricted family of subsets of M.

3 Interpretation of Low and High Order Rules Using the Language $\mathcal{L}$

We interpret different types of rules of an information table as concrete applications to show the usefulness of the language $\mathcal{L}$.

3.1 Information Table

An information table provides a convenient way to describe a finite set of objects by a finite set of attributes [7]. Formally, an information table can be expressed as:

$$S = (U, At, \{V_a \mid a \in At\}, \{\{R_a\} \mid a \in At\}, \{I_a \mid a \in At\}),$$

where

$$\begin{aligned}
&U \text{ is a finite nonempty set of objects called universe,}\\
&At \text{ is a finite nonempty set of attributes,}\\
&V_a \text{ is a nonempty set of values for } a \in At,\\
&\{R_a\} \text{ is a family of binary relations on } V_a,\\
&I_a : U \to V_a \text{ is an information function.}
\end{aligned}$$

Each information function I_a maps an object in U to a value of V_a for an attribute $a \in At$.

Our definition of an information table considers more knowledge and information about relationships between values of attributes. Each relation R_a can represent similarity, dissimilarity, or ordering of values in V_a [1]. The equality relation $=$ is only a special case of R_a. The rough set theory and the *DL* language use the trivial equality relation on attribute values [7].

Pawlak and Skowron [8] consider a more generalized notion of an information table. For each attribute $a \in At$, a relational structure $\Re_a$ over V_a is introduced. Furthermore, a language can be defined based on the relational structures. A binary relation is a special case of relational structures. Thus, the discussion of this paper may be viewed as a special case of Pawlak and Skowron's formulation.

3.2 Low Order Rules

For interpreting low order rules, we construct a language by using U as the model M. That is, individuals of M are objects in the universe U. The set of atomic formulas are constructed as follows. With respect to an attribute $a \in At$ and an attribute value $v \in V_a$, an atomic formula of the language $\mathcal{L}$ is denoted by (a, R_a, v). An object $x \in U$ satisfies an atomic formula (a, R_a, v) if the value of x on attribute a is R_a-related to the value v, that is $I_a(x)\ R_a\ v$, we write:

$$x \models (a, R_a, v) \text{ iff } \quad I_a(x)\ R_a\ v.$$

We denote the language for interpreting low order rules as $\mathcal{L}(\{(a, R_a, v)\}, U)$. The granule corresponding to the atomic formula (a, R_a, v), namely, its meaning set, is defined as:

$$m(a, R_a, v) = \{x \in U \mid I_a(x) R_a v\}.$$

Granules corresponding to compound formulas are defined by Equation (1).

A low order rule can be derived according to the relationships between these granules. We can express rules in the form $\phi \Rightarrow \psi$ by using formulas of the language $\mathcal{L}$. For easy understanding, we reexpress the formula (a, R_a, v) in another form based on the definition of satisfiability of the atomic formulas. An example of a low order rule is given as:

$$\text{Low Order rule:} \qquad \bigwedge_{i=1}^{n} (I_{a_i}(x)\ R_{a_i}\ v_{a_i}) \Rightarrow \bigwedge_{j=1}^{m} (I_{d_j}(x)\ R_{d_j}\ v_{d_j}),$$

where $x \in U$, $v_{a_i} \in V_{a_i}$, $v_{d_j} \in V_{d_j}$, $a_i \in At$, and $d_j \in At$. For simplicity, we only use conjunction $\wedge$ in the rule.

3.3 High Order Rules

For interpreting high order rules, we construct a language by using $U \times U$ as the model M. That is, individuals of M are object pairs in $U \times U$. The set of atomic formulas are constructed as follows. With respect to an attribute $a \in At$, an atomic formula of the language $\mathcal{L}$ is denoted by (a, R_a). A pair of objects $(x, y) \in U \times U$ satisfies an atomic formula (a, R_a) if the value of x is R_a-related to the value of y on the attribute a, that is, $I_a(x)\ R_a\ I_a(y)$. We write:

$$(x, y) \models (a, R_a) \ \text{ iff } \ I_a(x)\ R_a\ I_a(y).$$

For clarity, we denote the language as $\mathcal{L}(\{(a, R_a)\}, U \times U)$. The granule corresponding to the atomic formula (a, R_a), i.e., the meaning set, is defined as:

$$m(a, R_a) = \{(x, y) \in U \times U \mid I_a(x) R_a I_a(y)\}.$$

Granules corresponding to the compound formulas are defined by Equation (1).

A high order rule expresses the relationships between these granules. An example of a high order rule is given as:

$$\text{High Order rule:} \qquad \bigwedge_{i=1}^{n} (I_{a_i}(x)\ R_{a_i}\ I_{a_i}(y)) \Rightarrow \bigwedge_{j=1}^{m} (I_{d_j}(x)\ R_{d_j}\ I_{d_j}(y)),$$

where $(x, y) \in U \times U$, $a_i \in At$, $d_j \in At$.

3.4 Quantitative Measures of Rules

The meanings and interpretations of a rule $\phi \Rightarrow \psi$ can be further clarified by using the extensions $m(\phi)$ and $m(\psi)$ of the two concepts. More specifically, we can define quantitative measures indicating the strength of a rule. A systematic analysis of probabilistic quantitative measures can be found in [14]. Two examples of probabilistic quantitative measures are [12]:

$$accuracy(\phi \Rightarrow \psi) = \frac{\mid m(\phi \wedge \psi) \mid}{\mid m(\phi) \mid}, \qquad coverage(\phi \Rightarrow \psi) = \frac{\mid m(\phi \wedge \psi) \mid}{\mid m(\psi) \mid}, \tag{2}$$

where $\mid \cdot \mid$ denotes the cardinality of a set. The two measures are applicable to both low and high order rules. This demonstrates the flexibility and power of the language $\mathcal{L}$.

While the accuracy reflects the correctness of the rule, the coverage reflects the applicability of the rule. If $accuracy(\phi \Rightarrow \psi) = 1$, we have a strong association between ϕ and ψ. A smaller value of accuracy indicates a weak association. A higher coverage suggests that the relationships of more individuals can be derived from the rule. The accuracy and coverage are not independent of each other, one may observe a trade-off between accuracy and coverage. A rule with higher coverage may have a lower accuracy, while a rule with higher accuracy may have a lower coverage.

4 Rough Set Approaches on Rules

In this section, we use two rough set approaches [2,3,4,6,7,8,11,15] as examples to illustrate the usefulness of the language $\mathcal{L}$. The basic results are summarized in Table 1. For comparison, we also include the results of generalized rough set analysis based on an arbitrary binary relation R_a on attribute values.

Table 1. Rough Set Approaches for Studying Low and High Order Rules

Relation	Low Order Rule	High Order Rule	Method
R	$I_a(x) R_a v_a \Rightarrow I_d(x) R_d v_d$	$I_a(x) R_a I_a(y) \Rightarrow I_d(x) R_d I_d(y)$	Generalized Rough Set Analysis
$=$	$I_a(x) = v_a \Rightarrow I_d(x) = v_d$	$I_a(x) = I_a(y) \Rightarrow I_d(x) = I_d(y)$	Standard Rough Set Analysis
$\succeq$	$I_a(x) \succeq_a v_a \Rightarrow I_d(x) \succeq_d v_d$	$I_a(x) \succeq_a I_a(y) \Rightarrow I_d(x) \succeq_d I_d(y)$	Dominance-based Rough Set Analysis

4.1 Standard Rough Set Analysis

The standard rough set analysis is based on the trivial equality relation on attribute values [6,7,8]. It is used for the extraction of rules for classification and attribute dependency analysis. By using the language $\mathcal{L}$, the standard rough set approach can be formulated as follows.

For low order rules, the language is given by $\mathcal{L}(\{(a, =, v)\}, U)$ with atomic formulas of the form of $(a, =, v)$. An object $x \in U$ satisfies an atomic formula $(a, =, v)$ if the value of x on attribute a is v, that is, $I_a(x) = v$. We write:

$$x \models (a, =, v) \text{ iff } \ I_a(x) = v.$$

The granule corresponding to the atomic formula $(a, =, v)$ is:

$$m(a, =, v) = \{x \in U \mid I_a(x) = v\}.$$

The granule $m(a, =, v)$ is also referred to as the block defined by the attribute-value pair (a, v) [5]. Blocks correspond to the atomic formulas and are used to construct rules. Low order rules can be expressed based on the equality relation $=$. An example of a low order rule is:

$$\bigwedge_{i=1}^{n} (I_{a_i}(x) = v_{a_i}) \Rightarrow \bigwedge_{j=1}^{m} (I_{d_j}(x) = v_{d_j}),$$

where $x \in U$, $v_{a_i} \in V_{a_i}$, $v_{d_j} \in V_{d_j}$, $a_i \in At$, and $d_j \in At$.

For high order rules, the language is given by $\mathcal{L}(\{(a, =)\}, U \times U)$ with atomic formulas of the form of $(a, =)$. A pair of objects $(x, y) \in U \times U$ satisfies an atomic formula $(a, =)$ if the value of x equals to the value of y on attribute a, that is, $I_a(x) = I_a(y)$. We write:

$$(x, y) \models (a, =) \text{ iff } \ I_a(x) = I_a(y).$$

Table 2. An Information Table

Object	Height	Hair	Eyes	Class
o_1	short	blond	blue	+
o_2	short	blond	brown	-
o_3	tall	red	blue	+
o_4	tall	dark	blue	-
o_5	tall	dark	blue	-
o_6	tall	blond	blue	+
o_7	tall	dark	brown	-
o_8	short	blond	brown	-

The granule corresponding to the atomic formula $(a, =)$ is:

$$m(a,=) \;=\; \{(x,y) \in U \times U \mid I_a(x) = I_a(y)\}.$$

High order rules can be expressed by using the equality relation $=$. An example of a high order rule is:

$$\bigwedge_{i=1}^{n}(I_{a_i}(x) \;=\; I_{a_i}(y)) \Rightarrow \bigwedge_{j=1}^{m}(I_{d_j}(x) \;=\; I_{d_j}(y)),$$

where $(x, y) \in U \times U$, $a_i \in At$, $d_j \in At$.

Example 1. Table 2 is an information table taken from [9]. Each object is described by four attributes. The column labeled by "Class" denotes an expert's classification of the objects.

An example of a low order rule in this information table is:

$$\mathrm{LR}_1 : \;(I_{\mathrm{Hair}}(x) = \mathrm{blond}) \wedge (I_{\mathrm{Eyes}}(x) = \mathrm{blue}) \Rightarrow (I_{\mathrm{Class}}(x) = +).$$

That is, if an object has blond hair and blue eyes, then it belongs to class +. An example of a high order rule is:

$$\mathrm{HR}_1 : \;(I_{\mathrm{Height}}(x) = I_{\mathrm{Height}}(y)) \wedge (I_{\mathrm{Eyes}}(x) = I_{\mathrm{Eyes}}(y)) \Rightarrow (I_{\mathrm{Class}}(x) = I_{\mathrm{Class}}(y)).$$

That is, if two objects have the same height and the same eye color, then they belong to the same class. By using the probabilistic quantitative measures, we have:

$$accuracy(\mathrm{LR}_1) = 1, \qquad coverage(\mathrm{LR}_1) = 2/3.$$

The association between $(\mathrm{Hair}, =, \mathrm{blond}) \wedge (\mathrm{Eyes}, =, \mathrm{blue})$ and $(\mathrm{Class}, =, +)$ reaches the maximum value 1, and the applicability of the rule is also high. For rule HR_1, we have:

$$accuracy(\mathrm{HR}_1) = 7/11, \qquad coverage(\mathrm{HR}_1) = 7/17.$$

In this case, $(\mathrm{Height}, =) \wedge (\mathrm{Eyes}, =)$ does not tell us too much information about the overall objects classification in terms of both accuracy and coverage.

4.2 Dominance-Based Rough Set Analysis

The dominance-based rough set analysis proposed by Greco, Matarazzo and Slowinski [2,3,4] is based on preference relations on attribute values. It is used for the extraction of rules for ranking and attribute dependency analysis. Several different types of rules are introduced in dominance-based rough set analysis. In what follows, we interpret two types of such rules by using the language $\mathcal{L}$, as demonstrated in [11,15,16].

For low order rules, the language is given by $\mathcal{L}(\{(a, \succeq_a, v)\}, U)$. The granule corresponding to the atomic formula $(a, \succeq_a, v)$ is defined as:

$$m(a, \succeq_a, v) \ = \ \{x \in U \mid I_a(x) \succeq_a v\}.$$

Low order rules can be expressed by using preference relations. An example of a low order rule is:

$$\bigwedge_{i=1}^{n} (I_{a_i}(x) \ \succeq_{a_i} \ v_{a_i}) \Rightarrow \bigwedge_{j=1}^{m} (I_{d_j}(x) \ \succeq_{d_j} \ v_{d_j}),$$

where $x \in U$, $v_{a_i} \in V_{a_i}$, $v_{d_j} \in V_{d_j}$, $a_i \in At$, and $d_j \in At$.

For high order rules, the language is given by $\mathcal{L}(\{(a, \succeq_a)\}, U \times U)$. The granule corresponding to the atomic formula $(a, \succeq_a)$ is defined as:

$$m(a, \succeq_a) \ = \ \{(x, y) \in U \times U \mid I_a(x) \succeq_a I_a(y)\}.$$

High order rules can also be expressed by using the preference relations. An example of a high order rule is:

$$\bigwedge_{i=1}^{n} (I_{a_i}(x) \ \succeq_{a_i} \ I_{a_i}(y)) \Rightarrow \bigwedge_{j=1}^{m} (I_{d_j}(x) \ \succeq_{d_j} \ I_{d_j}(y)).$$

where $(x, y) \in U \times U$, $a_i \in At$, $d_j \in At$.

Example 2. Table 3, taken from [11], is an information table with preference relations on attribute values. It is a group of five products by five manufactures, each product is described by four attributes. The final ranking labeled by Overall may be determined by their market share of the products. The preference relations induce the following orderings:

$$\begin{array}{ll}
\succ_{\text{Size}}: & \text{small} \succ_{\text{Size}} \text{middle} \succ_{\text{Size}} \text{large}, \\
\succ_{\text{Warranty}}: & 3 \text{ years} \succ_{\text{Warranty}} 2 \text{ years}, \\
\succ_{\text{Price}}: & \$200 \succ_{\text{Price}} \$250 \succ_{\text{Price}} \$300, \\
\succ_{\text{Weight}}: & \text{very light} \succ_{\text{Weight}} \text{light} \succ_{\text{Weight}} \text{heavy} \succ_{\text{Weight}} \text{very heavy}, \\
\succ_{\text{Overall}}: & 1 \succ_{\text{Overall}} 2 \succ_{\text{Overall}} 3.
\end{array}$$

An example of a low order rule in this information table is:

$$\text{LR}_2: \quad (I_{\text{Size}}(x) \succeq \text{middle}) \wedge (I_{\text{Warranty}}(x) \succeq 3 \text{ years}) \Rightarrow I_{\text{Overall}}(x) \succeq 2.$$

Table 3. An Information Table with Preference Relations

Objects	Size	Warranty	Price	Weight	Overall
p_1	middle	3 years	\$200	heavy	1
p_2	large	3 years	\$300	very heavy	3
p_3	small	3 years	\$300	light	3
p_4	small	3 years	\$250	very light	2
p_5	small	2 years	\$200	very light	3

That is, if a product's size is greater than or equal to middle and warranty is greater than or equal to 3 years, then its overall ranking will be greater than or equal to 2. An example of a high order rule is:

$$\mathrm{HR}_2: \quad (I_{\mathrm{Size}}(x) \succeq I_{\mathrm{Size}}(y)) \wedge (I_{\mathrm{Price}}(x) \succeq I_{\mathrm{Price}}(y)) \Rightarrow I_{\mathrm{Overall}}(x) \succeq I_{\mathrm{Overall}}(y).$$

That is, if one product's size is smaller than or the same as another product and the price is not higher, then this product's overall ranking will be greater than or equal to the other product. By using the quantitative measures, we have:

$$accuracy(\mathrm{LR}_2) = 2/3, \qquad coverage(\mathrm{LR}_2) = 1.$$

There exists a strong association between the two concepts, and applicability of the rule reaches the highest level. Similarly, for rule HR_2, we have:

$$accuracy(\mathrm{HR}_2) = 11/13, \qquad coverage(\mathrm{HR}_2) = 11/18.$$

The concept $(\mathrm{Size}, \succeq) \wedge (\mathrm{Price}, \succeq)$ reflects the overall objects ranking positively in terms of both accuracy and coverage.

5 Conclusion

A granular computing based interpretation is presented in this paper. By extracting the high-level similarity from existing decision logic languages [7,16], we introduce a more general language $\mathcal{L}$ for granular computing. Two basic features of the language $\mathcal{L}$ are the set of atomic formulas $\mathcal{A}$ and a model M consisting of individuals. For each formula, the collection of all individuals satisfying the formula form a granule, called the meaning of the formula. A rule is therefore expressed as connection between two formulas and interpreted based on the corresponding granules of the two formulas.

Depending on particular applications, we can construct concrete languages by using different types of atomic formulas and the associated models. This flexibility of the language $\mathcal{L}$ is demonstrated by considering two rough set approaches, namely, the standard rough set analysis and dominance-based rough set analysis. The main differences of the two approaches are their respective treatments of atomic formulas and models. An information table is used to construct a concrete granular computing model. For standard rough set analysis, two types of granules are constructed based on two families of atomic formulas. One consists of a set of objects that share the same attribute value.

The other consists of object pairs that cannot be distinguished based on the values of an attribute. Low and high order rules are defined to describe relationships between these two types of granules. For dominance-based rough set analysis, similar interpretations can be obtained by using two different families of atomic formulas.

The results of the paper suggest that one may study rule mining at a more abstract level. Algorithms and evaluation measures can indeed be designed uniformly for both low and high order rules.

References

1. Demri, S., Orlowska, E.: Logical Analysis of Indiscernibility. In: Orlowska, E. (ed.) Incomplete Information: Rough Set Analysis, pp. 347–380. Physica Verlag, Heidelberg (1997)
2. Greco, S., Matarazzo, B., Slowinski, R.: Rough Approximation of a Preference Relation by Dominance Relations. European Journal of Operational Research 117, 63–83 (1999)
3. Greco, S., Matarazzo, B., Slowinski, R.: Rough Approximation by Dominance Relations. International Journal of Intelligent Systems 17, 153–171 (2002)
4. Greco, S., Slowinski, R., Stefanowski, J.: Mining Association Rules in Preference-ordered Data. In: Proceedings of the 13th International Symposium on Foundations of Intelligent Systems (ISMIS'02), pp. 442–450 (2002)
5. Grzymala-Busse, J.W.: Incomplete Data and Generalization of Indiscernibility Relation, Definability, and Approximations. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. In: Proceedings of 10th International Conference, LNAI, vol. 3641, pp. 244–253, Springer, Heidelberg (2005)
6. Nguyen, H.S., Skowron, A., Stepaniuk, J.: Granular Computing: a Rough Set Approach. Computational Intelligence 17, 514–544 (2001)
7. Pawlak, Z.: Rough Sets - Theoretical Aspects of Reasoning about Data. Kluwer Publishers, Boston (1991)
8. Pawlak, Z., Skowron, A.: Rough Sets: Some Extensions. Information Science 177, 28–40 (2007)
9. Quinlan, J.R.: Learning Efficient Classification Procedures and Their Application to Chess End-games. In: Michalski, J.S., Carbonell, J.G., Mirchell, T.M. (eds.) Machine Learning: An Artificial Intelligence Approach, vol. 1, pp. 463–482. Morgan Kaufmann, San Francisco (1983)
10. Quinlan, J.R.: Learning Logical Definitions from Relations. Machine Learning 5, 239–266 (1990)
11. Sai, Y., Yao, Y.Y., Zhong, N.: Data Analysis and Mining in Ordered Information Tables. In: Proceedings of the 2001 IEEE International Conference on Data Mining. pp. 497–504 (2001)
12. Tsumoto, S.: Antomated Discovery of Plausible Rules Based on Rough Sets and Rough Inclusion. In: Zhong, N., Zhou, L. (eds.) Proceedings of PAKDD'99. LNCS (LNAI), vol. 1574, pp. 210–219. Springer, Heidelberg (1999)
13. Wille, R.: Concept Lattices and Conceptual Knowledge Systems. Computers Mathematics with Applications 23, 493–515 (1992)
14. Yao, Y.Y., Zhong, N.: An Analysis of Quantitative Measures Associated with Rules. In: Zhong, N., Zhou, L. (eds.) Proceedings of PAKDD'99. LNCS (LNAI), vol. 1574, pp. 479–488. Springer, Heidelberg (1999)
15. Yao, Y.Y., Sai, Y.: On Mining Ordering Rules. In: Terano, T., Nishida, T., Namatame, A., Tsumoto, S., Ohsawa, Y., Washio, T. (eds.) New Frontiers in Artificial Intelligence. LNCS (LNAI), vol. 2253, pp. 316–321. Springer, Heidelberg (2001)
16. Yao, Y.Y.: Mining High Order Decision Rules. In: Inuiguchi, M., Hirano, S., Tsumoto, S. (eds.) Rough Set Theory and Granular Computing, pp. 125–135. Springer, Berlin (2003)

Attribute Reduction Based on Fuzzy Rough Sets

Degang Chen[1], Xizhao Wang[2], and Suyun Zhao[2]

[1] Department of Mathematics and Physics, North China Electric Power University,
Beijing, 102206,P.R. China
chengdegang@263.net

[2] Department of Mathematics and Computer Science, Hebei University, Baoding,
Hebei, 071002, P.R. China

Abstract. In $T-$fuzzy rough sets a fuzzy $T-$similarity relation is employed to describe the similar degree between two objects and to construct lower and upper approximations for arbitrary fuzzy sets. The existing researches on $T-$fuzzy rough sets mainly concentrate on constructive and axiomatic approaches of lower and upper approximation operators. In this paper we define attribute reduction based on $T-$fuzzy rough sets. The structure of proposed attribute reduction is investigated in detail by the approach of discernibility matrix. At last an example is proposed to illustrate our idea in this paper.

Keywords: Rough sets, fuzzy rough sets, attribute reduction, discernibility matrix.

1 Introduction

Fuzzy rough sets mainly deal with databases with fuzzy nature such as attributes with real values[8], and the traditional rough set approaches[1-4] will have difficulty to handle this kind of databases. Due to its wide applied background, fuzzy rough set has drown more and more attentions from both theoretical and practical fields [5,6,8-16].

There are mainly two topics in fuzzy rough sets theory, one is approximations of arbitrary fuzzy set; the other is attribute reduction with fuzzy rough sets, including reasonable definition of reduction and algorithm to compute reducts. Research on fuzzy rough sets is begun by construction of approximations of arbitrary fuzzy set. Instead of the equivalence relation in crisp rough sets, a fuzzy similarity relation is employed to describe the similar degree between two objects and to construct lower and upper approximations for arbitrary fuzzy sets. Dubois and Prade [9, 10] was one of the first researchers to propose the concept of fuzzy rough sets, they constructed a pair of upper and lower approximation operators of fuzzy sets with respect to a fuzzy similarity relation by using the $t-$norm Min and its dual conorm Max, these two approximation operators were studied in detail from constructive and axiomatic approaches in [11, 12]. Noticed that Min and Max are special $t-$norm and $t-$conorm, Radzikowska and Kerre[15] presented a more general framework to the fuzzification of rough sets. Specifically,

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 381–390, 2007.

they defined a broad family of fuzzy rough sets with respect to a fuzzy similarity relation, each one of which is determined by an implicator and a $t-$norm. Moris and Yakout[13] studied a set of axioms on fuzzy rough sets, their works were restricted to $T-$fuzzy rough set defined by fuzzy $T-$similarity relations. It is claimed in [13] that the upper and lower approximation operators are not dual with each other. Another upper approximation operator was proposed in [14] to obtain the dual upper approximation operator of the lower approximation operator in [13]. These researches on fuzzy rough set have been concluded and studied completely from constructive, axiomatic, lattice and fuzzy topological viewpoints in [6].

The second topic in existing research of fuzzy rough sets is attribute reduction with fuzzy rough sets. Since fuzzy rough set can deal with more complex practical problems than traditional rough sets, it is clearly that development of attribute reduction with fuzzy rough sets is meaningful from both theoretical and practical viewpoints. Comparing with research on approximations of fuzzy sets, less effort has been put on attribute reduction with fuzzy rough sets. In [5] a dependency function was proposed to design algorithm to compute reduct with fuzzy rough sets in [9,10], and this algorithm had been discussed in [16] from the viewpoint of computability and has been pointed out not to converge in many real datasets due to its poorly designed termination criteria. The concept of fuzzy rough sets is put forward on a compact computational domain in [16], which is then utilized to improve the computational efficiency in [5]. However, both [5] and [16] had not proposed a clearly definition of attribute reduction with fuzzy rough sets and the structure of reduct is not clear; many interesting and important topics were not discussed. For example, the core of reduct which plays an important role when considering attribute reduction in traditional rough set theory has not been considered in both [5] and [16]. All of attribute reductions in [5] and [16] employ the fuzzy rough sets in [9,10], and this fuzzy rough sets is defined by a $Min-$fuzzy similarity relation, i.e., the transitivity of fuzzy similarity relation is characterized by the triangular norm Min. As argued in [8,21,22], the triangular norm Min may not always be the best selection to characterize the transitivity. This statement motivates our idea in this paper to consider an arbitrary triangular norm T instead of the triangular norm Min. It is pointed in [6] that there are two lower approximation operators and two upper approximation operators in the existing fuzzy rough sets, we only consider the lower and upper approximation operators in [13] and call them $T-$fuzzy rough sets in the rest of this paper.

In this paper we mainly focus on attribute reduction based on $T-$fuzzy rough sets. We define attribute reduction based on $T-$fuzzy rough sets and its relative concepts such as core of reduction. The structure of reduct is completely studied and an algorithm using discernibility matrix to compute all the attribute reducts is developed. Thus a solid mathematical foundation is set up for attribute reduction based on $T-$fuzzy rough sets for its further application. An example is also employed to illustrate our idea in this paper.

This paper isstructured as following. In Section 2 we mainly review basic content about $T-$fuzzy rough sets In Section 3 we study the structure of attribute reduction based on $T-$fuzzy rough sets by the approach of discernibility matrix. In Section 4 an illustrated example is proposed.

2 On $T-$Fuzzy Rough Sets

In this section we only review the basic contents of $T-$fuzzy rough sets found in [13] and attributes reduction with $Min-$fuzzy rough sets, a detail review of the existing fuzzy rough sets can be found in [6], we omit discussion on this topic and refer the readers to [6] for the length of this paper.

A triangular norm, or shortly $t-$norm, is an increasing, associative and commutative mapping $T : [0, 1] \times [0, 1] \to [0, 1]$ that satisfies the boundary condition $(\forall x \in [0, 1], T(x, 1) = x)$. The most popular continuous $t-$norms are:

- the standard *min* operator $T_M(x, y) = \min\{x, y\}$(the largest $t-$norm),
- the algebraic product $T_P(x, y) = x \cdot y$,
- the bold intersection(also called the Lukasiewicz $t-$norm) $T_L(x, y) = \max\{0, x + y - 1\}$.

Given a triangular norm T, $\vartheta_T(\alpha, \gamma) = \sup\{\theta \in I : T(\alpha, \theta) \le \gamma\}$,$\alpha, \gamma \in [0, 1]$ is called a $R-$implicator based on T. If T is lower semi-continuous, then ϑ_T is called the residuation implication of T, or the $T-$residuated implication. The properties of $T-$residuated implication ϑ_T are listed in[13].

Suppose U is a nonempty universe. A $T-$fuzzy similarity relation R is a fuzzy relation on U which is reflexive($R(x, x) = 1$), symmetric($R(x, y) = R(y, x)$) and $T-$transitive ($R(x, y) \ge T(R(x, z), R(z, y))$), for every $x, y, z \in U$. If ϑ is the T-residuated implication of a lower semi-continuous $t-$norm T, then the lower and upper approximation operators were defined as for every $A \in F(U)$, $\underline{R_\vartheta}A(x) = \inf_{u\in U} \vartheta(R(u, x), A(u)), \overline{R_T}A(x) = \sup_{u\in U} T(R(u, x), A(u))$

In [6,13] these two operators were studied in detail from constructive and axiomatic approaches, we only list their properties as following.

Theorem 1[6,13,17]. Suppose R is a fuzzy $T-$similarity relation. The following statements hold:

1)$\underline{R_\vartheta}(\underline{R_\vartheta}A) = \underline{R_\vartheta}A, \overline{R_T}(\overline{R_T}A) = \overline{R_T}A$;

2) Both of $\underline{R_\vartheta}$ and $\overline{R_T}$ are monotone;

3)$\overline{R_T}(\underline{R_\vartheta}A) = \underline{R_\vartheta}A, \underline{R_\vartheta}(\overline{R_T}A) = \overline{R_T}A$;

4)$\overline{R_T}A = A \Leftrightarrow \underline{R_\vartheta}A = A$;

5)$\underline{R_\vartheta}A = \cup\{\overline{R_T}x_\lambda : \overline{R_T}x_\lambda \subseteq A\}, \overline{R_T}A = \cup\{\overline{R_T}x_\lambda : x_\lambda \subseteq A\}$, here x_λ is a fuzzy set defined as $x_\lambda(y) = \begin{cases} \lambda, y = x \\ 0, y \ne x \end{cases}$.

Theorem 2[6]. For two $T-$similarity relations R_1 and R_2, the following statements are equivalent:

1)$R_1 \subseteq R_2$; 2)$\underline{R_{1\vartheta}}A \supseteq \underline{R_{2\vartheta}}A$; 3)$\overline{R_{1T}}A \subseteq \overline{R_{2T}}A$.

In [5] an algorithm of computing relative reduct was proposed for fuzzy rough sets. This algorithm employs the idea in relative reduction of rough sets to keep the dependence function invariant. Keeping this idea in mind, a QUICKREDUCT algorithm is designed to compute the reduct. However, this algorithm can obtain only one reduct. Thus it is unclear which attribute in the reduct is indispensable, i.e., the core of reduct is unknown. On the other hand, this algorithm lacks mathematical foundation and theoretical analysis, and many interesting topics relative to reduction are not discussed.

The algorithm in [5] has been discussed in [16] from the viewpoint of computability and has been pointed out not to converge in many real datasets due to its poorly designed termination criteria. In [16] they first improve the definition of lower approximation operator in [10] on a compact computational domain. Based on this idea, they design an algorithm with a shorter running time than the algorithm in [5] on some datasets they proposed.

It should be pointed out that a clear definition of attributes reduction with fuzzy rough sets and its relative concepts such as core cannot be found in both [5] and [16]. So the structure of reduction is not clear, this may influence the further application of attribute reduction with fuzzy rough sets. As well known, in traditional rough sets theory, approach of discernibility matrix is employed to investigate structure of reduction and an algorithm can be designed to compute all the reductions by the discernibility matrix [7]. And to our knowledge this method has not been employed to study the structure of reduct with fuzzy rough sets.

In the definition of a fuzzy $T-$similarity relation, the triangular norm T controls the selection of similarity, and different triangular norm T identify different kind of similarity. And a reasonable selection of similarity, i.e., selection of triangular norm T, will capture the connections among data sufficiently. However, it is pointed in [8] that the triangular Min may not always be a reasonable selection, thus we consider an arbitrary triangular norm instead of Min to develop attribute reduction so that different selections are available.

3 On Attribute Reduction Based on Fuzzy Rough Sets

In this section we will define attribute reduction based on $T-$fuzzy rough sets for fuzzy decision system and propose some equivalence conditions to describe the structure of attribute reduction. We also develop an algorithm using discernibility matrix to compute all the attribute reducts.

Following the attributes with real values will be called fuzzy attributes. For every fuzzy attribute, a fuzzy $T-$similarity relation can be employed to measure the similar degree between every pair of objects [8]. If we substitute every fuzzy attribute by its corresponding fuzzy $T-$similarity relation and substitute the decision attribute by its corresponding equivalence relation, we can get a $T-$fuzzy decision system consisting of three parts, a finite universe of discourse, a family of conditional fuzzy attributes and a symbolic decision attribute. Thus every dataset with real value conditional attributes and symbolic decision attribute can be expressed as a $T-$fuzzy decision system so that it is convenient to deal with by techniques of $T-$fuzzy rough sets.

Two key problems must be solved before we define attribute reduction based on $T-$fuzzy rough sets. One is what should be invariant after reduction. We employ the idea in traditional rough sets of keeping the positive region of decision attribute invariant to define relative reduction with $T-$fuzzy rough sets; here the positive region of decision attribute will be defined as the union of lower approximations of decision classes. Another problem is the selection of aggregation operator for several $T-$fuzzy similarity relations. By Theorem 2, a smaller fuzzy $T-$similarity relation can provide more precise lower approximations, thus triangular Min is a reasonable selection of aggregation operator for several fuzzy $T-$similarity relations. We can define attribute reduction for $T-$fuzzy decision system based on $T-$fuzzy rough sets with these discussions.

Suppose U is a finite universe of discourse, $\mathbf{R}$ is a finite set of fuzzy $T-$similarity relations called conditional attributes set, D is an equivalence relation called decision attribute with symbolic values, then $(U, \mathbf{R} \cup D)$ is called a $T-$fuzzy decision system. Denote $Sim(\mathbf{R}) = \cap\{R : R \in \mathbf{R}\}$, then $Sim(\mathbf{R})$ is also a fuzzy $T-$similarity relation. Suppose $[x]_D$ is the equivalence class with respect to D for $x \in U$, then the positive region of D relative to $Sim(\mathbf{R})$ is defined as $Pos_{Sim(\mathbf{R})}D = \cup_{x \in U} \underline{Sim(\mathbf{R})_{\vartheta}}([x]_D)$. We will say that R is dispensable relative to D in $\mathbf{R}$ if $Pos_{Sim(\mathbf{R})}D = Pos_{Sim(\mathbf{R}-\{R\})}D$, otherwise we will say R is indispensable relative to D in $\mathbf{R}$. The family $\mathbf{R}$ is independent relative to D if each $R \in \mathbf{R}$ is indispensable relative to D in $\mathbf{R}$; otherwise $\mathbf{R}$ is dependent relative to D. $\mathbf{P} \subset \mathbf{R}$ is an attributes reduct of relative to D if $\mathbf{P}$ is independent relative to D and $Pos_{Sim(\mathbf{R})}D = Pos_{Sim(\mathbf{P})}D$, for short we call $\mathbf{P}$ a relative reduct of $\mathbf{R}$. The collection of all the indispensable elements relative to D in $\mathbf{R}$ is called the core of $\mathbf{R}$ relative to D, denoted as $Core_D(\mathbf{R})$. Similar to the result in traditional rough sets we have $Core_D(\mathbf{R}) = \cap \mathrm{Red}_D(\mathbf{R})$, $Red_D(\mathbf{R})$ is the collection of all relative reducts of $\mathbf{R}$. Following we study under what conditions that $\mathbf{P} \subset \mathbf{R}$ could be a relative reduct of $\mathbf{R}$.

By (5) of Theorem 1 we know that $\{\overline{R_T} x_\lambda : x \in U, \lambda \in (0, 1]\}$ could be the basic granular set to construct lower and upper approximations of fuzzy sets since every lower or upper approximation is just the union of fuzzy sets with the form as $\overline{R_T} x_\lambda$. Thus the structure of lower approximation of every $[x]_D$ is clear by $\underline{R_\vartheta}([x]_D) = \cup\{\overline{R_T}(y_\lambda) : \overline{R_T}(y_\lambda) \subseteq [x]_D\}$. For $y \notin [x]_D$, clearly $\underline{R_\vartheta}([x]_D)(y) = 0$ holds. For $y \in [x]_D$, the following theorem develops a sufficient and necessary condition for$\overline{R_T}(y_\lambda) \in \underline{R_\vartheta}([x]_D)$.

Theorem 3. Suppose $y \in [x]_D, \overline{R_T}(y_\lambda) \subseteq \underline{R_\vartheta}([x]_D)$, if and only if $\overline{R_T}(y_\lambda)(z) = 0$ for $z \notin [x]_D$.

Proof. If $\overline{R_T}(y_\lambda) \subseteq \underline{R_\vartheta}([x]_D)$, then $\overline{R_T}(y_\lambda)(z) = 0$ for $z \notin [x]_D$ is clear. Conversely, suppose$\overline{R_T}(y_\lambda)(z) = 0$ for $z \notin [x]_D$. We have $\overline{R_T}(y_\lambda) \subseteq [x]_D$ by $[x]_D(u) = 1$ for every $u \in [x]_D$, this implies $\overline{R_T}(y_\lambda) \subseteq \underline{R_\vartheta}([x]_D)$ hold. According to Theorem 3, $y_\lambda \subseteq \underline{R_\vartheta}([x]_D)$ if and only if $\overline{R_T}(y_\lambda)(z) = 0$ for $z \notin [x]_D$. This statement is the key point to develop sufficient and necessary condition for relative reduction with $T-$fuzzy rough sets.

Theorem 4. Suppose $\mathbf{P} \subset \mathbf{R}$,$Pos_{Sim(\mathbf{R})}D = Pos_{Sim(\mathbf{P})}D$ if and only if $\overline{Sim(\mathbf{P})_T}(x_{\lambda(x)}) \subseteq [x]_D$ for every $x \in U$, here $\lambda(x) = \underline{Sim(\mathbf{R})_\vartheta}([x]_D)(x)$.

Proof. Since every two different decision classes have empty overlap, we know to keep $Pos_{Sim(\mathbf{R})}D = Pos_{Sim(\mathbf{P})}D$ is equivalent to keep $\underline{Sim(\mathbf{R})_\vartheta}([x]_D) = \underline{Sim(\mathbf{P})_\vartheta}([x]_D)$ for every $x \in U$, and the latter statement is equivalent to $\overline{Sim(\mathbf{P})_T}(y_{\lambda(y)}) \subseteq [x]_D$ for $y \in [x]_D$, and it is equivalent to $\overline{Sim(\mathbf{P})_T}(x_{\lambda(x)}) \subseteq [x]_D$ for every $x \in U$ since $y \in [x]_D$ implies $[x]_D = [y]_D$.

Thus we have the following theorem to characterize the relative reduction by Theorem 3 and Theorem 4.

Theorem 5. Suppose $\mathbf{P} \subset \mathbf{R}$, then $\mathbf{P}$ contains a relative reduction of $\mathbf{R}$ if and only if $\overline{Sim(\mathbf{P})_T}(x_{\lambda(x)})(z) = 0$ for every $x, z \in U$ and $z \notin [x]_D$, here $\lambda(x) = \underline{Sim(\mathbf{R})_\vartheta}([x]_D)(x)$.

Theorem 6. Suppose $\mathbf{P} \subset \mathbf{R}$, then $\mathbf{P}$ contains a relative reduction of $\mathbf{R}$ if and only if there exists $p \in \mathbf{P}$ such that $T(P(x,z), \lambda(x)) = 0$ for every $x, z \in U$ and $x \notin [x]_D$.

Proof. For $z \notin [x]_D$, $\overline{Sim(\mathbf{P})_T}(x_{\lambda(x)})(z) = \sup_{y \in U} T(Sim(\mathbf{P})(z,y), x_{\lambda(x)}(y)) = T(Sim(\mathbf{P})(x,z), \lambda(x)) = \min\{T(P(x,z), \lambda(x)) : P \in \mathbf{P}\}$, thus we finish the proof.

Clearly $\mathbf{P}$ is a relative reduction of $\mathbf{R}$ if and only if $\mathbf{P}$ is the minimal subset of $\mathbf{R}$ satisfying conditions in Theorem 5 and Theorem 6. And condition in Theorem 6 is can be employed to design algorithm to compute reducts.

With above discussion, we can design an algorithm to compute all the relative reductions. Suppose $U = \{x_1, x_2, ..., x_n\}$, $R = \{R_1, R_2, ..., R_m\}$. By $M_D(U, \mathbf{R})$ we denote a $n \times n$ matrix (c_{ij}), called the discernibility matrix of $(U, \mathbf{R} \bigcup \mathbf{D})$, such that

1) $c_{ij} = \{R \in \mathbf{R} : T(R(x_i, x_j), \lambda(x_i)) = 0\}$ if $x_j \notin [x_i]_D$;
2) $c_{ij} = \emptyset$, otherwise.

$M_D(U, \mathbf{R})$ may not be symmetric and clearly $c_i i = \emptyset$. $R \in c_{ij}$ implies $\overline{R_T}((x_i)_{\lambda(x_i)})(x_j) = 0$, thus c_{ij} is the collection of conditional attributes to ensure $\overline{R_T}((x_i)_{\lambda(x_i)})(x_j) = 0$ for $x_j \notin [x_i]_D$.

A discernibility function $f_D(U, \mathbf{R})$ for $(U, \mathbf{R} \bigcup D)$ is a Boolean function of m Boolean variables $\overline{R_1}, \overline{R_2}, ..., \overline{R_m}$ corresponding to the fuzzy attributes $R_1, R_2, ..., R_m$ respectively, and defined as follows $f_D(U, \mathbf{R})(\overline{R_1}, \overline{R_2}, ..., \overline{R_m}) = \wedge\{\vee(c_{ij}) : c_{ij} \neq 0\}$, where $\vee(c_{ij})$ is the disjunction of all variables $\overline{R}$ such that $R \in c_{ij}$. In the sequel we shall write R_i instead of $\overline{R_i}$ when no confusion can arise.

We have the following theorem for the relative core.

Theorem 7. $Core_D(\mathbf{R}) = \{R : c_{ij} = \{R\}\}$ for some $1 \leq i, j \leq n$.

Proof. $R \in Core_D(\mathbf{R}) \Leftrightarrow Pos_{Sim(\mathbf{R})}D \neq Pos_{Sim(\mathbf{R}-\{R\})}D \Leftrightarrow$ there exists $x_i \in U$ such that $R \in Core_D(\mathbf{R}) \Leftrightarrow Pos_{Sim(\mathbf{R})}D \neq Pos_{Sim(\mathbf{R}-\{R\})}D \Leftrightarrow$ there exists $x_j \in U$ such that $T(R(x_i, x_j), \lambda(x_i)) = 0$, and $T(R'(x_i, x_j), \lambda(x_i)) > 0$ holds for any other $R' \in \mathbf{R} \Leftrightarrow c_{ij} = \{R\}$. The statement $c_{ij} = \{R\}$ implies that R is the unique attribute to maintain $T(R(x_i, x_j), \lambda(x_i)) = 0$.

Theorem 8. Suppose $\mathbf{P} \subset \mathbf{R}$, then $\mathbf{P}$ contains a relative reduction of $\mathbf{R}$ if and only if $\mathbf{P} \bigcap c_{ij} \neq \emptyset$ for every $c_{ij} \neq \emptyset$.

The proof is straightforward by Theorem 6 and definition of c_{ij}.

Theorem 9. Suppose $\mathbf{P} \subset \mathbf{R}$, then $\mathbf{P}$ contains a relative reduction of $\mathbf{R}$ if and only if $\mathbf{R}$ is the minimal set satisfying $\mathbf{P} \bigcap c_{ij} \neq \emptyset$ for every $c_{ij} \neq \emptyset$.

Let $g_D(U, \mathbf{R})$ be the reduced disjunctive form of $f_D(U, \mathbf{R})$ obtained from $f_D(U, \mathbf{R})$ by applying the multiplication and absorption laws as many times as possible. Then there exist l and $\mathbf{R}_k \subseteq \mathbf{R}$ for $k = 1, 2, ..., l$ such that $g_D(U, \mathbf{R}) = (\wedge \mathbf{R}_1) \vee ... \vee (\wedge \mathbf{R}_l)$ where every element in $\mathbf{R}_k$ only appears one time.

We have the following theorem.

Theorem 10. $Red_D(\mathbf{R}) = \{\mathbf{R}_1, ..., \mathbf{R}_l\}$

Proof. For every $k = 1, 2, ..., l$ and $c_{ij} \neq \emptyset$, we have $\wedge \mathbf{R}_k \leq \vee c_{ij}$ by $\vee_{k=1}^{l}(\wedge \mathbf{R}_k) = \wedge\{\vee c_{ij} : c_{ij} \neq \phi\}$, so $\mathbf{R}_k \cap c_{ij} \neq \phi$ for every $c_{ij} \neq \emptyset$. Let $R \in \mathbf{R}_k$ and $\mathbf{R}'_k = \mathbf{R}_k - \{R\}$, then $g_D(U, \mathbf{R}) < \overset{k-1}{\underset{r=1}{\vee}}(\wedge R_r) \vee (\wedge R'_k) \vee (\overset{l}{\underset{r=k+1}{\vee}}(\wedge R_r))$. If for every $c_{ij} \neq \emptyset$, we have $\mathbf{R}'_k \cap c_{ij} \neq \emptyset$, then $\wedge \mathbf{R}'_k \leq \vee c_{ij}$ for every $c_{ij} \neq \emptyset$. This implies $g_D(U, \mathbf{R}) \geq \overset{k-1}{\underset{r=1}{\vee}}(\wedge R_r) \vee (\wedge R'_k) \vee (\overset{l}{\underset{r=k+1}{\vee}}(\wedge R_r))$ which is a contradiction. Hence there exists $c_{i_0 j_0} \neq \emptyset$ such that $\mathbf{R}'_k \cap c_{i_0 j_0} = \emptyset$ which implies $\mathbf{R}_k$ is a relative reduction of $\mathbf{R}$.

For every $\mathbf{X} \in Red_D(\mathbf{R})$, we have $\mathbf{X} \bigcap c_{ij} \neq \emptyset$ for every $c_{ij} \neq \emptyset$, this implies $\wedge \mathbf{X} \leq g_D(U, \mathbf{R})$. Suppose for every $k = 1, 2, ..., l$ we have $\mathbf{X}_k - \mathbf{X} \neq \emptyset$, then for every $k = 1, 2, ..., l$ one can find $R_k \in \mathbf{X}_k - \mathbf{X}$. By rewriting $g_D(U, \mathbf{R}) = (\vee_{k=1}^{l} R_k) \wedge \emptyset$, we have $\wedge \mathbf{X} \leq \vee_{k=1}^{l} R_k$. So there is R_{k_0}such that $\wedge \mathbf{X} \leq R_{k_0}$, this implies $R_{k_0} \in \mathbf{X}$ which is a contradiction. So there exists k' such that $\mathbf{R}_{k'} - \mathbf{X} = \emptyset$, which implies $\mathbf{R}_{k'} \subseteq \mathbf{X}$. Since both $\mathbf{X}_{k'}$ and $\mathbf{X}$ are relative reductions, we have $\mathbf{R}_{k'} = \mathbf{X}$. Hence we have $Red_D(\mathbf{R}) = \{\mathbf{R}_1, ..., \mathbf{R}_l\}$.

Remark. If every $T-$fuzzy similarity relation in $\mathbf{R}$ is a crisp equivalence relation, then the lower approximation is just the crisp one in traditional rough sets, and our method in this section coincides with the crisp one found in [7]. Thus our idea and method are really the generalization of the crisp one found in [7] for fuzzy case.

4 An Illustrated Example

To keep the length of this paper, we will not discuss computational complexity of the proposed algorithm in this paper; this will be our future work. Following we employ an example to illustrate our idea in this paper.

Example 4.1. Let us consider an evaluation problem of credit card applicants. Suppose $U = \{x_1, x_2, ..., x_9\}$ is a set of nine applicants, every applicant is described by six fuzzy attributes: C_1=best education, C_2=better education, C_3=good education, C_4=high salary, C_5= middle salary and C_6=low salary. The membership degrees of every applicant are given in the following table.

Table 1. Samples of credit card evaluation problem

	c_1	c_2	c_3	c_4	c_5	c_6
x_1	0.8	0.1	0.1	0.5	0.2	0.3
x_2	0.3	0.5	0.2	0.8	0.1	0.1
x_3	0.2	0.2	0.6	0.7	0.3	0.2
x_4	0.6	0.3	0.1	0.2	0.5	0.3
x_5	0.3	0.4	0.3	0.3	0.6	0.1
x_6	0.2	0.3	0.5	0.3	0.5	0.2
x_7	0.3	0.3	0.4	0.2	0.6	0.2
x_8	0.3	0.4	0.3	0.1	0.4	0.5
x_9	0.3	0.2	0.5	0.4	0.4	0.2

Every fuzzy attribute C_k can define a T_L-fuzzy similarity relation R_k as $R_k(x_i, x_j) = 1 - |C_k(x_i) - C_k(x_j)|$, $Sim(\mathbf{R})$ is computed as follows

$$(Sim(\mathbf{R})(x_i, x_j)) = \begin{pmatrix} 1 & 0.5 & 0.4 & 0.7 & 0.7 & 0.4 & 0.5 & 0.5 & 0.5 \\ & 1 & 0.6 & 0.4 & 0.5 & 0.5 & 0.4 & 0.3 & 0.6 \\ & & 1 & 0.5 & 0.6 & 0.6 & 0.5 & 0.4 & 0.7 \\ & & & 1 & 0.7 & 0.6 & 0.7 & 0.7 & 0.6 \\ & & & & 1 & 0.8 & 0.9 & 0.6 & 0.8 \\ & & & & & 1 & 0.9 & 0.7 & 0.9 \\ & & & & & & 1 & 0.7 & 0.8 \\ & & & & & & & 1 & 0.7 \\ & & & & & & & & 1 \end{pmatrix}$$

Suppose a decision partition is $A = \{x_1, x_2, x_4, x_7\}, B = \{x_3, x_5, x_6, x_8, x_9\}$ then

$$Sim(\mathbf{R})_*(A)(x) = \begin{cases} 0.3, x = x_1 \\ 0.4, x = x_2 \\ 0.3, x = x_4 \\ 0.1, x = x_7 \\ 0, otherwise \end{cases}, Sim(\mathbf{R})_*(B)(x) = \begin{cases} 0.4, x = x_3 \\ 0.1, x = x_5 \\ 0.1, x = x_6 \\ 0.3, x = x_8 \\ 0.2, x = x_9 \\ 0, otherwise \end{cases},$$

and the discernibility matrix of (c_{ij}) is as follows:

$$\begin{pmatrix}
\phi & \phi & \{1,2,4\} & \phi & \{1,2,5\} & \{1,3,5\} & \phi & \{1,2,4\} & \{1,3\} \\
\phi & \phi & \{3\} & \phi & \{4,5\} & \{4,5\} & \phi & \{4,6\} & \{4\} \\
\{1,3\} & \{3\} & \phi & \{1,3,4\} & \phi & \phi & \{4\} & \phi & \phi \\
\phi & \phi & \{1,3,4\} & \phi & \{1\} & \{1,3\} & \phi & \{1\} & \{1,3\} \\
\{1,2,3,4,5,6\} & \{2,3,4,5\} & \phi & \{1,2,3,4,5,6\} & \phi & \phi & \{2,3,4,6\} & \phi & \phi \\
\{1,2,3,4,5,6\} & \{1,2,3,4,5,6\} & \phi & \{1,3,4,6\} & \phi & \phi & \{1,3,4,5\} & \phi & \phi \\
\phi & \phi & \{1,2,3,4,5\} & \phi & \{2,3,4,6\} & \{1,3,4,5\} & \phi & \{2,3,4,5,6\} & \{2,3,4,5\} \\
\{1,2,4\} & \{4,5,6\} & \phi & \{1\} & \phi & \phi & \{6\} & \phi & \phi \\
\{1,3,5\} & \{2,3,4,5\} & \phi & \{1,3,4\} & \phi & \phi & \{4,5\} & \phi & \phi
\end{pmatrix}$$

Where $i \in c_{ij}$ means $R_i \in c_{ij}, (i = 1, ..., 6)$. We can get that $\{1, 3, 4, 6\}$ is the only reduction of $\mathbf{R}$.

References

[1] Pawlak, Z.: Rough Sets. Internat. J. Comput. Inform. Sci. 11(5), 341–356 (1982)

[2] Pawlak, Z.: Rough Sets: Theoretical aspects of Reasoning about Data. Kluwer academic Publishers, Dordrecht (1991)

[3] Slowinski, R. (ed.): Intelligent decision support: Handbook of applications and advances of the rough sets theory. Kluwer Academic Publishers, Boston (1992)

[4] Ziarko, W.P. (ed.): Rough sets, fuzzy sets and knowledge discovery. In: Workshop in Computing. Springer, London (1994)

[5] Jensen, R., Shen, Q.: Qiang Shen: Fuzzy-rough attributes reduction with application to web categorization. Fuzzy Sets. and Systems 141, 469–485 (2004)

[6] Yeung, D.S., Degang, C., Tsang, C.C., Lee, W.T., Xizhao, W.: On the Generalization of Fuzzy Rough Sets. IEEE Trans. on Fuzzy Systems 13(3), 343–361 (2005)

[7] Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, pp. 331–362. Kluwer Academic Publishers, Dordrecht (1992)

[8] Fernandez Salido, J.M., Murakami, S.: Rough set analysis of a general type of fuzzy data using transitive aggregations of fuzzy similarity relations. Fuzzy Sets and Systems 139, 635–660 (2003)

[9] Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. Internat. J. Genaral Systems 17(2-3), 191–209 (1990)

[10] Dubois, D., Prade, H.: Putting rough sets and fuzzy sets together. In: Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, Kluwer Academic Publishers, Dordrecht (1992)

[11] Weizhi, W., Wenxiu, Z.: Constructive and axiomatic approaches of fuzzy approximation operators. Information Sciences 159(3-4), 233–254 (2004)

[12] Weizhi, W., Jusheng, M., Wenxiu, Z.: Generalized fuzzy rough sets. Information Sciences 151, 263–282 (2003)

[13] Morsi, N.N., Yakout, M.M.: Axiomatics for fuzzy rough sets. Fuzzy Sets and Systems 100, 327–342 (1998)

[14] Jusheng, M., Wenxiu, Z.: An axiomatic characterization of a fuzzy generalization of rough sets. Information Sciences 160(1-4), 235–249 (2004)

[15] Radzikowska, A.M., Kerre, E.E.: A comparative study of fuzzy rough sets. Fuzzy Sets and Systems 126, 137–155 (2002)

[16] Rajen, B.B., Gopal, M.: On fuzzy rough sets approach to feature selection. Pattern Recognition Letters 26(7), 965–975 (2005)

[17] Degang, C., Wenxiu, Z., Yeung, D.S., Tsang, E.C.C.: Rough approximations on a complete completely distributive lattice with applications to generalized rough sets. Information Sciences 176, 1829–1848 (2006)

[18] Greco, S., Inuiguchi, M., Slowinski, R.: A new proposal for fuzzy rough approximations and gradual decision rule representation. In: Transactions on Rough Sets II. LNCS, vol. 3135, pp. 319–342. Springer, Heidelberg (2004)

[19] Cattaneo, G.: Fuzzy extension of rough sets theory. In: RSCTC 1998. LNCS (LNAI), vol. 1424, pp. 275–282. Springer, Heidelberg (1998)

[20] Yao, Y.Y.: Combination of rough and fuzzy sets based on level sets. In: Rough Sets and Data mining: Analysis for Imprecise Data, pp. 301–321. Kluwer Academic Publishers, Dordrecht (1997)

[21] Bezdek, J.C., Harris, J.O.: Fuzzy partitions and relations: an axiomatic basis of clustering. Fuzzy Sets. and Systems 84, 143–153 (1996)

[22] Sudkamp, T.: Similarity, interpolation, and fuzzy rule construction. Fuzzy Sets and Systems 58, 73–86 (1993)

On Logic with Fuzzy and Rough Powerset Monads

Patrik Eklund[1] and M.A. Galán[2,*]

[1] Department of Computing Science, Umeå University, Sweden
peklund@cs.umu.se
[2] Department of Applied Mathematics, University of Málaga, Spain
magalan@ctima.uma.es

Abstract. Non-standard logics departs from traditional logic mostly in extended views, on one hand syntactically related to logical operators, and on the other hand semantically related to truth values. Typical for these approaches is the remaining traditional view on 'sets and relations' and on terms based on signatures. Thus the cornerstones of the languages remain standard, and so does mostly the view on knowledge representation and reasoning using traditional substitution theories and unification styles. In previous papers we have dealt with particular problems such as generalizing terms and substitution, extending our views on sets and relations, and demonstrated the use of these non-standard language elements in various applications such as for fuzzy logic, generalized convergence spaces, rough sets and Kleene algebras. In this paper we provide an overview and summarized picture of what indeed happens when we drop the requirement for using traditional sets with relations and terms with equational settings.

1 Introduction

The use of single points and unstructured set is very rooted in syntactic and semantic aspects of logic, and indeed logic based programming in general. Language constructions are based on point, such as substitution where variables are substituted with single point objects in form of constants or terms. Mathematically one may argue that a complicated term is far from just a singleton, but nevertheless the set of terms over a signature is handled as a set of points, and terms are handled as unambiguous points.

The alternative is to allow variables being substituted with sets of points, or, for that matter, any more or less complicated object build upon points into sets and sets of sets of various kind.

Fuzzy sets are traditionally viewed as mappings from a ground set (or a universe) X to a lattice, frequently assumed at least to be completely distributive. Thus fuzzy sets are represented as $f : X \to L$ without further reflexions on their structural origin. Category theory is an excellent machinery for providing these structural origins and furthermore for representing entities like fuzzy sets in a perspective where fuzzy sets become the tool rather than the target. Arithmetics provides some basic illuminations of this point. With fuzzy sets as targets we are doing 'arithmetics with fuzzy' rather than 'fuzzy arithmetics'. The former means defining extended arithmetic operators that act in some algebraic fashion, and then obviously fuzzy sets are the targets of these

[*] Partially supported by Spanish projects TIC2003-09001-C02-01 and TIN2006-15455-C03-01.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 391–399, 2007.

operators. Doing fuzzy arithmetic in the right sense of these words means using fuzzy sets as tools to identify fuzzy values related to results of algebraic operations. We need to provide a strict categorical formalization to explain this intuitive distinction between fuzzy sets as tools and targets.

Semantically, the situation is equally poor. Semantic domains and even 'universes' are mostly sets and even without any structure. Again, work on semantic domains should invite possibilities for more elaborate use of more complicated sets, in the sense of aiming at modelling real world situations. The situation for logic semantics is, however, not all that light-weighted. Take e.g. generalizing logic with tools for jumping from one logic to the other [24]. Managing strictly all the machineries in one theory including logic operators, entailment, logic consequences, and even logic calculi, is hard enough with conventional themes such as predicate and equational logic. Generalizing from there to a useful and application-oriented involvement of non-classical logic is future work for decades to come.

Keeping logic out of the picture for a while, the obvious mathematical tools for managing 'sets of sets' in a strict formal framework is category theory, and indeed functorial and monadic approaches to sets. The last decade has shown much development in these fields, theoretical applications having been driven by areas of algebra and topology [16]. Recent years show developments also in computer science where a beginning understanding of substitution theories has seen daylight [10].

The purpose of this paper is to present the formal framework underlying the discussion above. The paper is partly a survey of applications developed for sets, topologies and algebras, partly includes new results and also views on future work and trends for these research directions. Section 2 presents the categorical picture of and framework for fuzzy and rough sets. In Section 3 we discuss also other applications apart from fuzzy and rough sets. In Section 4 we discuss the inverses and negation as they appear in our generalized setting. Section 5 describes similarity relations and their use within unifications. Finally, Section 6 concludes the paper.

2 From Sets and Relations Even Beyond Fuzzy and Rough

We will introduce 'complicated sets' by viewing the situation within standard substitution theory. Let $T_\Omega X$ be the set of terms with variables from a set X over the operator domain Ω. Categorically, a substitution σ is a mapping, $\sigma\colon X \to T_\Omega Y$ from the set of variables X to the set of terms $T_\Omega Y$ with variables in the set Y.

Applying σ to a term t (in logic programming usually written as $t\sigma$) means replacing each variable x in t by $\sigma(x)$. With this notation, a unifier of two terms s, t is a substitution σ such that $s\sigma = t\sigma$. The main idea of considering a most general unifier (mgu) is to consider the substitution (if it exists) that does 'as little as possible'. A substitution σ is more general than τ if $\tau = \sigma\bar{\tau}$ for some $\bar{\tau}$ and a unifier of s and t is a mgu if it is more general than all other unifiers.

Substitutions may be composed by replacing variables in terms by terms. Given the substitutions $\sigma_1\colon X \to T_\Omega Y$ and $\sigma_2\colon Y \to T_\Omega Z$, a remark here is that their composition cannot be the usual mapping composition, but there is indeed a 'standard' way of composing them. Naturality properties associated to the term functor make it possible to define the composition of substitution as the following sequence of mappings:

$$X \overset{\sigma_1}{\to} T_\Omega Y \overset{T_\Omega \sigma_2}{\to} T_\Omega T_\Omega Z \overset{\mu_Z}{\to} T_\Omega Z$$

In this case the 'flattening' operator μ_Z is a natural transformation and, since the term functor is idempotent, the transformation is the identity. This corresponds with the classical definition 'terms over terms are terms', i.e. $T_\Omega T_\Omega = T_\Omega$.

Generalizing terms now means departing from singleton terms to various types of sets of terms. This essentially means composing the corresponding set functor, e.g. the powerset functor to take a simple example, with the term functor. The be a bit more strict we need say that these set functors must be extendable to monads[1] (over the category of sets).

An interesting example from fuzzy and rough set point of view is the consideration of generalized substitutions that replace variables by many-valued set of terms, i.e, $\theta \colon X \to L_{id} T_\Omega Y$, where L_{id} denotes the fuzzy powerset functor[2].

Fuzzy sets are traditionally viewed as mappings from a ground set (or a universe) X to a lattice, frequently assumed at least to be completely distributive. Thus fuzzy sets are represented as $f : X \to L$ without further reflections on their structural origin.

Monads where the underlying endofunctor is equipped with an order structure gives further possibilities for applications within management of sets and algebras. A strict definition of partially ordered monads $(\varphi, \leq, \eta, \mu)$ is given in [16]. For the purpose of our paper it is enough to observe that imposing order means having a partial order in $(\varphi X, \leq)$. The classical example of a partially ordered monad is the power set partially ordered monad $(P, \leq, \eta, \mu)$, where PX is the ordinary power set of X and $\leq$ is set inclusion $\subseteq$ making $PX, \leq$ a partially ordered set. The unit $\eta : X \to PX$ is given by $\eta(x) = \{x\}$ and the multiplication $\mu : PPX \to PX$ by $\mu(\mathcal{B}) = \cup \mathcal{B}$.

[1] A *monad* (φ, η, μ) over a category C consists of a covariant functor $\varphi : \mathsf{C} \to \mathsf{C}$, together with natural transformations $\eta : id \to \varphi$ and $\mu : \varphi \circ \varphi \to \varphi$ fulfilling the conditions $\mu \circ \varphi \mu = \mu \circ \mu \varphi$ and $\mu \circ \varphi \eta = \mu \circ \eta \varphi = id_\varphi$.

[2] The many-valued extension of P to L_{id} is as follows. Let L be a completely distributive lattice. The functor L_{id} is obtained by $L_{id} X = L^X$, i.e. the set of mappings $A : X \to L$. The partial order $\leq$ on $L_{id} X$ is given pointwise. Morphism $f \colon X \to Y$ in `Set` are mapped according to

$$L_{id} f(A)(y) = \bigvee_{f(x)=y} A(x).$$

Finally $\eta_X : X \to L_{id} X$ is given by

$$\eta_X(x)(x') = \begin{cases} 1 & \text{if } x' \leq x \\ 0 & \text{otherwise} \end{cases}$$

and $\mu_X : L_{id} X \circ L_{id} X \to L_{id} X$ by

$$\mu_X(\mathcal{M})(x) = \bigvee_{A \in L_{id} X} A(x) \wedge \mathcal{M}(A).$$

Partially ordered monads provide the appropriate categorical formalization for modelling rough sets. To this let R be a relation on X, i.e. $R \subseteq X \times X$. Equivalently, the relation is a mapping $\rho_X : X \to PX$, where $\rho_X(x) = \{y \in X | xRy\}$ and the inverse relation R^{-1} is represented as $\rho_X^{-1}(x) = \{y \in X | xR^{-1}y\}$.

Rough sets can then be described categorically [12]. In the crisp situation, the lower approximation of $A \subseteq X$ is obtained by

$$A^{\downarrow} = \bigvee_{\rho_X(x) \leq A} \eta_X(x)$$

and the upper approximation by

$$A^{\uparrow} = \bigvee_{\rho_X(x) \wedge A > 0} \eta_X(x) = \mu_X \circ P\rho_X^{-1}(A).$$

The corresponding R-weakened and R-substantiated sets of a subset A of X are given by

$$A^{\Downarrow} = \bigvee_{\rho_X^{-1}(x) \leq A} \eta_X(x)$$

and

$$A^{\Uparrow} = \mu_X \circ P\rho_X(A).$$

In a general situation, rough monads are defined by means of the Φ-ρ-upper and Φ-ρ-lower approximations, and further the Φ-ρ-weakened and Φ-ρ-substantiated sets. The following condition (valid for both P and L_{id}) is required for any $f\colon X \to \varphi X$:

$$\varphi f(\bigvee_i a_i) = \bigvee_i \varphi f(a_i).$$

Let $\rho_X : X \to \varphi X$ be a Φ-relation and let $a \in \varphi X$. The inverse must be specified for the given set functor φ. Rough monads are given by:

$$\begin{aligned}
\Uparrow_X (a) &= \mu_X \circ \varphi\rho_X(a) \\
\downarrow_X (a) &= \bigvee_{\rho_X(x) \leq a} \eta_X(x) \\
\uparrow_X (a) &= \mu_X \circ \varphi\rho_X^{-1}(a) \\
\Downarrow_X (a) &= \bigvee_{\rho_X^{-1}(x) \leq a} \eta_X(x)
\end{aligned}$$

Note that in this generalization equivalence relations might need to be softened when moving beyond the ordinary powerset functor.

3 Partially Ordered Monads in Applications

Partially ordered monads are powerful tools for working with topologies and convergence spaces [16,18], Kleene algebras [28,21,7] and rough sets [12].

One can say that the development of partially ordered monads has its origin within the area of topology and convergence structures, originally involving filters [22]. Based on filters, Cauchy structures were initiated in [20]. A general structure theory was presented in [13], where more general set functors for convergence were used. The use of monads makes convergence more powerful [14] and the provision of examples like the fuzzy filter monad as well as techniques for compactification constructions for filter based limit spaces, can be found in [5,6]. The introduction of partially ordered monads and its use within extension structures is due to [16,17], with a follow-up on considerations for compcatifications in [18].

Partially ordered monads are useful also in other areas. They contribute to providing a generalized notion of powerset Kleene algebras. This generalization builds upon a more general powerset functor setting far beyond just strings [21] and relational algebra [28]. Kleene algebras are widely used e.g. in formal languages [27] and analysis of algorithms [1].

Further, these monads contain sufficient structure for modelling rough sets [25] in a generalized setting with set functors. Even for the ordinary relations, the adaptations through partially ordered monads increases the understanding of rough sets in a basic many-valued logic [19] setting.

4 The Role of Boolean Algebras

Classical rough sets and computing with rough sets make use of the fact that PX with its set operations is a Boolean algebra. Further the inverse relation is specific for P and its generalisation is far from obvious. Negation and inverses in the general case, i.e. based on partially ordered monads, thus are not straightforward to define.

Concerning inverses we at least have that

$$\bigvee_{\rho_X(x) \wedge A > 0} \eta_X(x) = \mu_X \circ P\rho_X^{-1}(A)$$

if and only if

$$\rho^{-1}(x) = \bigcup_{x \in \rho(y)} \eta(y).$$

This observation provides the means to define inverse relations.

The generalization from the ordinary power set monad to involving a wide range of set functors and their corresponding partially ordered monads now requires a appropriate management of relational inverses and complement. Logic comes to rescue where complement as negation can be represented as a residual.

Note also that, in the case of the ordinary power set monad,

$$\rho^{-1}(x) = \bigcup_{x \in \rho(y)} \eta(y).$$

Let $\Phi = (\varphi, \leq, \eta, \mu)$ be a partially ordered monad. We say that $\rho_X : X \to \varphi X$ is a *Φ-relation* on X, and by $\rho_X^{-1} : X \to \varphi X$ we denote its *inverse*. The inverse must specified for the given set functor φ.

Example 1. In the case of $\varphi = P$ we have

$$\rho_X^{-1}(x) = \bigvee_{x \in \rho_X(y)} \eta_X(y),$$

and in the case of $\varphi = L^{id}$ we define

$$\rho_X^{-1}(x)(x') = \bigvee_{y \in X} \rho_X(x')(x) \wedge \eta_X(y)(x'),$$

and thus $\rho_X^{-1}(x)(x') = \rho_X(x')(x)$.

We can define a general implication $\longrightarrow: \varphi X \times \varphi X \to \varphi X$ by

$$a \longrightarrow b = \bigvee_{a \wedge x \leq b} x$$

and the negation operator $\neg_X : \varphi X \to \varphi X$ is then logically given by

$$\neg_X(a) = (a \longrightarrow 0).$$

In case of $\varphi = P$, negation is the complement of sets.

The implication $\longrightarrow: \varphi X \times \varphi X \to \varphi X$ fulfills

(i) If $x \leq a$ and $y \leq (a \longrightarrow b)$, then $x \wedge y \leq b$
(ii) $a \wedge x \leq b$ if and only if $x \leq (a \longrightarrow b)$

and it easily seen that the following properties hold:

(i) $a \leq b$ if and only if $(a \longrightarrow b) = 1$
(ii) $(1 \longrightarrow a) = a$

Proposition 1. *(i)* $a \leq \neg_X \neg_X a$
(ii) $a \leq b$ *implies* $\neg_X a \geq \neg_X b$
(iii) $\neg_X(a \vee b) = \neg_X a \wedge \neg_X b$

Proof. To show (i) by the definition of $\neg_X \neg_X a = \bigvee_{\neg_X a \wedge t \leq 0} t$ and the condition fulfilled by the definition of $\neg_X a$, i.e. $a \wedge \neg_X a \leq 0$, we immediately obtain: $a \leq \neg_X \neg_X a$. Property (ii) follows immediately from the definition of the negation and (i). For (iii) we have:

$$\neg_X(a \vee b) = \bigvee_{(a \vee b) \wedge z \leq 0} z$$

In particular we have:

$$(a \vee b) \wedge \neg_X(a \vee b) = (a \wedge \neg_X(a \vee b)) \vee (b \wedge \neg_X(a \vee b)) \leq 0$$

which implies $(a \wedge \neg_X(a \vee b)) \leq 0$ and $(b \wedge \neg_X(a \vee b)) \leq 0$. Now by definition of the negation of a and negation of b we obtain that $\neg_X(a \vee b) \leq \neg_X a$, $\neg_X(a \vee b) \leq \neg_X b$ and therefore

$$\neg_X(a \vee b) \leq \neg_X a \wedge \neg_X b$$

Similarly considering $z = \neg_X a \wedge \neg_X b$ we obtain

$$\neg_X a \wedge \neg_X b \leq \neg_X(a \vee b)$$

Proposition 2. *The implication* $\longrightarrow: \varphi X \times \varphi X \to \varphi X$ *fulfills the following properties:*

(i) $a \longrightarrow (b \longrightarrow a) = 1$
(ii) $\big(a \longrightarrow (b \longrightarrow c)\big) \longrightarrow \big((a \longrightarrow b) \longrightarrow (a \longrightarrow c)\big) = 1$

Proof. For (i), we need to see if $a \leq (b \longrightarrow a)$. This is equivalent to $b \wedge a \leq a$ that always hold. To show condition (ii) we can equivalently show the inequality:

$$\begin{aligned} \bar{x} &= (a \longrightarrow (b \longrightarrow c)) \\ &\leq ((a \longrightarrow b) \longrightarrow (a \longrightarrow c)) = \bigvee_{(a \longrightarrow b) \wedge y \leq (a \longrightarrow c)} y \end{aligned}$$

Let us see that $\bar{x}$ is one of the y, i.e. $\bar{x}$ fulfills the condition:

$$(a \longrightarrow b) \wedge \bar{x} \leq (a \longrightarrow c)$$

By the definition of $\bar{x}$ we have, $a \wedge \bar{x} \leq b \longrightarrow c$. Now, applying some properties of the implication ($a \wedge b = a \wedge (a \longrightarrow b)$) to the condition for $\bar{x}$, we obtain:

$$a \wedge \bar{x} \wedge b = a \wedge (a \longrightarrow b) \wedge \bar{x} \leq c$$

This can now be rewritten into the condition we were searching for.

Proposition 3. *(i) If* $a \leq b$ *then* $a = b \wedge (b \longrightarrow a)$
(ii) We can always write $a \wedge b = a \wedge (a \longrightarrow b)$

Proof. For (i), note that by definition we have

$$b \wedge (b \longrightarrow a) = b \wedge \bigvee_{b \wedge x \leq a} x = \bigvee_{b \wedge x \leq a} b \wedge x \leq a$$

On the other hand by Proposition 2 we know that $a \longrightarrow (b \longrightarrow a) = 1$ which immediately, using properties of the implication, bring us to $a \leq (b \longrightarrow a)$. By hypothesis we have that $a \leq b$ and therefore

$$a \leq b \wedge (b \longrightarrow a)$$

Finally putting together both inequalities we end the proof. To see condition (ii) we will consider two cases. In the case $a \leq b$, this is equivalent to $a \longrightarrow b = 1$ and therefore $a \wedge (a \longrightarrow b) = a$ In the case $a > b$ the result is immediate from (i).

5 Similarities

In previous work we have developed some interesting tools in the abstract language of category theory with the aim of providing a ground foundation to the development of a general framework for unification, working with powersets of terms. At this point, the concept of monad arises as a fundamental one, and managing to provide/construct examples of useful monads turns out to be essential. Suitable composition of monads are shown to provide a concept for generalised term and Kleisli categories are a response for generalised substitutions.

Considering powersets of terms, in [11], similarity relations were presented and some suggestions on extending the concept of unifiers were proposed.

The concept of unifiers can be extended by considering powersets of terms. Generalised terms, as given by powersets of terms, can be handled in equational settings involving substitutions and unifiers. Similarity relations, denoted by E, are particular fuzzy relations between two objects. In the crisp case i.e. $L = P$, similarity relations are equivalence relations: two elements x, y can be either fully similar ($E(x, y) = 1$) or fully dissimilar ($E(x, y) = 0$).

The classical rough set method is based on crisp sets. Going beyond the crisp situation, approximation of sets can be achieved by using fuzzy similarities relations. We, therefore can consider rough approximations between powerset of terms based on similarities relations on $L \circ T$.

In [9] we presented an extension of the generalized powerset monad to the context of partially ordered monads over acSLAT, the category of almost complete semi-lattices. Nevertheless the existence of a partially ordered extension of the term monad remains still as an open question.

6 Future Work

The categorical structure of fuzzy and rough sets opens up for many investigations on generalized views of these sets. There utility in logic and logic programming is underway, and some cornerstone results are already establish.

Future work needs to improve views and properties of logic operators derived from these generalized relational structures. Further, negation and inverse needs to better understood also from application point of view. In general, applications need to be developed, both small and large.

Composing partially ordered monads will play an important role for semantic considerations of programming languages and decision support models involving formal logic and uncertainties. For many-valued extensions, the open question concerning T and further, $L \circ T$ being extendable to a partially ordered monad over acSLAT is an important one.

References

1. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The Design and Analysis of Computer Algorithms. Addison-Wesley, London (1975)
2. Barr, M., Wells, C.: Toposes, Triples and Theories. Springer, Heidelberg (1985)
3. Butzmann, H.P., Kneis, G.: On Čech-Stone compactifications of pseudo-topological spaces. Math. Nachr. 128, 259–264 (1986)
4. Eklund, P., Gähler, W.: Generalized Cauchy spaces. Math. Nachr. 147, 219–233 (1990)
5. Eklund, P., Gähler, W.: Fuzzy Filter Functors and Convergence, Applications of category theory to fuzzy subsets. In: Rodabaugh, S.E., Klement, E.P., Höhle, U. (eds.) Theory and Decision Library B, pp. 109–136. Kluwer, Dordrecht (1992)
6. Eklund, P., Gähler, W.: Completions and Compactifications by Means of Monads. In: Lowen, R., Roubens, M. (eds.) Fuzzy Logic, State of the Art, pp. 39–56. Kluwer, Dordrecht (1993)

7. Eklund, P., Gähler, W.: Partially ordered monads and powerset Kleene algebras. In: Proc. 10th Information Processing and Management of Uncertainty in Knowledge Based Systems Conference (IPMU (2004)
8. Eklund, P., Galán, M.A.: The rough powerset monad. In: Proc. of the 37th International Symposium on Multiple Valued Logics, ISMVL-2007, Oslo (Norway) (Accepted)
9. Eklund, P., Galán, M.A., Gähler, W., Medina, J., Ojeda Aciego, M., Valverde, A.: A note on partially ordered generalized terms. In: Proc. of Fourth Conference of the European Society for Fuzzy Logic and Technology and Rencontres Francophones sur la Logique Floue et ses applications (Joint EUSFLAT-LFA 2005), pp. 793–796
10. Eklund, P., Galán, M.A., Medina, J., Ojeda Aciego, M., Valverde, A.: A categorical approach to unification of generalised terms, Electronic Notes in Theoretical Computer Science 66(5) (2002) URL: http://www.elsevier.nl/locate/entcs/volume66.html
11. Eklund, P., Galán, M.A., Medina, J., Ojeda-Aciego, M., Valverde, A.: Similarities between powersets of terms, Fuzzy Sets and Systems, 144, 213–225 (2004) Possibilistic Logic and Related Issues. Godo, L., Sandri, S. (eds.) (2004)
12. Eklund, P., Galán, M.A.: Monads can be rough. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 77–84. Springer, Heidelberg (2006)
13. Gähler, W.: A topological approach to structure theory. Math. Nachr. 100, 93–144 (1981)
14. Gähler, W.: Monads and convergence. In: Proc. Conference Generalized Functions, Convergences Structures, and Their Applications, Dubrovnik (Yugoslavia), 1987, pp. 29–46. Plenum Press, New York (1988)
15. Gähler, W.: Completion theory, in: Mathematisches Forschungsinstitut Oberwolfach, Tagungsbericht, vol. 48, p. 8 (1991)
16. Gähler, W.: General Topology – The monadic case, examples, applications. Acta Math. Hungar.$\tilde{8}$8, 279–290 (2000)
17. Gähler, W.: Extension structures and completions in topology and algebra, Seminarberichte aus dem Fachbereich Mathematik, Band 70, FernUniversität in Hagen (2001)
18. Gähler, W., Eklund, P.: Extension structures and compactifications. In: Categorical Methods in Algebra and Topology (CatMAT 2000), pp. 181–205 (2000)
19. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer Academic Publishers, Dordrecht (1998)
20. Keller, H.H.: Die Limesuniformisierbarkeit der Limesräume. Math. Ann. 176, 334–341 (1968)
21. Kleene, S.C.: Representation of events in nerve nets and finite automata. In: Shannon, C.E., McCarthy, J. (eds.) Automata Studies, pp. 3–41. Princeton University Press, Princeton (1956)
22. Kowalsky, H.-J.: Limesräume und Komplettierung. Math. Nachr. 12, 301–340 (1954)
23. Manes, E.G.: Algebraic Theories. Springer, Heidelberg (1976)
24. Meseguer, J.: General logics, In: H.-D., et al. (eds.) Logic Colloquium ’87, Ebbinghaus, pp. 275–329, Elsevier, North-Holland (1989)
25. Pawlak, Z.: Rough sets. Int. J. Computer and Information Sciences 5, 341–356 (1982)
26. Rydeheard, D.E., Burstall, R.M.: A categorical unification algorithm. In: Poigné, A., Pitt, D.H., Rydeheard, D.E., Abramsky, S. (eds.) Category Theory and Computer Programming. LNCS, vol. 240, pp. 493–505. Springer, Heidelberg (1986)
27. Salomaa, A.: Two complete axiom systems for the algebra of regular events. J. ACM 13, 158–169 (1966)
28. Tarski, A.: On the calculus of relations. J. Symbolic Logic 6, 65–106 (1941)

A Grey-Rough Set Approach for Interval Data Reduction of Attributes

Daisuke Yamaguchi[1], Guo-Dong Li[1], and Masatake Nagai[2]

[1] Graduate School of Kanagawa University, Department of Engineering, 3-27-1
Rokkakubashi, Kanagawa-ku, Yokohama City, 221–8686 Japan
daicom0204@yahoo.co.jp, guodong_li2004@yahoo.co.jp

[2] Kanagawa University, Faculty of Engineering, 3-27-1 Rokkakubashi, Kanagawa-ku,
Yokohama City, 221–8686 Japan
masatake4263@oregano.ocn.ne.jp

Abstract. Reduction in rough set theory is useful to compact given attributes of large-scale decision tables in data mining. In this paper a new method called grey-rough reduction is proposed for decision tables containing non-interval data and interval data complexly called grey-decision tables. First of all, a grey-rough approximation is introduced after summarized grey numbers, their operations and functions. Two sorts of reduction based on grey-rough sets, a basic approach and advanced approach are proposed with several illustrative examples. Three experiments, compatibility with the classical model, an application of the basic approach to decision-making and influence of the parameter in the advanced approach are shown. The advantages of the proposal are (1) it is compatible with the classical reduction model for non-interval data, (2) it is useful for complex decision tables and (3) it provides a possible reduction of attributes with a parameter by the advanced approach.

1 Introduction

Current database systems become more and more complex and, more and more massive data are stored in them, therefore, finding valuable information from such databases becomes a hard work. In the real applications, many measured values, for example, blood pressure, current, height, mass, power, temperature, time, voltage and weight are often described in interval data to be compacted. Discovering valuable information is an important paradigm in data mining. Recently, a clustering model for interval data is suggested by a number of researchers in terms of Symbolic Data Analysis (SDA) [1]. SDA is a new approach in knowledge discovery, in which data of units called symbolic are dealt with according to Ref. [1]. Clustering methods, decision trees [2,3] and Principle Component Analysis (PCA) for interval data are reported in SDA. Clustering is a useful method to compact records (tuples or objects) in databases, for example, de Souza and de Carvalho [4,5] suggest two clustering methods for interval data, one of which is based on city-block distances and another is based on Chebyshev distances. Asharaf [6] also suggests a clustering method based on rough approximations.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 400–410, 2007.

Rough reduction [7, 8, 9] is also powerful to compact columns (attributes) in databases, because the result of a reduction provides information on useful attributes of a given data set. For example, a reduction for set-valued decision systems are suggested [10, 11], in which set-valued objects are given in decision tables called incomplete information systems or non-deterministic information systems. Given data in such systems, however, are integers. Interval data are usually given as real numbers; thus the reduction approach is needed to be expanded for interval data.

With the above-mentioned motivation, a new reduction approach for interval data based on grey system theory [12, 13, 14, 15] is proposed in this paper. Grey system theory covers grey classification, grey control, grey decision-making, grey prediction, grey structural modeling, grey relational analysis as well as grey-rough sets [16]. One of the main concepts in grey system theory is how systems should be controlled under incomplete or lack of information situation. Grey number denoting an uncertain value is described in interval from this concept. The grey-rough set approach is suitable for interval data reduction of attributes.

2 Grey System Theory

2.1 Grey Numbers and Grey Lattice Operation

First of all, grey numbers and their operations for grey-rough sets are introduced.

Let $\mathbb{U}$ be the universal set, x be an element of $\mathbb{U}(x \in \mathbb{U})$, $\mathbb{R}$ be the set of real numbers and $X \subseteq \mathbb{R}$ be the set of value range that x may hold. Let G be a grey set of $\mathbb{U}$ defined by two mappings of the upper membership function $\overline{\mu}_G(x)$ and the lower membership function $\underline{\mu}_G(x)$ as follows:

$$\left.\begin{array}{l} \overline{\mu}_G(x) : \mathbb{U} \rightarrow [0, 1] \\ \underline{\mu}_G(x) : \mathbb{U} \rightarrow [0, 1] \end{array}\right\} \tag{1}$$

where $\underline{\mu}_G(x) \leq \overline{\mu}_G(x)$ and $x \in \mathbb{U}$. When $\underline{\mu}_G(x) = \overline{\mu}_G(x)$, the grey set becomes a fuzzy set, which means that grey system theory deals with fuzzy situation more flexibly.

When two values $\underline{x}, \overline{x}(\underline{x} = \inf X, \overline{x} = \sup X)$ are given in x, then x using a form $\otimes x = x|_{\underline{\mu}}^{\overline{\mu}}$ is called as follows:

1. If $\underline{x} \rightarrow -\infty$ and $\overline{x} \rightarrow +\infty$, then $\otimes x$ is called a black number
2. If $\underline{x} = \overline{x}$, $\otimes x$ is called a white number or a whitened value denoted by $\tilde{\otimes} x$
3. Otherwise $\otimes x \rightleftarrows [\underline{x}, \overline{x}]$ is called a grey number

where a symbol '$\rightleftarrows$' denotes that the left-hand side interval equals to the right-hand side called the grey lattice coincidence relation. Grey numbers indicate interval data. White numbers usually indicate real numbers; those data are called non-interval data in this paper.

In grey system theory, the grey arithmetic operation [15, 17, 18] and the grey lattice operation [19, 16] are introduced for grey numbers. In this paper, the grey lattice operation is used for Boolean reasoning of grey-rough sets. The operators, *Join* ($\vee$), *Meet* ($\wedge$), *Complement* ($\otimes x^c$) and *Exclusive Join* ($\oplus$) are given for two grey numbers $\otimes x \rightleftarrows [\underline{x}, \overline{x}]$ and $\otimes y \rightleftarrows [\underline{y}, \overline{y}]$ as follows:

$$\otimes x \vee \otimes y \rightleftarrows [\min(\underline{x}, \underline{y}), \max(\overline{x}, \overline{y})] \tag{2}$$

$$\tilde{\otimes} x \vee \tilde{\otimes} y \rightleftarrows [\min(\tilde{\otimes} x, \tilde{\otimes} y), \max(\tilde{\otimes} x, \tilde{\otimes} y)] \tag{3}$$

$$\otimes x \wedge \otimes y \rightleftarrows \begin{cases} [\underline{x}, \overline{x}] & \text{if } \otimes x \to \otimes y \\ [\underline{y}, \overline{y}] & \text{if } \otimes y \to \otimes x \\ [\underline{x}, \overline{y}] \text{ if } \underline{x} \to \otimes y \text{ and } \overline{y} \to \otimes x \\ [\underline{y}, \overline{x}] \text{ if } \underline{y} \to \otimes x \text{ and } \overline{x} \to \otimes y \\ \varnothing & \text{otherwise} \end{cases} \tag{4}$$

$$\tilde{\otimes} x \wedge \tilde{\otimes} y \rightleftarrows \begin{cases} \tilde{\otimes} x & \text{if } \tilde{\otimes} x = \tilde{\otimes} y \\ \varnothing & \text{else} \end{cases} \tag{5}$$

$$\otimes x^c = \{x \in X^c | x < \underline{x}, \overline{x} < x\} \tag{6}$$

$$\otimes x \oplus \otimes y \rightleftarrows \begin{cases} (\otimes x^c \wedge \otimes y^c)^c & \text{if } \otimes x \wedge \otimes y \rightleftarrows \varnothing \\ (\otimes x \vee \otimes y) \wedge (\otimes x \wedge \otimes y)^c & \text{if } \otimes x \wedge \otimes y \not\rightleftarrows \varnothing \end{cases} \tag{7}$$

where '$\otimes x \to \otimes y$' denotes $\underline{y} \le \underline{x}$ and $\overline{x} \le \overline{y}$ called the grey lattice inclusion relation.

Whitening functions [17, 16] which compute a whitened value from a grey number are given as follows:

Midpoint $\mathrm{mid}(\otimes x) = (\underline{x} + \overline{x})/2$ **Size** $\mathrm{size}(\otimes x) = (|\underline{x}| + |\overline{x}|)/2$
Diameter $\mathrm{dia}(\otimes x) = \overline{x} - \underline{x}$ **Radius** $\mathrm{rad}(\otimes x) = (\overline{x} - \underline{x})/2$
Magnitude $\mathrm{mag}(\otimes x) = \max(|\underline{x}|, |\overline{x}|)$ **Mignitude** $\mathrm{mig}(\otimes x) = \min(|\underline{x}|, |\overline{x}|)$
Sign $\mathrm{sign}(\otimes x) = \begin{cases} 1 & \text{if } 0 < \underline{x} \\ 0 & \text{if } 0 \to \otimes x \\ -1 & \text{if } \overline{x} < 0 \end{cases}$ **Heaviside** $\mathrm{hv}(\otimes x) = \begin{cases} 1 & \text{if } 0 \le \underline{x} \\ 0 & \text{if } \overline{x} < 0 \\ \text{Unknown} & \text{if } 0 \to \otimes x \end{cases}$
Absolute $\mathrm{abs}(\otimes x) = \mathrm{mag}(\otimes x) - \mathrm{mig}(\otimes x)$
Pivot $\mathrm{piv}(\otimes x) = \sqrt{\mathrm{mag}(\otimes x) \cdot \mathrm{mig}(\otimes x)}$
Overlap $ov(\otimes x, \otimes y) = \frac{\mathrm{dia}(\otimes x \wedge \otimes y)}{\mathrm{dia}(\otimes x \vee \otimes y)}$
where $\otimes x \wedge \otimes y \rightleftarrows \varnothing \Leftrightarrow ov(\otimes x, \otimes y) = 0$; $\otimes x \rightleftarrows \otimes y \Leftrightarrow ov(\otimes x, \otimes y) = 1$

Especially the meet operation, the diameter and the overlap are mainly used to make a discernibility matrix of grey-rough reduction in this paper.

Example 1. For grey numbers $\otimes x_1 \rightleftarrows [0.1, 0.4]$, $\otimes x_2 \rightleftarrows [0.7, 1.1]$, $\otimes x_3 \rightleftarrows [1.2, 1.7]$ and $\otimes x_4 \rightleftarrows [0.5, 1.3]$ corresponding to objects x_1, x_2, x_3 and x_4, we have

a	b	$a \vee b$	$a \wedge b$	$\mathrm{dia}(a \vee b)$	$\mathrm{dia}(a \wedge b)$	$ov(a, b)$
$\otimes x_1$	$\otimes x_2$	$[0.1, 1.1]$	$\varnothing$	1.0	0	0
$\otimes x_1$	$\otimes x_3$	$[0.1, 1.7]$	$\varnothing$	1.6	0	0
$\otimes x_1$	$\otimes x_4$	$[0.1, 1.3]$	$\varnothing$	1.2	0	0
$\otimes x_2$	$\otimes x_3$	$[0.7, 1.7]$	$\varnothing$	1.0	0	0
$\otimes x_2$	$\otimes x_4$	$[0.5, 1.3]$	$[0.7, 1.1]$	0.8	0.4	0.5
$\otimes x_3$	$\otimes x_4$	$[0.5, 1.7]$	$[1.2, 1.3]$	1.2	0.1	0.0833

2.2 Grey-Rough Approximation

Let $IS = (U, A, V, f_\otimes)$ denote an information system called a grey information system [18], where

- U: a set of objects called the universe
- A: a set of attributes (conditional attributes)
- V: a set of values, $V \subseteq \mathbb{R}$ in this paper
- $f_\otimes$: the information function as $f_\otimes : U \times A \longrightarrow V$

A grey-rough approximation [16] for a grey information system IS is based on the meet operation and the grey lattice inclusion ($\rightarrow$). Let x be an object of U, a be an attribute of A and $f_\otimes(x, a) \rightleftarrows [\underline{f_\otimes}(x, a), \overline{f_\otimes}(x, a)]$ be a value which x holds on the attribute a, where an ordered pair $(x, a) \in U \times A$, $\underline{f_\otimes}(x, a) = \inf V_a$ and $\overline{f_\otimes}(x, a) = \sup V_a$. Let $f_\otimes(s, a) \rightleftarrows [\underline{f_\otimes}(s, a), \overline{f_\otimes}(s, a)]$ be a value on a called an objective of approximation; the upper approximation $GL^*(f_\otimes(s, a))$ and lower approximation $GL_*(f_\otimes(s, a))$ are given as follows:

$$GL^*(f_\otimes(s, a)) = \{x \in U | f_\otimes(x, a) \wedge f_\otimes(s, a) \not\rightleftarrows \varnothing\} \tag{8}$$

$$GL_*(f_\otimes(s, a)) = \{x \in U | f_\otimes(x, a) \rightarrow f_\otimes(s, a)\} \tag{9}$$

$GL(f_\otimes(s, a))$ is a single-attribute approximation on an attribute a of A.

A multi-attribute approximation is also given. Let $A = \{a_1, a_2, \cdots, a_n\}$ be a set of n attributes, $S = \{f_\otimes(s, a_1), f_\otimes(s, a_2), \cdots, f_\otimes(s, a_n)\}$ be a set of n values on attributes of A denoting an objective. The upper approximation $GW^*(S)$ and lower approximation $GW_*(S)$ are given as follows:

$$GW^*(S) \rightleftarrows [\underline{GW}^*(S), \overline{GW}^*(S)] \tag{10}$$

$$GW_*(S) \rightleftarrows [\underline{GW}_*(S), \overline{GW}_*(S)] \tag{11}$$

$$\underline{GW}^*(S) = \textstyle\bigcap_{k=1}^{n} GL^*(f_\otimes(s, a_k)) \tag{12}$$

$$\overline{GW}^*(S) = \textstyle\bigcup_{k=1}^{n} GL^*(f_\otimes(s, a_k)) \tag{13}$$

$$\underline{GW}_*(S) = \textstyle\bigcap_{k=1}^{n} GL_*(f_\otimes(s, a_k)) \tag{14}$$

$$\overline{GW}_*(S) = \textstyle\bigcup_{k=1}^{n} GL_*(f_\otimes(s, a_k)) \tag{15}$$

A pair of interval sets $\langle GW^*(S), GW_*(S)\rangle$ is a multi-attribute grey-rough set. The multi-attribute approximation is mainly used in approximation [16]. In this paper, the single-attribute approximation is mainly used for reduction.

In the classical model, the positive region $POS(X) = R_*(X)$, the upper region $UPP(X) = R^*(X)$, the negative region $NEG(X) = U - R^*(X)$ and the boundary region $BND(X) = R^*(X) - R_*(X)$ are given. It has been shown that the grey-rough approximation is compatible with Pawlak's classical rough set model [16]. These regions for grey-rough sets are newly definable.

Definition 1. The positive region POS, the upper region UPP, the negative region NEG and the boundary region BND of $GL(f_\otimes(s, a))$ are given as follows:

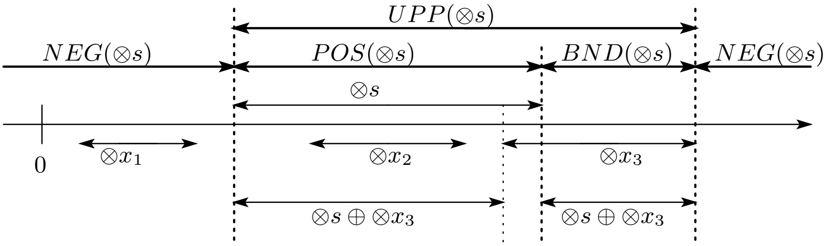


Fig. 1. A grey-rough approximation: an illustration of Example 2

$$POS(f_\otimes(s,a)) = GL_*(f_\otimes(s,a)) \tag{16}$$

$$UPP(f_\otimes(s,a)) = GL^*(f_\otimes(s,a)) \tag{17}$$

$$NEG(f_\otimes(s,a)) = U - GL^*(f_\otimes(s,a)) \tag{18}$$

$$BND(f_\otimes(s,a)) = GL^*(f_\otimes(s,a)) - GL_*(f_\otimes(s,a)) \tag{19}$$

Example 2. For the data of Example 1, redefine $\otimes x_4$ a grey number $\otimes s \rightleftarrows [0.5, 1.3]$ corresponding to the objective s. In this example, $U = \{x_1, x_2, x_3, s\}$ is given. In this grey information system, $GL_*(\otimes s) = \{x_2, s\}$ and $GL^*(\otimes s) = \{x_2, x_3, s\}$ are given by the grey-rough approximation of s. Therefore, $POS(\otimes s) = \{x_2, s\}$, $UPP(\otimes s) = \{x_2, x_3, s\}$, $NEG(\otimes s) = \{x_1\}$ and $BND(\otimes s) = \{x_3\}$ are given. This example is illustrated in Fig. 1.

3 Grey-Rough Reduction

3.1 Principle

Let $IS = (U, A \cup D, V, f_\otimes)$ be a decision table called a grey decision table [18], where D is a set of decision (class) attributes. Decision is often given one attribute $d \in D$; $f_\otimes(x, d) \in V_d$ given for an ordered pair $(x, d) \in U \times D$ is a decision (class) value.

The classical reduction approach is based on an indiscernibility relation $xI(B)y$ (an equivalence relation [9]), if and only if $a(x) = a(y)$ for all attributes $a \in B \subseteq A$, where $a(x)$ is the value for object x on a. The proposal reduction is also based on a discernibility matrix. An indiscernibility relation is expanded for interval data that is based on the equivalence class $[x]_{GR} = \{y \in U | f_\otimes(x,a) \rightleftarrows f_\otimes(y,a), \forall a \in B \subseteq A\}$ in grey-rough sets [16]. According to Example 2, it is shown that the object x_1 of NEG is discernible to the objective s. The object x_3 of BND is conditionally discernible each other: s and x_3 are discernible with the exclusive-joined region $\otimes s \oplus \otimes x_3$ as shown in Fig. 1. The object x_2 of POS is also conditionally discernible, however, almost indiscernible, because $\otimes x_2$ is included by $\otimes s$. A basic reduction approach is proposed for NEG and an advanced approach is also proposed for UPP (BND and POS) with a parameter.

3.2 Basic Reduction Approach

Definition 2. Let $xIGR(B)y$ be an indiscernibility relation in grey rough sets, if and only if $f_{\otimes}(x,a) \wedge f_{\otimes}(y,a) \not\rightleftarrows \varnothing$, $\forall a \in B \subseteq A$.

Definition 3. A discernibility matrix for interval data is given as follows:

$$\delta_{ij} = \{a \in A | f_{\otimes}(x_i,a) \wedge f_{\otimes}(x_j,a) \rightleftarrows \varnothing; f_{\otimes}(x_i,d) \neq f_{\otimes}(x_j,d)\} \tag{20}$$

where both x_i and $x_j \in U$; $1 \leq i,j \leq \text{card}(U)$.

Definition 4. A discernibility function $\mathcal{F}_{IS}$ is given as follows:

$$\mathcal{F}_{IS}(A^*) = \bigwedge_{\exists \delta_{ij}(\delta_{ij} \neq \varnothing)} \left(\bigvee_{\forall a \in \delta_{ij}} a \right) \tag{21}$$

where $A^* = \bigcup_{\forall i,j} \delta_{ij}$ is a set of conditional attributes given as $A^* \subseteq A$.

The indiscernibility relation $xIGR(B)y$ denotes that the grey number interval $f_{\otimes}(x,a)$ overlaps another grey number interval $f_{\otimes}(y,a)$, which is expanded from $a(x) \neq a(y)$ of the classical model. The function shown in Eq. (21) is a Boolean function written a conjunctive form; the reduction procedure transforms this function into a disjunctive form. The attributes of A^* are candidates for a reduct, the result of reduction. The *core* [8], the intersection of all terms denoting a set of key attributes of a reduct, follow the classical approach in this paper. When the transformation is run on the computers, the elements δ_{ij} $(1 \leq i < j \leq \text{card}(U))$ are used for fast computation.

Example 3. Assume that each object $\{x_1, x_2, x_3, x_4\} \in U$ of Example 1 holds a unique class in one attribute. According to Eq. (20), the ordered pairs except (x_2, x_4) and (x_3, x_4) are discernible each other (see the column '$a \wedge b$' of the table in Example 1).

3.3 Advanced Reduction Approach

The proposed discernibility matrix can be expanded with a parameter.

Definition 5. The discernibility matrix with a parameter for interval data in a grey decision table is given as follows:

$$\delta_{ij} = \begin{cases} \{a \in A | f_{\otimes}(x_i,a) \wedge f_{\otimes}(x_j,a) \rightleftarrows \varnothing; f_{\otimes}(x_i,d) \neq f_{\otimes}(x_j,d)\} \\ \hfill \text{if } f_{\otimes}(x_i,a) \text{ or } f_{\otimes}(x_j,a) \text{ is } \tilde{\otimes} \\ \{a \in A | ov(f_{\otimes}(x_i,a), f_{\otimes}(x_j,a)) \leq p; f_{\otimes}(x_i,d) \neq f_{\otimes}(x_j,d)\} \\ \hfill \text{if and only if both } f_{\otimes}(x_i,a) \text{ and } f_{\otimes}(x_j,a) \text{ are } \otimes \end{cases} \tag{22}$$

where $0 \leq p \leq 1$ is a real parameter to adjust the elements of the matrix.

The first form of Eq. (22) is applied two objects for at least one object holds a non-interval value; for example, $3.0 \wedge 3.0 \not\rightleftarrows \varnothing$ indiscernible, $3.0 \wedge 2.8 \rightleftarrows \varnothing$ discernible, $2.8 \wedge [2.3, 3.4] \rightleftarrows [2.8, 2.8]$ indiscernible and $3.7 \wedge [2.3, 3.4] \rightleftarrows \varnothing$ discernible.

The second form is applied two objects both holding an interval value. The advanced approach implies possible discernment; the parameter p is introduced to adjust its possibility. The overlap function measures the degree of possibility of discernment between two objects. If $p = 0$ is given, the advanced approach becomes almost the basic approach but not equal and if $p = 1$ is given, the condition is unlimited. Compared with the overlap function, the standard inclusion relation $v_{\mathrm{SRI}}(X, Y)$ in the classical approach [20] is

$$v_{\mathrm{SRI}}(X, Y) = \begin{cases} \frac{\mathrm{card}(X \cap Y)}{\mathrm{card}(X)} & \text{if } X \neq \varnothing \\ 1 & \text{otherwise} \end{cases} \tag{23}$$

which is based on the cardinal of two sets, not of two objects.

Example 4. Assume that each object $\{x_1, x_2, x_3, x_4\} \in U$ of Example 1 holds a unique class in one attribute. According to Eq. (22), the ordered pairs except (x_2, x_4) and (x_3, x_4) are discernible each other for $0 \leq p < 0.0833$. Only the pair (x_2, x_4) is indiscernible for $0.0833 \leq p < 0.5$ and all pairs become discernible for $0.5 \leq p < 1$ (see the column '$ov(a, b)$' of the table in Example 1).

4 Experiments

4.1 Compatibility with the Classical Approach

It has been already shown in Ref. [16] that the grey-rough approximation is compatible with Pawlak's classical rough approximation to replace given values, for example, 'yes $\Leftrightarrow 1 \rightleftarrows [1, 1]$' and 'no $\Leftrightarrow 2 \rightleftarrows [2, 2]$'. This paper reports that the proposal reduction is also compatible with the classical reduction approach for non-interval data sets as shown in Table 1: 11 data sets including no missing values have been picked from UCI repository [21]. In this research, Rough Set Exploration System (RSES[1]) [22] has been used to confirm compatibility[2]. Both RSES and the proposal have provided the same results for each data set.

4.2 Practical Example of the Basic Approach for Interval Data

Table 2 [18] shows a grey decision table for suppliers selection problem, one of the decision-making problems. Decision makers are often unable to determine their judgment exactly, and then estimated judgment data include uncertainty. In this example, the grey decision table IS is given as follows: $U = \{x_i; i = 1, 2, \cdots, 7\}$ a set of objects called alternatives, $A = \{a_k; k = 1, 2, \cdots, 4\}$ a set of conditional

[1] RSES Version 2.2.2, Exhaustive algorithm, Full discernibly with Modulo decision.

[2] The grey-rough reduction program is supported by MATLAB 7.1.

Table 1. Reduction results of the two models for non-interval data sets

Data set name	Nr of objects	Nr of condition attributes	Nr of classes	Nr of reducts by RSES	Nr of reducts by the proposal
Australian	690	14	2	44	44
Balance-Scale	625	4	3	1	1
Balloons	16	4	2	1	1
Flare	323	12	6	1	1
German	1000	20	2	846	846
Heart-disease	270	13	2	109	109
Lenses	24	4	3	1	1
Lymphography	148	18	4	424	424
Pima-Indian-diabetes	768	8	2	28	28
Tic-Tac-Toe	958	9	2	9	9
Zoo	101	17	7	34	34

Table 2. A grey decision table for suppliers selection problem

	a_1	a_2	a_3	a_4	d
x_1	[0.697,1.000]	[0.733,1.000]	[0.667,0.909]	[0.656,0.955]	yes
x_2	[0.606,0.788]	[0.733,1.000]	[0.697,1.000]	[0.724,1.000]	yes
x_3	[0.576,0.758]	[0.433,0.600]	[0.636,0.879]	[0.553,0.700]	yes or no
x_4	[0.545,0.667]	[0.267,0.467]	[0.667,0.909]	[0.636,0.913]	yes
x_5	[0.545,0.667]	[0.333,0.500]	[0.545,0.545]	[0.778,1.000]	yes or no
x_6	[0.636,0.818]	[0.267,0.467]	[0.515,0.636]	[0.553,0.700]	no
x_7	[0.545,0.667]	[0.267,0.467]	[0.667,0.909]	[0.636,0.913]	yes or no

Li et al. A grey-based rough set approach to suppliers selection problem. Proc. RSCTC2006, LNAI4259 (2006) pp.487–496.

attributes, product quality (a_1), service quality (a_2), delivery time (a_3) and price (a_4). A value $f_\otimes(x_i, a_k)$ becomes an estimated judgment given as an interval: the best evaluation in each upper endpoint and worst evaluation in each lower endpoint, respectively. The decision $f_\otimes(x_i, d)$ denotes the total judgment, where $V_d = \{\text{yes, no, yes or no}\}$. The grey-rough reduction investigates which attribute is effective to select the best ideal supplier.

The discernibility matrix of the basic approach is given as follows:

$$\begin{array}{c|ccccccc}
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\
\hline
x_1 & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
x_2 & \star & \ast & \ast & \ast & \ast & \ast & \ast \\
x_3 & \{a_2\} & \{a_2, a_4\} & \ast & \ast & \ast & \ast & \ast \\
x_4 & \star & \star & \varnothing & \ast & \ast & \ast & \ast \\
x_5 & \{a_1, a_2, a_3\} & \{a_2, a_3\} & \star & \{a_3\} & \ast & \ast & \ast \\
x_6 & \{a_2, a_3\} & \{a_2, a_3, a_4\} & \varnothing & \{a_3\} & \{a_4\} & \ast & \ast \\
x_7 & \{a_1, a_2\} & \{a_2\} & \star & \varnothing & \star & \{a_3\} & \ast
\end{array}$$

Table 3. Reduction of the grey decision table by the advanced approach

p	Nr of reducts	Reduct(s)
0, 0.1	1	$a_2a_3a_4$
0.2	2	$a_1a_2a_3$ and $a_2a_3a_4$, core: a_2a_3
0.3	3	$a_1a_2a_3$, $a_1a_3a_4$ and $a_2a_3a_4$, core: a_3
0.4, 0.5	3	a_2a_4, $a_1a_2a_3$ and $a_1a_3a_4$
0.6, 0.7, 0.8	5	a_1a_2, a_1a_3, a_1a_4, a_2a_3 and a_2a_4
0.9	6	a_1a_2, a_1a_3, a_1a_4, a_2a_3, a_2a_4 and a_3a_4
1.0	1	$a_1a_2a_4$

where a symbol '$*$' is an omitted element for fast reduction computation and a symbol '$\star$' is also an omitted element since $f_\otimes(x_i, d) = f_\otimes(x_j, d)$. According to the matrix, the discernibility function $\mathcal{F}_{IS}$ and its reduct are given as follows:

$$\begin{aligned}\mathcal{F}_{IS} &\equiv a_2 \cdot a_3 \cdot a_4 \cdot (a_1 \vee a_2) \cdot (a_2 \vee a_3) \cdot (a_2 \vee a_4) \cdot (a_1 \vee a_2 \vee a_3) \cdot (a_2 \vee a_3 \vee a_4) \\ &\Leftrightarrow a_2 \cdot a_3 \cdot a_4\end{aligned}$$

therefore, service quality, delivery time and price are important to select the best supplier, in other words, only product quality is not so important in this grey decision table. Thus the basic approach is available for interval data reduction.

4.3 Property on the Advanced Approach

The aim of this example is to investigate influence of the parameter p in the advanced approach. Table 3 shows the result of the advanced approach for the same grey decision table of Table 2, where the parameter $0 \le p \le 1.0$ in units of 0.1. The equal reduct has been given both in the basic approach and the advanced approach at $p = 0$ and $p = 0.1$. A number of reducts has increased as the parameter increases, which implies the advanced approach has done possible reduction. At $p = 1.0$, only a_3 has been out of the reduct though the condition is unlimited, because the value $f_\otimes(x_5, a_3) = 0.545$ is a non-interval value and then the first form of Eq. (22) has been applied to a_3 only in this example. When $p = 0.2$ and $p = 0.3$ the cores $a_2 \cdot a_3$ and a_3 have been given, respectively. According to these results, a_3 (delivery time) seems to be a key attribute in that grey decision table. Thus the advanced approach is available for interval data reduction under complex decision tables in data mining.

5 Conclusion

A new grey-rough reduction of attributes for interval data was proposed in this paper. Two sorts of grey-rough reduction model, a basic approach and advanced approach were proposed with several illustrative examples. Conclude this paper with advantages of the proposal as follows:

- The grey-rough reduction is compatible with the classical reduction approaches for data sets of non-interval data.

- The grey-rough reduction is useful for decision tables containing both non-interval and interval data complexly.
- The advanced approach carries out a possible reduction with a parameter, in which an attribute containing both non-interval values and interval values is distinguished in the practical example.

The proposal is needed to investigate more effectiveness, for example, comparison with related models such as Dembczynski' model [23].

References

1. Diday, E., Esposito, F.: An Introduction to Symbolic Data Analysis and the SODAS software. Intelligent Data Analysis. 7, 583–601 (2003)
2. Limam, M.M., Diday, E., Winsberg, S.: Symbolic Class Description with Interval Data. The Electronic Journal of Symbolic Data Analysis. 1, 1–10 (2003)
3. Mballo, C., Diday, E.: Decision trees on interval valued variables. The Electronic Journal of Symbolic Data Analysis. 3, 8–18 (2005)
4. de Souza, R.M.C.R., de Carvalho, F.A.T.: Clustering of interval data based on city-block distances. Pattern Recognition Letters. 25, 353–365 (2004)
5. de Souza, R.M.C.R., de Carvalho, F.A.T.: Dynamic clustering of interval data based on adaptive Chebyshev distances. Electronics Letters. 40, 658–660 (2004)
6. Asharaf, S., Murty, M.N., Shevade, S.K.: Rough set based incremental clustering of interval data. Pattern Recognition Letters. 27, 515–519 (2006)
7. Pawlak, Z., Skowron, A.: Rough sets and Boolean reasoning. Information Sciences. 177, 41–73 (2007)
8. Skowron, A.: Rough sets and boolean reasoning. In: Pedrycz, W. (ed.) Granular Computing, pp. 95–124. Physica-Verlag, Heidelberg (2001)
9. Pawlak, Z.: Rough set elements. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery 1, pp. 10–30. Physica-Verlag, Heidelberg (1998)
10. Song, X.X., Zhang, W.X.: Knowledge reduction in set-valued decision information system. Lect. Notes Artif. Intell. 4259, 348–357 (2006)
11. Sakai, H., Nakata, M.: On possible rules and apriori algorithm in non-deterministic information systems. Lect. Notes Artif. Intell. 4259, 264–273 (2006)
12. Deng, J.L.: Grey Systems, China Ocean Press (1988)
13. Nagai, M., Yamaguchi, D.: Elements on Grey System Theory and its Applications (in Japanese), Kyoritsu-Shuppan (2004)
14. Wen, K.L.: Grey Systems: Modeling and Prediction. Yang's Scientific Research Institute (2004)
15. Liu, S.F., Lin, Y.: Grey Information. Springer, Heidelberg (2006)
16. Yamaguchi, D., Li, G.D., Nagai, M.: On the combination of rough set theory and grey theory based on grey lattice operations. Lect. Notes Artif. Intell. 4259, 507–516 (2006)
17. Yamaguchi, D., Li, G.D., Mizutani, K., Akabane, T., Nagai, M., Kitaoka, M.: On the Generalization of Grey Relational Analysis. Journal of Grey System. 9, 23–34 (2006)
18. Li, G.D., Yamaguchi, D., Lin, H.S., Wen, K.L., Nagai, M.: A grey-based rough set approach to suppliers selection problem. Lect. Notes Artif. Intell. 4259, 487–496 (2006)

19. Yamaguchi, D., Li, G.D., Mizutani, K., Nagai, M., Kitaoka, M.: Decision Rule Extraction and Reduction Based on Grey Lattice Classification. In: Proc. 4th Int. Conf. Machine Learning and Applications (ICMLA'05), pp. 31–36 (2005)
20. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences. 177, 28–40 (2007)
21. Merz, C.J., Murphy, P.M.: UCI Repository of machine learning databases: http://www.ics.uci.edu/~mlearn/MLRepository.html
22. Bazan, J.G., Szczuka, M.: RSES and RSESlib - a collection of tools for rough set computations. Lect. Notes Artif. Intell. 2005, 106–113 (2001)
23. Dembczynski, K., Greco, S., Slowinski, R.: Second-order rough approximations in multi-criteria classification with imprecise evaluations and assignments. Lecture Notes in Computer Science 3641, 54–63 (2005)

Minimal Templates and Knowledge Discovery

Barbara Marszał–Paszek and Piotr Paszek

Institute of Computer Science, University of Silesia
Będzińska 39, 41–200 Sosnowiec, Poland
paszek@us.edu.pl

Abstract. In this paper the dependences between the Dempster-Shafer theory and rough set theory have been used to find a minimal template in a given decision table. The Dempster-Shafer theory [5] is called a mathematical theory of evidence. This theory is based on belief functions and plausible reasoning is used to combine separate pieces of information (evidence) to calculate the probability of an event. Rough set theory was proposed by Pawlak in 1982 [3] as a mathematical tool for describing the uncertain knowledge. In 1987 [1] and 1991 [6] the basic functions of the evidence theory were defined, based on the notation from rough set theory. These definitions allow finding interesting dependences in decision tables.

1 Introduction

1.1 Preliminaries of Rough Set Theory

In rough set theory knowledge is based on possibility (capability) for classifying objects. The objects can be for instance real objects, statements, abstract concepts or processes. Let us recall some basic definitions of rough set theory [4].

A pair $\mathbb{A} = (U, A)$ will be called an *information system*, where U – is a non-empty, finite set called the *universe* and A – is a non-empty, finite set of *attributes.* Each attribute $a \in A$ is a function $a : U \to V_a$, where V_a – is called the value set of a.

Let $\mathbb{A}$ be an information system and let $A = C \cup D$ where C, D are non-empty, disjoint subsets of A. The set C is called the set of *condition* attributes and the set D is called the set of *decision* attributes. The triple $\mathbb{A} = (U, A, C, D)$ is referred to as a *decision table.* A simplified version of a decision table has a form $\mathbb{A} = (U, A \cup \{d\})$, where $d \notin A$. Now, the set of decision attributes D is limited to one decision attribute only. The decision d creates a partition of the universe U into decision classes $X_1, \ldots, X_{r(d)}$, where $r(d) = |\{k : \exists_{x \in U} : d(x) = k\}|$ is the number of different values of the decision attribute called the *rank* of the decision d.

1.2 Belief and Plausibility Functions in Rough Sets

In 1987 [1] and 1991 [6] Grzymała–Busse and Skowron suggested a clear way to connect rough sets theory and the evidence theory. They defined basic functions

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 411–416, 2007.

of the evidence theory based on the concepts of rough set theory. In this section we recall some basic definitions that are indispensable for further considerations.

Let $\Theta_A = \{1, 2, ..., r(d)\}$ be the frame of discernment defined by the decision d in the decision table $\mathbb{A}$.

For any $\theta \in \Theta_A$ the following equality holds:

$$Bel_A(\theta) = \frac{\left|\underline{A} \bigcup\limits_{i\in\theta} X_i\right|}{|U|}. \tag{1}$$

The equality above defines the relationship between the *belief function* $Bel_A(\theta)$ and the lower approximation of a set from rough set theory. The belief function is the ratio of the number of objects in U that can be certainly classified to the union $\bigcup\limits_{i\in\theta} X_i$ to the number of all objects in U.

Also for any $\theta \in \Theta_A$ the following equality holds:

$$Pl_A(\theta) = \frac{\left|\overline{A} \bigcup\limits_{i\in\theta} X_i\right|}{|U|}. \tag{2}$$

The equality above defines relationship between the *plausibility function* $Pl_A(\theta)$ and the upper approximation of a set from rough set theory. The plausibility function is the ratio of the number of objects that can be probably classified to the union $\bigcup\limits_{i\in\theta} X_i$ to the number of all objects in U.

1.3 Templates in a Decision Table

A *template* T [2] in a decision table is any sequence $v_1, \ldots, v_n$, where $v_i \in V_{a_i} \cup \{*\}$.

The symbol $'*'$ appearing in a given template means that the value of the marked attribute is not restricted by the template. Alternatively, a template can be defined as the conjunction of a certain number of the descriptors e.g.

$$T = (c = 0) \wedge (e = 1) \wedge (f = 1).$$

A given object matches a given template if $a_i(x) = v_i$, for each i such that $v_i \neq' *'$.

For a given template T the following notions are defined:

- *length*$(T) := |v_i \in T : v_i \neq' *'|$;
- *support*$(T) := |x \in U : \forall_{v_i \in T, v_i \neq' *'} x(a_i) = v_i|$.

2 Minimal Templates Problem

Let $\mathbb{A}$ be a decision table. By $\mathbb{A}_T = (U_T, A \cup d)$ we denote the restriction of $\mathbb{A}$ to a template T, i.e. $U_T = \{x \in U : x(a_i) = v_i, v_i \neq' *'\}$ for all $i \in \{1, \ldots, n\}$ and $A(T) = \{a_i \in A : v_i \neq' *' \in T\}$ is the set of attributes of $\mathbb{A}$ restricted to U_T.

We consider the following problem:

Minimal Templates Problem

Input:
A decision table $\mathbb{A}$; thresholds $\varepsilon_1, \varepsilon_2 \in (0,1)$ and a natural number $1 \leq k < r(d)$.

Output:
Minimal (with respect to the length) templates T for which there exists a set $\theta \subseteq \Theta_{\mathbb{A}_T}$ with at most k elements ($|\theta| \leq k$) satisfying the following conditions:

$$|Pl_{\mathbb{A}_T}(\theta) - Bel_{\mathbb{A}_T}(\theta)| < \varepsilon_1 \quad for \quad \varepsilon_1 \in (0,1); \quad \theta \subseteq \Theta_{A_T}; \tag{3}$$

$$|Pl_{\mathbb{A}_T}(\theta)| > 1 - \varepsilon_1 \quad for \quad \varepsilon_1 \in (0,1); \quad \theta \subseteq \Theta_{A_T}; \tag{4}$$

$$\frac{|U_T|}{|U|} > \varepsilon_2 \quad for \quad \varepsilon_2 \in (0,1). \tag{5}$$

Based on the conditions that direct the searching process of the minimal templates the following rule is obtained:

$$T \Rightarrow \theta.$$

The conditional part of this rule is a template and the decision part is a set θ. Such rules can be interesting in the case where there are no strong rules (with the right hand side described by a single decision value) in a given decision table that have satisfactory support. Then we search for rules that have a sufficiently large support with respect to the minimal set θ of decision values.

3 Problem Solution

Since the problem of the construction of templates that satisfy conditions (3–5) is *NP-hard* we use a genetic algorithm.

The starting point is a set of decision reducts. Each reduct is a set of the attributes. Among these attributes we are looking for the templates that are solution of the problem.

```
begin
  Decision reduct generation - R
  forall r in R do
     SGA(r); - Genetic algorithm
  endfor
end;
```

Genetic algorithm starts from a random population of objects. Every object has the same length as the reduct. Each object defines the set of the templates. We reject templates that do not satisfy condition (5). For each template we calculate corresponding θ.

```
Procedure SGA(r);
    begin
      nr:=0; - population number
      Generation(Pop(nr));
      RuleGeneration(Pop(nr));
      Fitt(Pop(nr));
      while (not STOP) do
         nr := nr + 1;
         Selection Pop(nr) from Pop(nr-1);
         Mutation Pop(nr);
         RuleGeneration(Pop(nr));
         Fitt(Pop(nr));
      endwhile;
      Solution is the best result for the last population
    end;
```

where

```
Procedure RuleGeneration(Pop(i));
begin
  forall t in Pop(i) do
    TemplatesGeneration(t);
     forall T in TemplatesGeneration(t)
        ThetaAlg(T);
     endfor;
  endfor;
end;
```

In the genetic algorithm, for each template a heuristic is used to find $\theta \in \Theta_T$. θ must satisfy both conditions (3) and (4).

The *fitness function* in the genetic algorithm must take into account the aim which is to find the minimal templates. So this function must reward the objects that have the smallest number of attributes. On the other hand the shortest template must satisfy condition (3), (4) and (5). We consider the following *fitness function*:

$$Fitt(x) = \frac{a}{Templ\_Lg} + \frac{b}{Av\_Gl} + c * \frac{Av\_Supp}{|U|} + Const$$

where $Templ\_Lg$ is the length of the template, Av_Gl is the average number of decision class gluing, Av_Supp is the average of the template support for the population, and a, b, c are non-negative real numbers such that:

$$a + b + c = 1$$

If it is impossible to find templates together with θ, the fitness function is defined in the following way:

$$Fitt(x) = 0 + const.$$

4 Results

For example, *Wisconsin Breast Cancer Database* was taken. This database has 699 objects, 9 condition attributes and 1 decision reduct. The presented algorithm for solving the minimal template problem generates the specific output file. For parameters $\varepsilon_1 = 0.1$ and $\varepsilon_2 = 0.1$ the following results were obtained:

```
4  1    8  13
0
4  130  5    1  3
8  108  3    1  3  7  4  8  2
3   80  4    1
6  145  1    1  2   7   8
```

line 1: 4 is the number of templates, 1 is the number of descriptors, 8 is the length of objects in the population, 13 is the average number of different values of the decision attribute in a template,

line 2: 0 is the number of the attribute,

line 3: 4 is the number of values that appear in this line, 130 is the support of the template, 5 is the value of attribute with the number from line 2, 1 and 3 are the numbers of decision class which are glued.

It means that the following rules were received:

- $(a_0 = 5) \Rightarrow (d = 1 \vee d = 3)$ with support = 130;
- $(a_0 = 3) \Rightarrow (d = 1 \vee d = 2 \vee d = 3 \vee d = 4 \vee d = 7 \vee d = 8)$ with support = 108;
- $(a_0 = 4) \Rightarrow (d = 1)$ with support = 80;
- $(a_0 = 1) \Rightarrow (d = 1 \vee d = 2 \vee d = 7 \vee d = 8)$ with support = 145.

For parameters $\varepsilon_1 = 0.1$ and $\varepsilon_2 = 0.05$ the following results were obtained:

```
7  1    8  25
0
4  130  5    1  3
8  108  3    1  3  7   4  8  2
3   80  4    1
6  145  1    1  2  7   8
7   46  8    1  2  8   4  7
4   50  2    1  5
7   69  10   1  2  3  10  8
```

The following rules were received:

- $(a_0 = 5) \Rightarrow (d = 1 \vee d = 3)$ with support = 130;
- $(a_0 = 3) \Rightarrow (d = 1 \vee d = 2 \vee d = 3 \vee d = 4 \vee d = 7 \vee d = 8)$ with support = 108;

- $(a_0 = 4) \Rightarrow (d = 1)$ with support = 80;
- $(a_0 = 1) \Rightarrow (d = 1 \vee d = 2 \vee d = 7 \vee d = 8)$ with support = 145;
- $(a_0 = 8) \Rightarrow (d = 1 \vee d = 2 \vee d = 4 \vee d = 7 \vee d = 8)$ with support = 46;
- $(a_0 = 2) \Rightarrow (d = 1 \vee d = 5)$ with support = 50;
- $(a_0 = 10) \Rightarrow (d = 1 \vee d = 2 \vee d = 3 \vee d = 8 \vee d = 10)$ with support = 69.

From the example it results that the algorithm generates rules with the large support. Besides the user decides what support of rules is sufficient. The user can also decide about the maximum number of decision classes, which are glued.

5 Summary

The paper demonstrates that the relationships between rough set theory and the evidence theory can be used to find the minimal templates for a given decision table.

Extracting the templates from data is a problem that consists in finding the set of attributes with a minimal number of attributes, which warrants, among others, the sufficiently small difference between the belief and plausibility functions. Moreover, the minimal templates problem gives the hint which decision values can be glued. Finally we get decision rules with the sufficiently large support.

Other heuristics for searching θ could possible bring better results. Therefore, they should be analyzed in further research.

References

1. Grzymała-Busse, J.: Rough-set and Dempster-Shafer Approaches to Knowledge Acqusition uder Uncertainty - a Comparison. Technical report, Department of Computer Science University of Kansas (1987)
2. Nguyen, S.H., Skowron, A., Synak, P., Wróblewski, J.: Knowledge Discovery in Databases: Rough Set Approach. In: Proc. of The Seventh International Fuzzy Systems Association World Congress, IFSA', Prague, Czech Republic, June 1997, pp. 204–209 (1997)
3. Pawlak, Z.: Rough Sets. Int. J. of Information and Computer Sci. 11, 344–356 (1982)
4. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer, Dordecht (1991)
5. Shafer, G.: A mathematical theory of evidence. Princeton University Press, Princeton (1976)
6. Skowron, A.: Grzymała-Busse, J.: From the Rough Set Theory to the Evidence Theory. Technical report 8/91, Institute of Computer Science Warsaw University of Technology (1991)

Universal Attribute Reduction Problem

Mikhail Ju. Moshkov[1], Marcin Piliszczuk[2], and Beata Zielosko[3]

[1] Institute of Computer Science, University of Silesia
39, Będzińska St., Sosnowiec, 41-200, Poland
moshkov@us.edu.pl
[2] ING Bank Śląski S.A., 34, Sokolska St., Katowice, 40-086, Poland
marcin.piliszczuk@ingbank.pl
[3] Institute of Computer Science, University of Silesia
39, Będzińska St., Sosnowiec, 41-200, Poland
zielosko@us.edu.pl

Abstract. In the paper, some generalizations of the notions of reduct and test (superreduct) are considered. The accuracy of greedy algorithm for construction of partial test is investigated. A lower bound on the minimal cardinality of partial reducts based on an information on greedy algorithm work is studied. Results of an experiment with greedy algorithm are described.

Keywords: partial test, partial reduct, greedy algorithm.

1 Introduction

The attribute reduction problem (it is required to find a reduct with minimal or close to minimal cardinality) is one of the main problems of rough set theory [12,14,19]. There are different variants of the notion of reduct: reducts for information systems [12], usual decision and local reducts for decision tables [12,18], decision and local reducts which are based on the generalized decision [18], etc. Interesting discussion of various kinds of reducts can be found in [14], page 12.

In this paper, we consider an "universal" definition of reduct which covers at least part of possible variants. We use an approach considered in test theory [27]. Let T be a decision table and $\mathcal{P}$ be a subset of pairs of discernible rows (objects) of T. Then a reduct for T relative to $\mathcal{P}$ is a minimal (relative to inclusion) subset of conditional attributes which separate all pairs from $\mathcal{P}$. All mentioned above kinds of reducts can be represented in such a form.

In this paper, we consider not only exact but also approximate (partial) reducts which are useful in inducing data models. Rough set theory often deals with decision tables containing noisy data. In this case exact reducts can be "over-learned", i.e., depend essentially on the noise. If we view constructed reducts as a way of knowledge representation [18], then instead of large exact reducts it is more appropriate to work with relatively small partial ones. In [13] Zdzisław Pawlak wrote that "the idea of an approximate reduct can be useful in cases when a smaller number of condition attributes is preferred over accuracy of classification".

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 417–426, 2007.

Last years in rough set theory approximate reducts are studied intensively [6,7,8,9,10,15,22,23,24,25,26,29]. Approximate reducts are investigated also in extensions of rough set model such as VPRS (variable precision rough sets) [28] and α-RST (alpha rough set theory) [16].

We begin our consideration from a data table in which columns are labeled by discrete and continuous variables, and rows are tuples of values of variables on some objects. It is possible that this data table contains missing values [2,4]. We consider the following classification problem: for a discrete variable we must find its value using values of all other variables. We do not use variables directly but create some attributes with relatively small number of values based on the considered variables. As a result, we obtain a decision table with missing values in the general case. We define the universal attribute reduction problem for this table and consider a number of examples of known attribute reduction problems which can be represented as the universal one.

Based on results from [8], we obtain bounds on precision of greedy algorithm for partial test (superreduct) construction. This algorithm is a simple generalization of greedy algorithm for set cover problem [3,5,11,20,21]. We prove that under some natural assumptions on the class NP the greedy algorithm is close to the best (from the point of view of precision) polynomial approximate algorithms for minimization of cardinality of partial tests. We show that based on an information received during greedy algorithm work it is possible to obtain a nontrivial lower bound on minimal cardinality of partial reduct. We obtain also a bound on precision of greedy algorithm which does not depend on the cardinality of the set $\mathcal{P}$.

Results of experiments with randomly generated decision tables [8] and real-life decision tables from UCI repository show that using greedy algorithm we can construct short partial tests with relatively high accuracy. Results of one of such experiments are described in this paper. In particular, these results illustrate the use of the lower bound on minimal cardinality of partial reducts based on an information received during greedy algorithm work (see Theorem 7). This bound can be useful in experiments connected with the construction of various kinds of reducts by greedy algorithm.

The paper consists of five sections. In Sect. 2 a transformation of a data table into a decision table is considered. In Sect. 3 the notion of the universal attribute reduction problem is discussed. In Sect. 4 greedy algorithm for construction of partial tests (partial superreducts) is studied. In Sect. 5 results of an experiment with the decision table “kr-vs-kp” from UCI repository are described.

2 From Data Table to Decision Table

The data table D is a finite table with k columns labeled by variables $x_1, \ldots, x_k$ and N rows which are interpreted as tuples of values of variables $x_1, \ldots, x_k$ on N objects $u_1, \ldots, u_N$. It is possible that D contains missing values which are denoted by “ $-$ ”.

As usual, we assume that each of variables x_i is either discrete (with values from some finite unordered set $V(x_i)$) or continuous (with values from a set $V(x_i) \subset \mathbb{R}$). We will assume that " $-$ " does not belong to $V(x_i)$.

Let us choose a variable $x_r \in \{x_1, \dots, x_k\}$ and consider the problem of prediction of the value of x_r on a given object using only values of variables from the set $X = \{x_1, \dots, x_k\} \setminus \{x_r\}$ on the considered object. If x_r is a discrete variable, then the problem of prediction is called the classification problem. If x_r is a continuous variable, then the considered problem is called the problem of regression. We only consider the classification problem. So x_r is a discrete variable.

We only consider two kinds of missing values: (i) missing value of x_i as an additional value of variable x_i, which does not belong to $V(x_i)$, and (ii) missing value as an undefined value. In the last case, based on the value of x_i it is impossible to discern an object u_l from another object u_t if the value $x_i(u_l)$ is missed (undefined).

We now transform the data table D into a data table D^*. For each variable $x_i \in \{x_1, \dots, x_k\}$, according to the nature of x_i we choose either the first or the second way for the work with missing values. In the first case, we add to $V(x_i)$ a new value which is not equal to " $-$ ", and write this new value instead of each missing value of x_i. In the second case, we leave all missing values of x_i untouched.

To solve the considered classification problem, we do not use variables from X directly. Instead of this we use attributes constructed on the basis of these variables. Let us consider some examples.

Let $x_i \in X$ be a discrete variable. Let us divide the set $V(x_i)$ into relatively small number of nonempty disjoint subsets $V_1, \dots, V_s$. Then the value of the considered attribute on an object u is equal to the value $j \in \{1, \dots, s\}$ for which $x_i(u) \in V_j$. The value of this attribute on u is missing if and only if the value of x_i on u is missing.

Let $x_i \in X$ be a continuous variable and $c \in \mathbb{R}$. Then the value of the considered attribute on an object u is equal to 0 if $x_i(u) < c$, and is equal to 1 otherwise. The value of this attribute on u is missing if and only if the value of x_i on u is missing.

Let $x_{i_1}, \dots, x_{i_t} \in X$ be continuous variables and f be a function from $\mathbb{R}^t$ to $\mathbb{R}$. Then the value of the considered attribute on an object u is equal to 0 if $f(x_{i_1}(u), \dots, x_{i_t}(u)) < 0$, and is equal to 1 otherwise. The value of this attribute on u is missing if and only if the value of at least one variable from $\{x_{i_1}, \dots, x_{i_t}\}$ on u is missing.

We now assume that the attributes $a_1, \dots, a_m$ are chose. Let, for the definiteness, $u_1, \dots, u_n$ be all objects from $\{u_1, \dots, u_N\}$ such that the value of the variable x_r on the considered object is definite (is not missing).

We now describe a decision table T. This table contains m columns labeled by attributes $a_1, \dots, a_m$, and n rows corresponding to objects $u_1, \dots, u_n$ respectively. For $j = 1, \dots, n$ the j-th row is labeled by the value $x_r(u_j)$, which will be considered later as the value of the decision attribute d. For any $i \in \{1, \dots, m\}$

and $j \in \{1, \ldots, n\}$ the value $a_i(u_j)$ is at the intersection of the j-th row and the i-th column. If the value $a_i(u_j)$ is missing then the symbol " $-$ " is at the intersection of the j-th row and the i-th column.

3 Problem of Attribute Reduction

3.1 Definition of Problem

Let T be a decision table with m columns labeled by attributes $a_1, \ldots, a_m$ and n rows which are identified with objects $u_1, \ldots, u_n$. It is possible that T contains missing values denoted by " $-$ ". Each row is labeled by a decision which is interpreted as the value of the decision attribute d. Let $A = \{a_1, \ldots, a_m\}$ and $U = \{u_1, \ldots, u_n\}$.

We now define the indiscernibility relation $IND(T) \subseteq U \times U$. Let $u_l, u_t \in U$. Then $(u_l, u_t) \in IND(T)$ if and only if $a_i(u_l) = a_i(u_t)$ for any $a_i \in A$ such that the values $a_i(u_l)$ and $a_i(u_t)$ are definite (are not missing). Since T can contain missing values, the relation $IND(T)$ is not an equivalence relation in the general case, but it is a tolerance relation.

By $DIS(T)$ we denote the set of unordered pairs of objects u_l and u_t from U such that $(u_l, u_t) \notin IND(T)$. Let $(u_l, u_t) \in DIS(T)$ and $a_i \in A$. We will say that the attribute a_i separates the pair (u_l, u_t) if the values $a_i(u_l)$ and $a_i(u_t)$ are definite and $a_i(u_l) \neq a_i(u_t)$. For any $a_i \in A$ we denote by $DIS(T, a_i)$ the set of pairs from $DIS(T)$ which the attribute a_i separates.

Let $\mathcal{P}$ be a subset of $DIS(T)$. Let Q be a subset of A and α be a real number such that $0 \leq \alpha < 1$. We will say that Q is an α-test for T relative to $\mathcal{P}$ (an $(\alpha, \mathcal{P})$-test for T) if attributes from Q separate at least $(1 - \alpha)|\mathcal{P}|$ pairs from $\mathcal{P}$. An $(\alpha, \mathcal{P})$-test for T is called an α-reduct for T relative to $\mathcal{P}$ (an $(\alpha, \mathcal{P})$-reduct for T) if each proper subset of this $(\alpha, \mathcal{P})$-test is not an $(\alpha, \mathcal{P})$-test for T. If $\mathcal{P} = \emptyset$, then any subset Q of A is an $(\alpha, \mathcal{P})$-test for T, but only the empty set of attributes is an $(\alpha, \mathcal{P})$-reduct for T. Note that each $(\alpha, \mathcal{P})$-test contains an $(\alpha, \mathcal{P})$-reduct as a subset. The parameter α can be interpreted as inaccuracy. If $\alpha = 0$, then we obtain the notion of exact test for T relative to $\mathcal{P}$ and the notion of exact reduct for T relative to $\mathcal{P}$.

The problem of attribute reduction is the following: for given decision table T, subset $\mathcal{P}$ of the set $DIS(T)$ and real α, $0 \leq \alpha < 1$, it is required to find an $(\alpha, \mathcal{P})$-reduct for T (an $(\alpha, \mathcal{P})$-test for T) with minimal cardinality. Let us denote by $R_{\min}(\alpha) = R_{\min}(\alpha, \mathcal{P}, T)$ the minimal cardinality of an $(\alpha, \mathcal{P})$-reduct for T. Of course, it is possible to use another measures of reduct quality.

The considered problem can be easily reformulated as a set cover problem: we must cover the set $\mathcal{P}$ using minimal number of subsets from the family $\{\mathcal{P} \cap DIS(T, a_1), \ldots, \mathcal{P} \cap DIS(T, a_m)\}$. Therefore, we can use results, obtained for the set cover problem, for analysis of the attribute reduction problem.

3.2 Examples

We now consider examples of sets $\mathcal{P}$ corresponding to different kinds of reducts. It was impossible for us to find definitions of some kinds of reducts which are applicable to decision tables with missing values. In such cases we have extended existing definitions (if it was possible) trying to preserve their spirit.

For an arbitrary $u_l \in U$, let $[u_l]_T = \{u_t : u_t \in U, (u_l, u_t) \in IND(T)\}$ and $\partial_T(u_l) = \{d(u_t) : u_t \in [u_l]_T\}$. The set $\partial_T(u_l)$ is called the generalized decision for u_l. The positive region $POS(T)$ for T is the set of objects $u_l \in U$ such that $|\partial_T(u_l)| = 1$. The set $BN(T) = U \setminus POS(T)$ is called the boundary region for T.

1. *Reducts for the information system, obtained from T by removing the decision attribute d.* The set $\mathcal{P}$ is equal to $DIS(T)$ (we must preserve the indiscernibility relation).
2. *Usual decision reducts for T.* The set $\mathcal{P}$ is equal to the set of all pairs $(u_l, u_t) \in DIS(T)$ such that $d(u_l) \neq d(u_t)$ and at least one object from the pair belongs to $POS(T)$ (we must preserve the positive region).
3. *Decision reducts for T based on the generalized decision.* Let us assume T is without missing values. The set $\mathcal{P}$ is equal to the set of all pairs $(u_l, u_t) \in DIS(T)$ such that $\partial_T(u_l) \neq \partial_T(u_t)$.
4. *Maximally discerning decision reducts for T.* The set $\mathcal{P}$ is equal to the set of all pairs $(u_l, u_t) \in DIS(T)$ such that $d(u_l) \neq d(u_t)$.
5. *Usual local reducts for T and object $u_l \in POS(T)$.* The set $\mathcal{P}$ is equal to the set of all pairs $(u_l, u_t) \in DIS(T)$ such that $d(u_l) \neq d(u_t)$.
6. *Local reducts for T and object $u_l \in U$ based on the generalized decision.* Let us assume T is without missing values. The set $\mathcal{P}$ is equal to the set of all pairs $(u_l, u_t) \in DIS(T)$ such that $\partial_T(u_l) \neq \partial_T(u_t)$.
7. *Maximally discerning local reducts for T and object $u_l \in U$.* The set $\mathcal{P}$ is equal to the set of all pairs $(u_l, u_t) \in DIS(T)$ such that $d(u_l) \neq d(u_t)$.

3.3 On Maximally Discerning Reducts

The notions of maximally discerning decision and local reducts (but without the use of the term "maximally discerning") were investigated by the authors in [6,7,8,15,29]. Maximally discerning decision reducts can give us additional information on the value of the decision attribute (for example, by the separation of groups of equal rows with the same generalized decision but with different probability distributions of decision values). The consideration of maximally discerning local reducts for objects from the boundary region can lead to construction of a decision rule system which is applicable to more wide class of new objects. We now consider two examples.

Example 1. Let us consider the decision table T_1 (see Figure 1). For this table, there is exactly one usual decision reduct (which is equal to the empty set), exactly one decision reduct based on the generalized decision (which is equal to the empty set too) and exactly one maximally discerning decision reduct (which is equal to $\{a_2\}$). Based on reducts of the first two kinds it is impossible to separate the rows $(0, 0)$ from the rows $(0, 1)$. However, for the considered two types of rows we have different probability distributions of decision values. The third kind of reducts allows us to separate these two types of rows.

T_1

a_1	a_2	
0	0	1
0	0	2
0	1	1
0	1	2
0	1	2

T_2

a_1	a_2	
0	0	1
0	0	2
0	1	2
1	0	1

S_1

$a_2 = 1 \rightarrow 2$
$a_1 = 1 \rightarrow 1$

S_2

$a_1 = 0 \wedge a_2 = 0 \rightarrow \{1, 2\}$
$a_2 = 1 \rightarrow \{2\}$
$a_1 = 1 \rightarrow \{1\}$

S_3

$a_2 = 0 \rightarrow 1$
$a_1 = 0 \rightarrow 2$
$a_2 = 1 \rightarrow 2$
$a_1 = 1 \rightarrow 1$

Fig. 1. Illustrations to Examples 1 and 2

Example 2. Let us consider the decision table T_2 and three systems of decision rules S_1, S_2 and S_3 obtained on the basis of usual local reducts, local reducts based on the generalized decision and maximally discerning local reducts (see Figure 1). Let us consider two new objects $(0, 2)$ and $(2, 0)$. Systems S_1 and S_2 have no rules which are realizable on the new objects. However, the system S_3 has rules which are realizable on these new objects and, moreover, correspond to these objects different decisions.

4 Greedy Algorithm

We now describe the greedy algorithm which for given α, $0 \leq \alpha < 1$, decision table T and set of pairs $\mathcal{P} \subseteq DIS(T)$, $\mathcal{P} \neq \emptyset$, constructs an $(\alpha, \mathcal{P})$-test for T. Let T have m columns labeled by attributes $a_1, \ldots, a_m$.

Let us choose an attribute a_{i_1} with minimal number i_1 which separates maximal number of pairs from $\mathcal{P}$. Add a_{i_1} to the constructed $(\alpha, \mathcal{P})$-test. If a_{i_1} separates at least $(1-\alpha)|\mathcal{P}|$ pairs from $\mathcal{P}$, then stop. Otherwise, choose an attribute a_{i_2} with minimal number i_2 which separates maximal number of unseparated pairs from the set $\mathcal{P}$. Add a_{i_2} to the constructed $(\alpha, \mathcal{P})$-test, etc.

By $R_{\text{greedy}}(\alpha) = R_{\text{greedy}}(\alpha, \mathcal{P}, T)$ we denote the cardinality of the constructed $(\alpha, \mathcal{P})$-test for T.

4.1 On Precision of Greedy Algorithm

The following three theorems are simple corollaries of results from [20,21,8].

Theorem 1. *Let* $0 \leq \alpha < 1$ *and* $\lceil (1-\alpha)|\mathcal{P}| \rceil \geq 2$. *Then* $R_{\text{greedy}}(\alpha) < R_{\min}(\alpha) \cdot (\ln \lceil (1-\alpha)|\mathcal{P}| \rceil - \ln \ln \lceil (1-\alpha)|\mathcal{P}| \rceil + 0.78)$.

Theorem 2. *Let* $0 \leq \alpha < 1$. *Then for any natural* $t \geq 2$ *there exists a decision table* T *and a subset* $\mathcal{P}$ *of the set* $DIS(T)$ *such that* $\lceil (1-\alpha)|\mathcal{P}| \rceil = t$ *and* $R_{\text{greedy}}(\alpha) > R_{\min}(\alpha)(\ln \lceil (1-\alpha)|\mathcal{P}| \rceil - \ln \ln \lceil (1-\alpha)|\mathcal{P}| \rceil - 0.31)$.

Theorem 3. *Let* $0 \leq \alpha < 1$. *Then*

$$R_{\text{greedy}}(\alpha) \leq R_{\min}(\alpha)(1 + \ln(\max_{j \in \{1,\ldots,m\}} |\mathcal{P} \cap DIS(T, a_j)|)) \ .$$

4.2 On Polynomial Approximate Algorithms

Immediately from results obtained in [10,25] the next theorem follows.

Theorem 4. *Let* $0 \leq \alpha < 1$*. Then the problem of construction, for given* T *and* $\mathcal{P} \subseteq DIS(T)$*, an* $(\alpha, \mathcal{P})$*-reduct for* T *with minimal cardinality is* NP*-hard.*

From statements obtained in [8] (based on results from [1,17,23,25]) the next two theorems follow.

Theorem 5. *Let* $\alpha \in \mathbb{R}$ *and* $0 \leq \alpha < 1$*. If* $NP \not\subseteq DTIME(n^{O(\log \log n)})$*, then for any* ε*,* $0 < \varepsilon < 1$*, there is no polynomial algorithm that, for given decision table* T *with* $DIS(T) \neq \emptyset$ *and nonempty subset* $\mathcal{P} \subseteq DIS(T)$*, constructs an* $(\alpha, \mathcal{P})$*-test for* T *which cardinality is at most* $(1 - \varepsilon) R_{\min}(\alpha, \mathcal{P}, T) \ln |\mathcal{P}|$*.*

From Theorem 3 it follows that $R_{\text{greedy}}(\alpha) \leq R_{\min}(\alpha)(1 + \ln |\mathcal{P}|)$. From this inequality and from Theorem 5 it follows that under the assumption $NP \not\subseteq DTIME(n^{O(\log \log n)})$ the greedy algorithm is close to the best polynomial approximate algorithms for partial test cardinality minimization.

Theorem 6. *Let* α *be a real number such that* $0 \leq \alpha < 1$*. If* $P \neq NP$*, then there exists* $\rho > 0$ *such that there is no polynomial algorithm that, for given decision table* T *with* $DIS(T) \neq \emptyset$ *and nonempty subset* $\mathcal{P} \subseteq DIS(T)$*, constructs an* $(\alpha, \mathcal{P})$*-test for* T *which cardinality is at most* $\rho R_{\min}(\alpha, \mathcal{P}, T) \ln |\mathcal{P}|$*.*

From Theorems 3 and 6 it follows that under the assumption $P \neq NP$ the greedy algorithm is not far from the best polynomial approximate algorithms for partial test cardinality minimization.

4.3 Lower Bound on $R_{\min}(\alpha)$

In this subsection, we fix some information about greedy algorithm work and find a lower bound on $R_{\min}(\alpha)$ depending on this information.

Let us apply the greedy algorithm to α, T and $\mathcal{P}$. Let during the construction of $(\alpha, \mathcal{P})$-test for T the greedy algorithm choose consequently attributes $a_{j_1}, \ldots, a_{j_t}$. Let us denote by δ_1 the number of pairs from $\mathcal{P}$ separated by the attribute a_{j_1}. For $i = 2, \ldots, t$ we denote by δ_i the number of pairs from $\mathcal{P}$ which are not separated by attributes $a_{j_1}, \ldots, a_{j_{i-1}}$ but are separated by the attribute a_{j_i}. Let $\Delta(\alpha, \mathcal{P}, T) = (\delta_1, \ldots, \delta_t)$. As information on the greedy algorithm work we will use the tuple $\Delta(\alpha, \mathcal{P}, T)$ and numbers $|\mathcal{P}|$ and α.

We now define the parameter $l(\alpha) = l(\alpha, |\mathcal{P}|, \Delta(\alpha, \mathcal{P}, T))$. Let $\delta_0 = 0$. Then

$$l(\alpha) = \max \left\{ \left\lceil \frac{\lceil (1 - \alpha)|\mathcal{P}| \rceil - (\delta_0 + \ldots + \delta_i)}{\delta_{i+1}} \right\rceil : i = 0, \ldots, t - 1 \right\} .$$

Next two theorems follow immediately from results obtained in [8].

Theorem 7. *Let* T *be a decision table,* $\mathcal{P} \subseteq DIS(T)$*,* $\mathcal{P} \neq \emptyset$*, and* α *be a real number such that* $0 \leq \alpha < 1$*. Then* $R_{\min}(\alpha, \mathcal{P}, T) \geq l(\alpha, |\mathcal{P}|, \Delta(\alpha, \mathcal{P}, T))$*.*

The value $l(\alpha) = l(\alpha, |\mathcal{P}|, \Delta(\alpha, \mathcal{P}, T))$ can be used for the obtaining of upper bounds on cardinality of partial tests constructed by the greedy algorithm.

Theorem 8. *Let α and β be real numbers such that $0 < \beta \le \alpha < 1$. Then $R_{\mathrm{greedy}}(\alpha) < l(\alpha - \beta) \ln\left(\frac{1-\alpha+\beta}{\beta}\right) + 1$.*

From Theorem 8 it follows that the lower bound $R_{\min}(\alpha) \ge l(\alpha)$ is nontrivial.

4.4 Upper Bound on $R_{\mathrm{greedy}}(\alpha)$

In this subsection, we obtain an upper bound on $R_{\mathrm{greedy}}(\alpha) = R_{\mathrm{greedy}}(\alpha, \mathcal{P}, T)$ which does not depend on $|\mathcal{P}|$. The next statement follows immediately from Theorems 7 and 8.

Theorem 9. *Let α and β be real numbers such that $0 < \beta \le \alpha < 1$. Then $R_{\mathrm{greedy}}(\alpha) < R_{\min}(\alpha - \beta) \ln\left(\frac{1-\alpha+\beta}{\beta}\right) + 1$.*

5 Example of Greedy Algorithm Work

In this section, we consider results of an experiment with the decision table T "kr-vs-kp" from UCI repository. This table contains 36 conditional attributes and 3196 rows. In the capacity of the set $\mathcal{P}$ we consider the set of all pairs $(u_l, u_t) \in DIS(T)$ such that $d(u_l) \ne d(u_t)$ (so we study maximally discerning decision reducts). We apply to the decision table T, the set $\mathcal{P}$ and $\alpha = 0$ the greedy algorithm. Results of this experiment can be found in Table 1. The column

Table 1. Results of the experiment with the decision table "kr-vs-kp"

#	%	α	$l(\alpha)$	attr.	#	%	α	$l(\alpha)$	attr.
1	53.86	0.461391687916226	1.0	wknck	16	22.73	0.000079844850496	3.0	bkblk
2	55.80	0.203919051411864	2.0	bxqsq	17	26.47	0.000058709448894	4.0	wtoeg
3	55.01	0.091742516013002	2.0	wkpos	18	24.00	0.000044619181160	4.0	mulch
4	59.55	0.037112982420434	2.0	rimmx	19	24.56	0.000033660084033	4.0	thrsk
5	54.48	0.016895013806505	2.0	bkxbq	20	29.07	0.000023875175884	4.0	reskr
6	55.69	0.007487020319340	2.0	katri	21	29.51	0.000016830042016	5.0	qxmsq
7	49.28	0.003797718550816	2.0	simpl	22	37.21	0.000010567700801	5.0	bkxcr
8	45.27	0.002078314490862	2.0	r2ar8	23	29.63	0.000007436530193	5.0	skrxp
9	45.12	0.001140528893855	2.0	blxwp	24	31.58	0.000005088152238	5.0	bkona
10	39.91	0.000685334966761	2.0	dwipd	25	38.46	0.000003131170608	5.0	skach
11	37.98	0.000425056409995	2.0	bkspr	26	37.50	0.000001956981630	5.0	wkcti
12	40.70	0.000252059233920	3.0	cntxt	27	40.00	0.000001174188978	5.0	bkon8
13	31.37	0.000172997176076	3.0	skewr	28	66.67	0.000000391396326	5.0	dsopp
14	21.72	0.000135423128783	3.0	rxmsq	29	100.0	0.000000000000000	5.0	spcop
15	23.70	0.000103328630054	3.0	wkovl					

"#" contains the number i of step of greedy algorithm, the column "attr." contains the name of attribute chosen during the i-th step, the column "α" contains the inaccuracy of partial test constructed during the first i steps, the column "%" contains the percentage of unseparated during first $i-1$ steps pairs which are separated during the i-th step, and the column "$l(\alpha)$" contains the lower bound on minimal cardinality of $(\alpha, \mathcal{P})$-test for the table T.

From the obtained results it follows that $R_{\text{greedy}}(0.1) = 3$, $R_{\text{greedy}}(0.01) = 6$, $R_{\text{greedy}}(0.001) = 10$, and $R_{\text{greedy}}(0) = 29$.

In such situation instead of long exact test (29 attributes) it is better, sometimes, to work with short partial test (6 attributes) which separates more than 99% pairs of rows.

Acknowledgement

The authors are greatly indebted to Andrzej Skowron for stimulating discussions.

References

1. Feige, U.: A threshold of ln n for approximating set cover (Preliminary version). In: Proceedings of 28th Annual ACM Symposium on the Theory of Computing, pp. 314–318 (1996)
2. Grzymala-Busse, J.W.: Data with missing attribute values: Generalization of indiscernibility relation and rule induction. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B., Świniarski, R.W., Szczuka, M. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, Springer, Heidelberg (2004)
3. Johnson, D.S.: Approximation algorithms for combinatorial problems. J. Comput. System Sci. 9, 256–278 (1974)
4. Kryszkiewicz, M.: Rules in incomplete information systems. Information Sciences 113, 271–292 (1999)
5. Lovász, L.: On the ratio of optimal integral and fractional covers. Discrete Math. 13, 383–390 (1975)
6. Moshkov, M.Ju., Piliszczuk, M., Zielosko, B.: On construction of partial reducts and irreducible partial decision rules. Fundamenta Informaticae 75(1-4), 357–374 (2007)
7. Moshkov, M.Ju., Piliszczuk, M., Zielosko, B.: On partial covers, reducts and decision rules with weights. In: LNCS Transactions on Rough Sets VI. LNCS, vol. 4374, pp. 211–246. Springer, Heidelberg (2007)
8. Moshkov, M.Ju., Piliszczuk, M., Zielosko, B.: On partial covers, reducts and decision rules. LNCS Transactions on Rough Sets, Springer-Verlag (submitted)
9. Nguyen, H.S.: Approximate Boolean reasoning: foundations and applications in data mining. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, Springer, Heidelberg (2006)
10. Nguyen, H.S., Ślęzak, D.: Approximate reducts and association rules - correspondence and complexity results. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) New Directions in Rough Sets, Data Mining, and Granular-Soft Computing. LNCS (LNAI), vol. 1711, pp. 137–145. Springer, Heidelberg (1999)

11. Nigmatullin, R.G.: Method of steepest descent in problems on cover. In: Memoirs of Symposium Problems of Precision and Efficiency of Computing Algorithms 5. Kiev, USSR (in Russian), pp. 116–126 (1969)
12. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
13. Pawlak, Z.: Rough set elements. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery 1. Methodology and Applications (Studies in Fuzziness and Soft Computing 18), pp. 10–30. Springer, Heidelberg (1998)
14. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007). Rough sets: Some extensions. Information Sciences 177(1), 28–40 (2007) Rough sets and boolean reasoning. Information Sciences 177(1), 41–73 (2007)
15. Piliszczuk, M.: On greedy algorithm for partial reduct construction. In: Proceedings of Concurrency, Specification and Programming Workshop 2. Ruciane-Nida, Poland, pp. 400–411 (2005)
16. Quafafou, M.: α-RST: a generalization of rough set theory. Information Sciences 124, 301–316 (2000)
17. Raz, R., Safra, S.: A sub-constant error-probability low-degree test, and sub-constant error-probability PCP characterization of NP. In: Proceedings of 29th Annual ACM Symposium on the Theory of Computing, pp. 475–484 (1997)
18. Skowron, A.: Rough sets in KDD. In: Proceedings of the 16-th World Computer Congress (IFIP' 2000), Beijing, China, pp. 1–14 (2000)
19. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Slowinski, R. (ed.) Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory, pp. 331–362. Kluwer Academic Publishers, Dordrecht (1992)
20. Slavík, P.: A tight analysis of the greedy algorithm for set cover (extended abstract). In: Proceedings of 28th Annual ACM Symposium on the Theory of Computing, pp. 435–441 (1996)
21. Slavík, P.: Approximation algorithms for set cover and related problems. Ph.D. thesis. University of New York at Buffalo (1998)
22. Ślęzak, D.: Approximate reducts in decision tables. In: Proceedings of the Congress Information Processing and Management of Uncertainty in Knowledge-based Systems 3. Granada, Spain, pp. 1159–1164 (1996)
23. Ślęzak, D.: Normalized decision functions and measures for inconsistent decision tables analysis. Fundamenta Informaticae 44, 291–319 (2000)
24. Ślęzak, D.: Approximate decision reducts. Ph.D. thesis. Warsaw University (in Polish) (2001)
25. Ślęzak, D.: Approximate entropy reducts. Fundamenta Informaticae 53, 365–390 (2002)
26. Wróblewski, J.: Ensembles of classifiers based on approximate reducts. Fundamenta Informaticae 47, 351–360 (2001)
27. Yablonskii, S.V.: Tests. Encyclopedia of Cybernetics. In: Glushkov, V.M. (ed.) Main Editorial Board of Ukrainian Soviet Encyclopedia, Kiev (in Russian), pp. 431–432 (1975)
28. Ziarko, W.: Analysis of uncertain information in the framework of variable precision rough sets. Foundations of Computing and Decision Sciences 18, 381–396 (1993)
29. Zielosko, B.: On partial decision rules. In: Proceedings of Concurrency, Specification and Programming Workshop 2. Ruciane-Nida, Poland, pp. 598–609 (2005)

Applying Rough Set Theory to Medical Diagnosing

Piotr Paszek and Alicja Wakulicz–Deja

Institute of Computer Science, University of Silesia,
Będzińska 39, 41–200 Sosnowiec, Poland
{paszek,wakulicz}@us.edu.pl

Abstract. The work contains an example of applying the rough set theory to application of support decision making - diagnose Mitochondrial Encephalomyopathies (MEM) in a child. The resulting decision support system maximally limits the indications for invasive diagnostic methods that finally decide about diagnosis. Moreover, it shortens the time necessary for making diagnosis. System has arisen using induction (machine learning from examples) – one of the methods artificial intelligence.

1 Introduction

Our work presents an example of solving a diagnose problem in the complex systems of decision making. It is related to the diagnosing system in the children neurology, concerning diagnosing the patients suspected of suffering from Mitochondrial Encephalomyopathies (MEM). In this case unquestionable decision can be made only after performing some invasive tests.

At present, the disease is widely spread among children and its prognosis is bad [1], [7], [9], [10], [11].

It is not only essential to diagnose MEM early but to limit as well the number of patients subjected to invasive and health threatening tests to a minimum.

2 Medical Problem

Progressive encephalopathy (PE) is the group of illnesses occurring in neurology. It is a progressive loss of psychomotor and neuromuscular functions occurring in the infancy or in older children. Essential reasons for PE are metabolic diseases.

In the work we have paid attention to Mitochondrial Encephalomyopathies in which respiratory enzymes of the cell located in mitochondria's are impaired. MEM occur with elevated levels of lactic and pyruvic acid in the blood serum and the cerebrospinal fluid (CSF) [4].

Diagnosing MEM takes place on the basis of many parameters. The preliminary diagnosis of inbred diseases is mainly based on the clinical symptoms. The most essential feature of their majority is a progressive character pointing generally to degenerative genetic disease.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 427–435, 2007.

However, clinical symptoms do not decide about the final diagnosis of the disease. For that reason another group of tests used in diagnosing MEM are invasive tests. It includes sampling blood and cerebrospinal fluid in order to measure levels of appropriate parameters. One is interested in concentration levels of lactic and pyruvic acids in these media. Results of these measurements are the basis for further diagnosis. In spite of the fact that hyperlactemy, namely elevated levels of lactic or pyruvic acids in the blood or CSF, is a MEM discriminant, it is not the final confirmation of the diagnosis [6].

Sampling muscles and nerves belongs to a yet another group of invasive tests. Different types of tests: biochemical, morphological, genetic, are carried out on these samples. Results – particularly the measurement of the level of enzymes – determine (or exclude) a specific disease entity from the MEM group. It is the final confirmation (or negation) of the preliminary diagnosis. As it results from the diagnostic process description, it is complex and time- consuming. The final diagnosis is obtained as a result of the performed invasive tests. It is essential to lower the number of patients subjected to them.

The MEM etiology is not clear for all disease entities. In majority of cases, it has the genetic background, confirmed by discovery of a gene causing that disease. An early diagnosis is very essential for all metabolic and degenerative diseases, because in some of them a specific therapy is possible and additionally the genetic counseling depends on the proper diagnosis. In connection with this, it is equally important to shorten time of making the final diagnosis.

3 Selection of an Information Method for Supporting Diagnosis

A detailed analysis of the considered medical problem results in creating a three-stage diagnostic process, which allows classifying children as those suffering from MEM and those suffering from other diseases [12]:

1. The first stage requires diagnosis on the basis of clinical data.
2. The second stage requires a spinal puncture, sampling the cerebrospinal fluid (an invasive test) and performing biochemical tests.
3. The third stage requires biopsy of muscles and nerves (which is particularly dangerous for a patient.

Designing and developing a system supporting the MEM diagnosing process enables to shorten time necessary to make the final diagnosis and to perform the optimal classification at the first and second of the above stages, i.e. it enables to minimize the number of children subjected to invasive tests.

The main aim – shortening time necessary to make the final diagnosis – is related to data (attributes) reduction, finding relationships between attributes and the proper classification of data.

Additionally developing a supporting decision system in diagnosing MEM was connected with reducing of knowledge, generating decision rules and with a suitable classification of new information.

An analysis of the problems presented here leads to a natural application of the Rough Set Theory (RST) [8] in the conducted decision making process.

RST was used both in diagnostic processes and during creating the decision making system. The proposed system is a multistage decision making system reflecting the multistage medical diagnosis.

4 The Project of the Application of Support Decision Making

To speed up the MEM diagnosing process we need a support system. The system should consider the stage character of the diagnosis. The selection of the appropriate set of attributes taken into account during the classification at each stage is required and it is necessary to determine the values, which those attributes can obtain.

Next the knowledge base should be created – a set of rules – which would describe the MEM diagnosing process at each of the stages in the most complete way. The knowledge base formation is possible with use of the knowledge induction so called *machine learning* [2].

4.1 Machine Learning

In most cases the rule sets, induced from machine learning system from training data, are used for classification of new example, unseen before by the learning system. Because input data (training, unseen) are - in general - imperfect, a data preprocessing is required.

For the MEM diagnosis support the training set consisted of patients suspected of MEM. New data (unseen set), subjected to classification on the basis of the knowledge base obtained from the training set, are new patients suspected of MEM, which require diagnosing.

In the literature there are a lot of applications generating knowledge bases on the basis of examples. In view of medical application and using the rough set theory to create the knowledge base, the LERS system was chosen to select decision rules at each stage of diagnosing.

LERS (Learning from Examples based on Rough Sets) [5] is a program generating rules on the basis of the knowledge base (decision tables). In the LERS system there are two algorithms for the rules generating – LEM1 and LEM2.

4.2 Classification of New Cases

There are many schemes for classification of new objects.

While classifying new object we can say about complete and partial matching. In complete matching all attribute-value pairs of a rule must match all values of the corresponding attributes for the example. In partial matching some attribute-value pairs of a rule match the values of the corresponding attributes.

LERS first attempts complete matching. Every rule is equipped with a strength and specificity. Strength is a number of correctly classified examples during training, specificity is the total number of attribute–value pairs of the rule. Support of a decision class C is defined as follows:

$$Support(X) = \sum_{R:R\, describes\, X} Strength(R) * Specifity(R)$$

The decision class with the highest support wins the contest. When complete matching is impossible, partial matching is considered. In this case, the matching factor (M_factor) for a rule R is computed as the ratio of the number of matched attribute-value pairs of R to their total number. Then the support of X takes the form of

$$Support(X) = \sum_{R:R\, describes\, X} Strength(R) * Specifity(R) * M\_factor(R)$$

Further, we refer to that way of choosing the decision as to the "concept support" method.

We also used the "rule strength" classification method, applicable both to the set of completely matched, as well as partially matched rules. In this approach, we classify each new example using a rule with the largest strength. In case of partially matched rules, we choose the rules with the largest value of the product of strength by the matching factor.

4.3 The Selection of the Set of Attributes

In order to created the system it is necessary to determine the set of appropriate attributes and its values on each stage of diagnosis MEM.

Stage I. Classification based on the clinical symptoms. In the first stage diagnostics of children suspected of mitochondrial encephalomyopathies is based on clinical symptoms.

On the basis of the data obtained from II Clinic Department of Pediatrics of the Silesian Academy of Medicine it has been established that in diagnosing MEM at the first stage 27 features – attributes should be used.

After further analysis, a number of attributes were reduced by combining features describing similar symptoms. There were 12 so called "grouped" attributes created. Majority of the new grouped attributes have been formed by combining original attributes describing similar features.

The way of creating new attributes and the way of enumerating the value of new attributes was described in [14].

Table 1 presents the description of the created grouped attributes.

Stage II. Classification on the basis of biochemical data. Initially, four parameters were used to classify patients in the second stage. They described levels of lactic and formic acids in the blood serum and cerebrospinal fluid. However, there were cases (patients) where levels of those four attributes were normal but the proportions between those acids lost balance. Therefore, two new

Table 1. Attribute in the first stage of diagnosing MEM

#	attribute
1	– retardation and/or regress
2	– hypotony
3	– spasticity
4	– epileptic seizures
5	– ophthalmologic changes
6	– episodic vomitus
7	– brain system dysfunction
8	– circulatory system disturbance
9	– liver dysfunction
10	– disturbed dynamics of heat circumference
11	– ataxia
12	– acute hemiplegia

Table 2. Attribute in the second stage of diagnosing MEM

Number	attribute
1	– lactate level in blood;
2	– pyruvate level in blood;
3	– ratio of lactate to pyruvate level in blood;
4	– lactate level in CSF;
5	– pyruvate level in CSF;
6	– ratio of lactate to pyruvate level in blood;
7	– changes in the lactate and/or pyruvate level in blood and CSF.

attributes were introduced. The resulting set of attributes included 7 parameters described in Table 2.

Attributes used in the second stage of diagnosing determined levels of acids and ratios of those acids in the blood serum and cerebrospinal fluid. Thus, values of those attributes are real numbers (continuous values). For such attributes discretization of values should be made [3].

In the work [13] we check quality of classification rules, which were obtained using different discretization methods of the attributes obtained in the second stage of diagnosing MEM. Results obtained in this work suggest explicitly that the method based on evaluation of norms on basis of a control group leads to the best results (smallest error rate).

The limit values obtained in this method are presented in Table 3. For attributes describing levels of lactic acid in blood and cerebrospinal fluid, yet another boundary value – pathology – was added. It has been added in order to maintain compliance with norms used by the physicians from the Clinic of Pediatrics.

Table 3. Norms for real attributes

A (attribute)							
1		2	3	4		5	6
norm	pathology	norm	norm	norm	pathology	norm	norm
1,96	2,5	0,43	33,1	2,47	3,0	0,3	24,1

Using the calculated boundary values of norms for acids discretization of data was made. Because some patients had some attributes measured several times and values of the same attributes were different in successive tests, another attribute was added which were to reflect changes in levels of acids (Table 2 - attribute #7).

With such treatment of the attribute value in the second stage of diagnosing and introducing a new attribute, the fact that tests of measurement of the level of acids were repeated for the same patient. On the basis of the modified set of attributes rules classifying patients for the third stage were made.

Stage III. Level of enzymes – the final diagnosis. In the third stage, segments of muscles or nerves are sampled to evaluate enzymatic activity. The segment of the skeletal muscle is most often taken, rarely a segment of nerves [6]. For these specimens different types of tests – biochemical, morphological and genetic – are made. The results of the above investigations – particularly the measurement of level of enzymes – give the final diagnosis.

All these tests were made outside the Clinic of Pediatrics, so we did not have access to these results. Thus, the third stage of diagnose isn't taking into consideration in the decision support system.

5 Quality of System Classification

The evaluation of the classifier can be performed using the machine learning method based on splitting data in the training and the testing samples, or the 10-fold validation test.

5.1 Results of the First Stage of Diagnosing

Quality of classification is presented in Table 4.

The results obtained by use of two different methods of quality evaluation of rules do not differ significantly.

The lowest classification error – 11 cases out of 186 (5.91%) – was obtained in the 10-fold cross validation of rules, with the LEM1 algorithm, classification: strength of rules or concept support. It should be noticed that for both methods it was enough to use the complete match of objects with the rules.

The lowest classification error both in machine learning and 10-fold cross validation of rules occurred for the LEM1 algorithm and classification based on strength of rules. Therefore in the MEM diagnosis support program in the

Table 4. Error rates in first stage of diagnosis

Classification scheme	rule strength				concept support			
matching	complete		partial		complete		partial	
algorithm	LEM1	LEM2	LEM1	LEM2	LEM1	LEM2	LEM1	LEM2
Machine learning								
Training set:	114				cases			
Unseen set:	72				cases			
Correctly classified	68	66	68	66	68	66	68	66
Incorrectly classified	4	6	4	6	4	6	4	6
Unclassified	0	0	0	0	0	0	0	0
error rate	0.06	0.08	0.06	0.08	0.06	0.08	0.06	0.08
10-fold cross validation								
Training set:	186				cases			
correctly classified	175	174	175	174	175	173	175	173
incorrectly classified	11	12	11	12	11	13	11	13
Unclassified	0	0	0	0	0	0	0	0
error rate	0.06	0.06	0.06	0.06	0.06	0.07	0.06	0.07

first stage of diagnosis, the LEM1 algorithm was used to generate rules, whereas during classification strength of rules was used while selecting rules.

5.2 Results of the Second Stage of Diagnosing

Table 5 present the classification quality for various combinations of the classifier parameters, while using the train & test (114 training and 92 testing cases) and 10-fold cross-validation (206 cases) methods.

Comparing the results obtained in the 10-fold cross validation of rules with ones obtained in the machine learning, great difference in the classification errors can be noticed. There was also a great difference in the number of non-classified cases for those two methods. Such great differences result from incomplete data (missing values of attributes). In the machine learning from examples method patients with incomplete data were only in the testing set. In the 10-fold validation of rules the patients with incomplete data were mixed (they were in learning and testing set). While using the partial match in the machine learning non-classified patients were correctly classified (all cases).

The lowest classification error was for the machine learning method with the partial match of an object with a rule, with the classification on the basis of strength of rules, for rules generated by the LEM1 algorithm and it was 6.52% (6 cases out of 92). For the 10-fold cross validation of rules at the partial match of an object with a rule, with the classification based on strength of rules, for rules generated by the LEM1 algorithm, it was 8.27% (17 out of 206).

For that reason in the second stage of diagnosis, in the MEM diagnosis support program, the LEM1 algorithm was used to generate rules, and during classification of new cases a scheme based on the partial match of an object with a rule on the basis of strength of a rule.

Table 5. Error rates in second stage of diagnosis

Classification scheme	rule strength				concept support			
matching	complete		partial		complete		partial	
algorithm	LEM1	LEM2	LEM1	LEM2	LEM1	LEM2	LEM1	LEM2
Machine learning								
Training set:	114				cases			
Unseen set:	92				cases			
correctly classified	57	57	86	86	55	56	84	85
incorrectly classified	6	6	6	6	8	7	8	7
unclassified	29	29	0	0	29	29	0	0
error rate	0.38	0.38	0.07	0.07	0.40	0.39	0.09	0.08
10-fold cross validation								
Training set:	206				cases			
Correctly classified	188	186	189	187	185	187	186	188
Incorrectly classified	17	19	17	19	20	18	20	18
Unclassified	1	1	0	0	1	1	0	0
error rate	0.09	0.10	0.08	0.09	0.10	0.09	0.10	0.09

6 Conclusion

The work contains an example of application of Rough Set Theory to decision making – diagnosing Mitochondrial Encephalomyopathies for children. The work presents a project and implementation of the support diagnostic process.

The proposed system consists of two stages of medical diagnosis. The resulting decision support system maximally limits the indications for invasive diagnostic methods (puncture, muscle and/or nerve specimens) that finally decide about diagnosis. Moreover, it shortens the time necessary for making diagnosis.

The solved problems have clearly applied aspects after verification on the real data, obtained from the Clinic of Pediatrics.

References

1. Barkovich, A.J.: Toxic and metabolic brain disorders. In: Pediatric Neuroimaging, pp. 55–105. Raven Press Ltd, New York (1995)
2. Carbonell, J.: Machine learning Paradigm and Methods. MIT Press, Cambridge (1989)
3. Chmielewski, M.R., Grzymała-Busse, J.W.: Global discretization of continuous attributes as preprocessing for machine learning. In: Lin, T.Y., Wilderberger, A.M. (eds.) Soft Computing, Simulation Councils, San Diego, pp. 294–297 (1995)
4. Eymard, B., Hauw, J.J.: Mitochondrial encephalomyopathies. Curr. Opin. Neurol. Neurosurg 5, 909–916 (1992)
5. Grzymała-Busse, J.: A New Version of the Rule Induction System LERS. Fundamenta Informaticae 31, 27–39 (1997)
6. Marszał, E. (ed.): Leukodystrofie i inne choroby ośrodkowego układu nerwowego z uszkodzeniem istoty białej u dzieci i młodzieży. Śląska Akademia Medyczna (1998)

7. Matthews, P.M., Anderman, F., Silver, K., Karpati, G., Arnold, D.L.: Proton MR spectroscopic characterization of differences in regional brain metabolic abnormalities in mitochondrial encefalomyopathies. Neurology 43, 2484–2490 (1993)
8. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer, Dordecht (1991)
9. Tulinius, M.H., Holme, E., Kristianson, B.: Mitochondrial encephalomyopathies in childhood: 1. Biochemical and morphologic investigations. J. Pediatrics 119, 242–250 (1991)
10. Tulinius, M.H., Holme, E., Kristianson, B.: Mitochondrial encephalomyopathies in childhood: 2. Clinical manifestation and syndromes. J. Pediatrics 119, 251–259 (1991)
11. Uvebrant, P., Lanneskog, K., Hagberg, B.: The Epidiemiology of Progressive Encephalopathies in Childhood. I Live Birth Prevalence in West Sweden. Neuropediatrics 23, 209–211 (1992)
12. Wakulicz-Deja, A., Paszek, P.: Diagnose Progressive Encephalopathy Applying the Rough Set Theory. International Journal of Medical Informatics 46, 119–127 (1997)
13. Wakulicz-Deja, A., Boryczka, M., Paszek, P.: Discretization of continuous attributes on Decision System in Mitochondrial Encephalomyopathies. In: Polkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS (LNAI), vol. 1424, pp. 483–490. Springer, Heidelberg (1998)
14. Wakulicz-Deja, A., Paszek, P.: Applying Rough Set Theory to Multi Stage Medical Diagnosing. Fundamenta Informaticae 20, 1–22 (2003)

Graph-Based Knowledge Representations for Decision Support Systems

Roman Simiński

University of Silesia, Institute of Computer Science, Poland, 41-200 Sosnowiec,
Będzińska 39, Phone (+48 32) 3 689 866
roman.siminski@us.edu.pl

Abstract. Expert systems are problem solvers for specialized domains of competence in which effective problem solving normally requires human expertise. Rough set theory is intelligent technique used in the discovery of data dependencies; it evaluates the importance of attributes, reduces all redundant objects and attributes. Moreover, it is being used for the extraction of rules from databases. Expert systems are often implemented using knowledge discovered in data bases, for example, using rough set based rule generation method. When we try to analyze large, possibly hierarchical rule sets, we often seek useful graphical representation.

A proposition of such graphical representation decision networks we can find in [4]. The main aim of this work is to present own graph-based rule base representation method and its utilization for verification task. The paper firstly presents the decision units conception needed to establish our verification approach. Next we present the usage of decision units net in global verification and modeling issues, and some remarks on decision units and data mining. The last chapter draws the main conclusions.

1 Introduction

In recent years, knowledge based systems technology has proven itself to be a valuable tool for solving hitherto intractable problems in domains such a telecommunication, aerospace, medicine and the computer industry itself. Some expert system are deemed successful if they make or save large sums of money, while other succeed because they help their users to understand better their own knowledge. The goals of expert system are often more ambitious than of conventional programs. They frequently perform not only as problem solvers but also as intelligent assistant and training aids.

Rough set theory [2][3] was developed by Z. Pawlak in Poland, in the early 1980s, and concerns itself with the classifcatory analysis of imprecise, uncertain or incomplete information expressed in terms of data acquired from experience. Rough set theory is intelligent technique used in the discovery of data dependencies; it evaluates the importance of attributes, reduces all redundant objects and attributes. Moreover, it is being used for the extraction of rules from databases.

Expert systems are often implemented using knowledge discovered in data bases, for example, using rough set based rule generation method. Rule based

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 436–444, 2007.

knowledge representations are perhaps the most popular ones. Expert system behavior arises from cooperative interaction of rules in knowledge base interpreted by inference engine. For knowledge bases with hundreds or thousands of the rules number of possible inference paths is very high. In such cases knowledge engineer can not be totally aware that all possible rules interactions are legal and provide expected results [7]. When we try to analyze large, possibly hierarchical rule sets, we often seek useful graphical representation. A proposition of such graphical representation decision networks we can find in [4].

The main aim of this work is to present own graph-based rule base representation method and its utilization for verification task. The paper firstly presents the decision units conception needed to establish our verification approach. Then we present briefly the decision unit as the tool for local rule base verification [1] [5] [6] and modeling [12]. Next we present the usage of decision units net in global verification and modeling issues, and some remarks on decision units and data mining. The last chapter draws the main conclusions.

2 Conception of Decision Units

We assume, that decision units are the main tool for our rule base verification method. Let us present decision units conception using the following example. We consider the following rule base, containing five rules. For simplification of presentation, we assume that the propositional level of logic is used.

$r_1 : p \wedge q \rightarrow s$
$r_2 : s \wedge t \rightarrow u$
$r_3 : p \wedge q \wedge t \rightarrow u$
$r_4 : s \wedge f \rightarrow p$
$r_5 : t \wedge s \rightarrow u$

We can present an example knowledge base as a directed graph. The first way of graph oriented presentation of knowledge base shows the following Fig. 1:

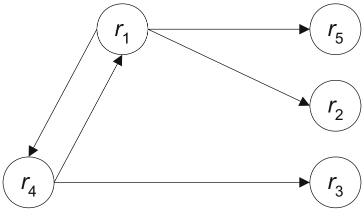


Fig. 1. First form of graphic representation — rule relationship diagram

The rule relationship diagram contains nodes and arcs. Rules with common literals in their antecedents and consequents might be naturally related. Each node corresponds to one rule. The arcs between the nodes define the relation between the conditions and conclusions of the rules. The dependences between rules might cause a cycles. Let suppose that our goal is s. A backward chaining inference engine might choose rule r_1 as relevant to this goal. Two subgoals are created from antecedent of r_1. The first rule relevant to subgoal p is rule r_4. Unfortunately, the antecedent of r_4 requires to confirm s, which is the initial goal and inference engine enters into a cycle. Using rule relationship diagram we can easily identify the circle between the rule r_1 and r_4.

Rule relationship diagram can not provide more information useful for verification. The second way of graph oriented presentation of knowledge base shows the following Fig. 2.

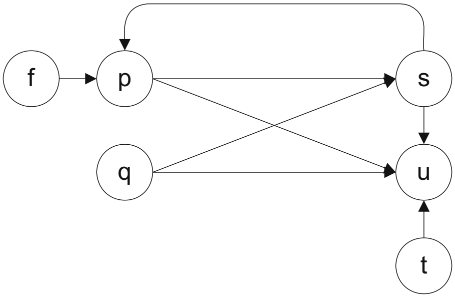


Fig. 2. Second form of graphic representation — literal relationship diagram

The literal relationship diagram provides a graphical display of dependences between literals used to reach an inference. Using the diagram of this kind we can detect circle $p \rightarrow s \rightarrow p$, but this form of presentation is not clear especially if the knowledge base contains, for example, a hundred rules. The graph oriented representation of rule base from Fig. 1 and Fig. 2 might be mixed together. Fig. 3 shows the result join representation.

The rule structure diagram is a directed graph which links literals and rules. On the right side we will find the start literal relevant to the main goal, with arcs to the rule in which the goal literal appears in the consequent. Further arcs will link to rule nodes in which these literals appear the antecedent of the rule. Thus, the diagram shows inference paths starting from goal literal. The rule structure diagram can be used to locate some kinds of anomalies in knowledge base. We can identify the circle between the rule r_1 and r_4, redundancy - rules r_2 and r_5 are the same. Also other kind of redundancy can be detected. Rule r_3 is a logical consequence of rules r_1 and r_2. Literal u can be inferred from rules r_1 and r_2. The rule structure diagram seems to be a very useful tool for verification of rule knowledge bases. Unfortunately, in the real word knowledge bases contains hundreds or thousands rules. There is no way to display real word rule bases on present computerdevices — this is the first disadvantage of rule structure diagrams. It is not theoretically hard to develop algorithms to check

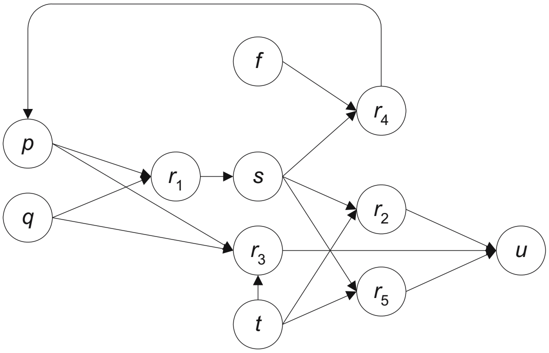


Fig. 3. The join graphic representation — rule structure diagram

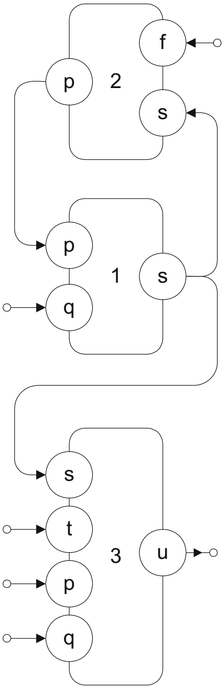


Fig. 4. Decision units net for an example knowledge base

for particular set of anomalies, but the existing algorithms in practice take too much time — this is the second disadvantage of rule structure diagrams.

We can show our example knowledge base in different way. All rules with the same concluding literal we can group together, this rule group we will call decision unit. Decision unit contains the set of rules with his same literal in the decision part (conclusion) of each rule. All literals which appear in the conditional part of each rule we will call input entry of decision unit. All literals appearing in the decision part of each rule we will call output entry of decision unit. Our example knowledge base contains three sets of rules with the same literal in the conclusion. Therefore we can show an example knowledge base using three decision units like on the following Fig. 4.

In next chapters we present more precise description of the decision units and their properties useful for knowledge base modeling and verification.

3 Description of Decision Units

In the real word rule knowledge bases literals are often coded using attribute value pairs. Now we introduce conception of decision units for literals as attribute value pairs. The paper [11] contains an example rule base with attribute value pairs. All rules with the same concluding literal we can group together, this rule group we will call decision unit. Decision unit U contains the set of rules R with this same literal in the decision part (conclusion) of each rule $r \in R$. All literals which appear in the conditional part of each r rule we will call input entry I of decision unit U. All literals appearing in the decision part of each r rule we will call output entry O of decision unit U. Fig. 5 presents the structure of the decision unit U.

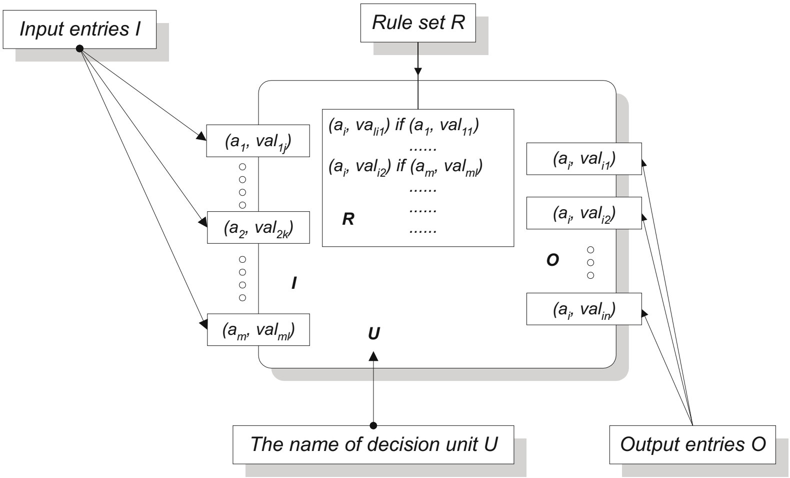
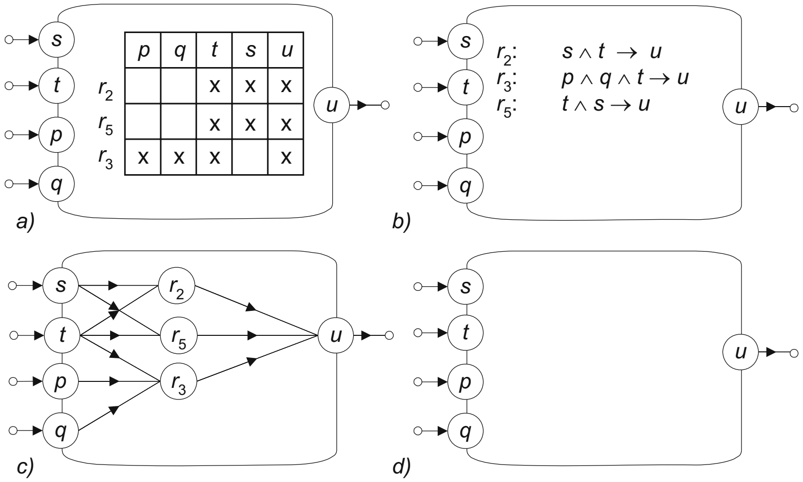


Fig. 5. The structure of the decision unit

4 The Usage of Decision Units for Modeling and Verification

Basing on considered example we can briefly describe how decision units may be used in verification and validation issues. Decision unit may be considered as a model of elementary decision produced by the knowledge base. Each decision unit allows to confirm set of goals described by the output entries O. Knowledge engineer can work with a decision unit like a programmer works with a procedure or function. Therefore decision unit separately considered is a tool for modeling on local level — the level of elementary decision. Decision unit can be considered as a tool for verification on the local level too. Verification may be done using back box or glass box testing method (Fig. 6). We can also apply static verification (classical anomaly detection) or dynamic techniques (using forward and backward chaining inferences).



	p	q	t	s	u
r_2			x	x	x
r_5			x	x	x
r_3	x	x	x		x

r_2: $s \wedge t \rightarrow u$
r_3: $p \wedge q \wedge t \rightarrow u$
r_5: $t \wedge s \rightarrow u$

Fig. 6. The structure of the decision unit

The net of the decision units may be considered as a global model of decisions produced by the system [9] [10]. Expert systems often confirm global goal using subgoals — we assume that the each subgoal is modeled by the appropriate decision unit. Therefore knowledge engineer may check if the current content of rule base is consistent with intended global decision model. It is specially useful in real-word problems if the particular knowledge representation language doesn't provide a solution for knowledge base partitioning.

The decision unit net allows us to formulate the global verification method similarly to local verification on the level of decision unit. Fig. 7 presents a conception of global verification. Unchained output entries represent main goals of rule base (like u on Fig. 7), chained output entries represent subgoals (like s).

Unchained input entries represent the input data (facts) necessary to produce proper inference results (like q, t). Booth chained and unchained entry may be a sign of anomaly presence (like p). We can apply static and dynamic verification on the global level using black box and glass box techniques. For example knowledge engineer can observe the current inference path on the global level for selected goal. An example on Fig. 7 contains circular relationship beetween units 1 and 2. In the case of improper system behavior, knowledge engineer can apply selected verification method for detection the sources of anomalies.

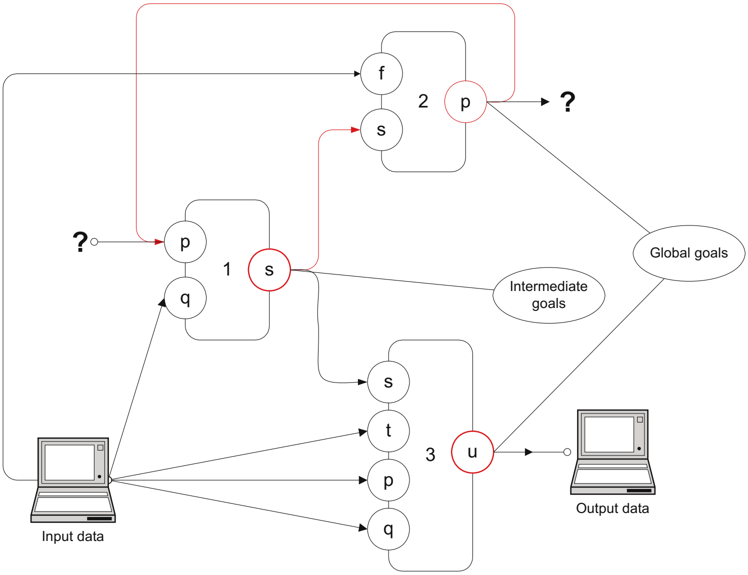


Fig. 7. The structure of the decision unit

5 Decision Units and Data Mining

The main problem in data mining consists of discovering knowledge hidden in data sets. This knowledge may be expressed by one distinguished attribute called decision attribute. The decision attribute may take several values though binary outcomes are rather frequent. Our goal is to discover the explicit knowledge — this knowledge is usually expressed in the form of decision rules. Decision rules typically contain one (this same) decision attribute in the conclusion part of the rule. Therefore rules generated from decision table we may consider as decision unit. Usually, if we consider decision system with one decision attribute, we will obtain one decision unit as a result of rule generation process. Thus decision units aren't very useful tool for rules generated from single data source.

But when we consider decision tables with more than one decision attributes, decision units allow us to divide rules set into the subsets. Decision units conception may be also useful for experiments with large, multidimensional data sets, when we try to discover partial knowledge from subtables, possibly hierarchically organized. Verification issues considered with decision units are typically useless when we generate rules from data. The methods of data exploration — rough sets method for example — take care of generated rules quality and anomaly extermination. Therefore decision units may be considered as a tool for clear and user friendly visualization.

6 Concluding Remarks

In our opinion the decision units conception allows us to consider different verification and validation issues together. Thanks to properties of the decision units we can perform different verification and validation actions during knowledge base development and realization. We can divide anomalies into the two levels — local and global anomalies and perform verification on those levels. Decision unit is a simple decision model, useful and efficient for knowledge base modeling and verification. The net of the decision units is a simple tool for modeling large real world knowledge bases.

Decision unit net allows us to perform global verification — i.e. circularity detection, dead end rules, auxiliary rules. Graphical representation of knowledge base in the form of the decision units net is user friendly and is an efficient and useful way of presentation of current knowledge base contents. The methods of data mining — rough sets method for example — take care of generated rules quality and anomaly extermination. Thus decision units aren't very useful tool for rules generated from single data source — decision units may be considered as a tool for clear and user friendly visualization.

References

1. O'Keefe, R.M., Balci, O., Smith, E.P.: Validating Expert Systems Performance. IEEE Expert 4(4), 81–90 (1987)
2. Pawlak, Z.: Rough sets. International Journal of Information and Computer Science 11(5), 341–356 (1982)
3. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data, volume 9 of Series D: System Theory. In: Knowledge Engineering and Problem Solving, Kluwer Academic Publishers, Dordrecht (1991)
4. Pawlak, Z.: Decision networks. In: Rough Sets and Current Trends in Computing. LNCS, Springer, Heidelberg (2004)
5. Preece, A.D.: Foundation and Application of Knowledge Base Verification. International Journal of Intelligent Systems 9 (1994)
6. Preece, A.D.: Validating Dynamic Properties of Rule-Based Systems. International Journal of Human-Computer Studies 44 (1996)
7. Simiński, R.: Methods and Tools for Knowledge Bases Verification and Validation. In: Proceedings of the CAI'98, Colloquia in Artificial Intelligence, 28–30.09.1998, ódź, Poland (1998)

8. Simiński, R., Wakulicz-Deja, A.: Principles and Practice in Knowledge Bases Verification. In: Proceedings of the IIS VII, Intelligent Information Systems, IIS'98, 15–19.6.1998, Malbork, Poland (1998)
9. Simiński, R., Wakulicz-Deja, A.: Errors And Anomalie. In: Rule Knowledge Bases, Proceedings of EUFIT'99, 11–14.9.1999, Aachen, Germany (1999)
10. Simiński, R., Wakulicz-Deja, A.: Dynamic Verification Of Knowledge Bases. In: Proceedings of the IIS VIII, Intelligent Information Systems,VIII, IIS'99, 14-18.6.1999, Ustro, Poland (1999a)
11. Simiński, R., Wakulicz Deja, A.: Verification of Rule Knowledge Bases Using Decision Units. In: Advances in Soft Computing, Intelligent Information Systems, Springer, Heidelberg (2000)
12. Simiński, R., Wakulicz-Deja, A.: Decision units s a tool for rule base modeling and verification. In: Information Processing and Web Maining, Advances in Soft Computing, Springer, Heidelberg (2003)

Rough Sets in Oligonucleotide Microarray Data Analysis

Magdalena Alicja Tkacz

University of Silesia, Institute of Computer Science, Będzińska 39,
41-200 Sosnowiec, Poland
tkacz@us.edu.pl

Abstract. This paper shows attempts of the rough set theory application to the oligonucleotide microarrays data analysis.

1 Background and Motivation

In natural sciences there is a lot of ways to gain data. High performance and throughput instruments give scientists possibility to make experiments in a more efficient way. But, unfortunately it does not significantly speeds up progress in researches. Unfortunately, in some scientific fields it reduces data usability - because of their quantity. Such kind of field are inter alia life sciences, biomedical sciences and some fields of bioinformatics. Today we are able to collect and manage data without problems. Database systems and engines are of pretty well development stage and quite good performance and quality - so any simple data manipulation process (eg. adding record, sorting, searching data under different criterion, deleting etc.) are rather a standard and not troublesome processes. To find useful information and relationships within data collected in databases is a more complicated and difficult task. But in the most cases, there are no simple methods for data manipulation and processing - this kind of analysis needs a more advanced methods and algorithms than basic statistical, searching, sorting and filtering algorithms for example algorithms usually classified as knowledge discovery from data, data mining, machine learning methods, computational or artificial intelligence. This paper shows an attempt of rough set utilization to gene expression analysis, which belongs to relatively new research field - bioinformatics.

2 Gene Expression

Living, growing, organization and working of all plants and animals on earth depends on hereditary information encoded in DNA strands. There are many attempts to decode this information, some of them are found, but most remains to be discovered. Gene is [2] a basic hereditary information, which is responsible for conveying features to the progeny and for susceptibility to diseases or, during lifetime of an organism, straightforward for disease or dysfunction appearing.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 445–453, 2007.

Genome is a complete, enclosed in the genetic cell information, including genes and other DNA sequences. Because genes responsible for every cell behavior, and all functions influences the cell activity of all living systems, it is important to find out impact and dependencies of cell functions in dependency of certain gene activity, or to recognize impact of drugs to the gene(s) behavior. One of gene function is coding of proteins. The higher ,,active" the gene, the more encoded protein products in the cell effects their properties. On the other hand, gene activity reduction causes decreased certain protein production - and has an impact on cell functions. Moreover - more proteins of one kind may cause increasing or decreasing production of proteins in an other gene - and so on - we have a kind of cascade of different reactions and processes. Thus the important question is what happens with certain genes - or its groups when their activity increased or decreased. All processes from reading encoded in gene information to production of proteins or different forms of RNA are referred to gene expression [3,4]. Sometimes, as a kind of mental shortcut, when gene activity can be measured in numbers, as in the case of microarray analysis they referred to gene expression [3,4]. Sometimes, as a kind of mental shortcut, when gene activity can be measured in numbers, as in the case of microarray analysis the term "gene expression" reflects the "activity" of genes which is directly proportional to gene number in the probe, and consequently to the amount of its products (for example - proteins). There are a few other methods to measure the gene expression, but most frequently used and relatively fast in utilization are the oligonucleotide microarrays. In one turn, from one genechip we are able to obtain information about 20 000 genes (in newest genechip - more than 40 000 genes).

3 Oligonucleotide Microarrays and Data Acquisition

Term "microarray analysis" is a some kind of simplification or mental shortcut, because it is composed from a few steps (see fig 1) form biological sample to data about gene expressions. These steps (not described in details) are:

1. Preparing a biologic tissue,
2. Preparation of the isolated and prepared sample on a genechip,
3. Reading information from the genechip by the use if a special scanner and transferring them to the computer,
4. Data processing and analysis for obtaining useful information.

The microarray method is not a single set of equipment. It is rather a set of special devices which makes possible conducting a microarray experiment. One component is a part named a gene- or genomechip. It consist of a one-strand DNA species placed on solid surface (glass, plastic etc), each of this species are placed in specific position on the surface - making some kind of array (see fig 2). Each probe - "point" on surface suits certain gene. Each microarray experiment always requires the use of two genechips, where one is an experimental chip, the second is a control chip.

After preparation, sample with labeled strands (genes) is drifting on the genechip. Complementary strands connects to probesets on genechip. After that,

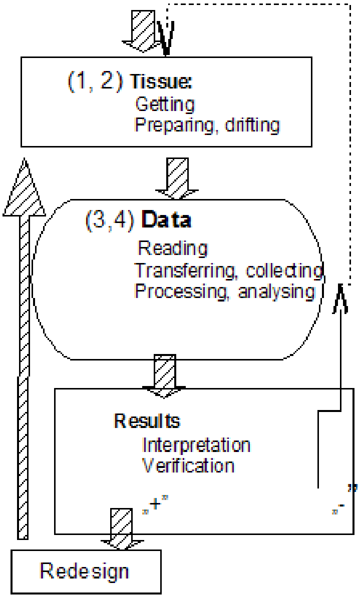


Fig. 1. Steps in microarray experiment

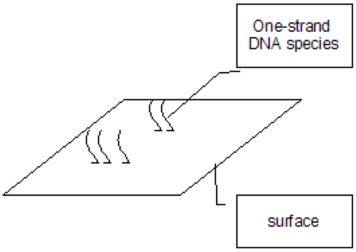


Fig. 2. DNA species on genechip

intensity of fluorescence (markups) is measured by the use of special scanner. So, we may have data in two possible formats: as a picture with spots of different intensity, or as a table with numbers reflected the intensity. For more information see [5]. For processing with rough sets, the second form is more suitable. Exemplary dataset from genechip is shown in table 1.

In fact, having only one fluorescence level reading from one microchip, the only thing we can do is to group together coexpressed genes. In most cases, in that way it is possible to separate only high- and low expressed genes. Genes for which one can looking for, are ,,inside" and ,,outside" of a ring - if one can decide to treat a number of reflected gene expressions as a radius, or above and below a belt if the final decision is to treat a number of reflected gene expression as a height of a bar (see fig 3 a, b). Coexpressed genes will be then either on

Table 1. Exemplary dataset (a few records from more than 20 000) from one genechip

Probeset ID	expression
1487_at	7,079
1494_f_at	4,845
177_at	4,013
179_at	7,544

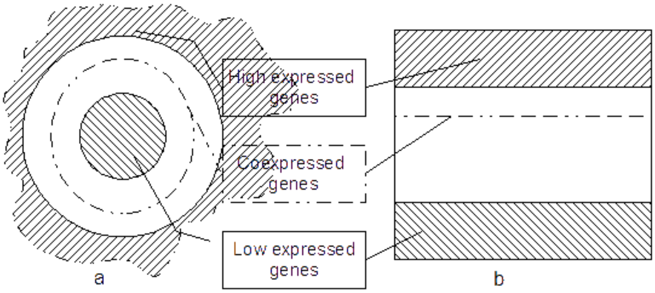


Fig. 3. Areas of high and low expressed genes in the case of one microarray experiment

the circumference of a circle with a given radius, or on one given level when expressions will be visualized as bars. However, the problem with setting up an appropriate "level" or boundary for high- and low expressed genes is still open.

As shown above, in the case of one microarray experiment there is not a lot of information to be obtained from the collected data. The power of gene analysis (and, at the same time - problems with that) becomes visible in comparative analysis of many microarray experiments data available. Almost always some (one or more) microarray experiments data are treated as a reference pattern (specimen - healthy cell(s)), and others, as samples from tissues with some dysfunction. The key is to find out genes which expression differs (higher or lower expression) form each other: in "healthy" samples and in "ill" samples. Here, because of some kind of uncertainty (caused for example by image processing, some kind of noise, diversity of biological specimens etc.) we can try to use rough sets [19,20,21,22] to catch genes with different expression in different probes from microarray readings, for feature selection [25], and, maybe in future research for process mining [16,17,23,24,26] from data[1]. The very first attempt,briefly presented below is an attempt of using some of the rough set theory to find out significantly differential expressed gene or group of genes, where the gene expression reflects the response of the genes ,,activity" in the human cancer cell lines to cytotoxic compounds [15]. The quantity of genes for future examination should be as small as possible for future processing (to avoid processing of all

[1] It is worth to remark, that nevertheless a [17] is very often cosidered as the first work about this problems, the paper of professor Z. Pawlak [23] appears earlier - in 1992.

genes - leaving only those with significance) - so, to achieve this result the idea of using equivalence classes, and next - core and reducts appears.

4 Rough Sets and Its Application in Gene Expression Analysis

First of all, we have to describe obtained data in any form suitable for machine processing. For that, we need to represent our data in form of triple (O,A,V) - object, attribute, attribute value. Because of a noise and diversity of biological specimens it is almost inevitable that we will have different gene expression for different Probeset ID and for different microarray experiments data. Such dataset has a form similar to presented in Table 2

Table 2. Raw dataset from microarray experiments

Probeset ID	Specimen	Sample1	Sample2	Sample3
1007_s_at	5,511	5,494	5,185	4,945
1053_at	6,758	6,796	7,345	5,061
117_at	3,715	3,715	3,892	3,885
121_at	3,717	6,769	6,856	6,437

Thus, first of all, a data discretization has been done to reduce quantity of data. The problem here is - how to set the ”boundary” in discretization. For example, if certain gene expression for different samples are: 4,363; 5,458; 5,689; 6,142 - should we make two sets with numbers in range from 4 to 5,5 and from 5,5 to 6,5 or three: from 4,3 to 5,3; from 5,31 to 5,6 and from 5,61 and more? This is a known problem concerning clustering - how many clusters will be appropriate to get the best solution? Which criterion will be optimal to divide the whole dataset? In this paper a primitive method for discretization has been used: all numbers from range 3,5 to 4,499 has been indicated as ”Range 4”, from 4,5 to 5,499 as ”Range 5”, from 5,5 to 6,499 as ,,Range 6”, from 6,5 to 7,499 as ”Range 7”, and all ranges are of equal length. it seems that choice of appropriate method for gene expression discretization will require at least several attempts in different dataset dividing. After each division, the rest of calculations should be made - and results should be next verified and interpreted by biosciences researchers in context to its usefulness in real. After such discretization we have a dataset, which is similar to the presented in Table 3. Together with discretization, a transposition of the initial table has been made.

So, now we have a: names of the samples as a set of objects (rows) $U = \{Specimen, Sample1, Sample2, Sample3\}$, Probeset IDs (geneID, columns) as attributes *Universum* $A = \{1007\_s\_at, 1053\_at, 117\_at, 121\_at\}$, and ranges of gene expression as attribute values. First that we can see after discretization process, is that all attribute values becomes the same for Sample 1 and Sample 2. Therefore we do not need to take into account both samples. One of them

Table 3. Dataset from microarray experiment after discretization and transposition

Probeset ID	1007_s_at	1053_at	117_at	121_at
Specimen	Range 6	Range 7	Range 4	Range 4
Sample1	Range 5	Range 7	Range 4	Range 7
Sample2	Range 5	Range 7	Range 4	Range 7
Sample3	Range 5	Range 5	Range 4	Range 6

Table 4. Dataset from microarray experiment after deletion of duplicated object

Probeset ID	1007_s_at	1053_at	117_at	121_at
Specimen	Range 6	Range 7	Range 4	Range 4
Sample1	Range 5	Range 7	Range 4	Range 7
Sample3	Range 5	Range 5	Range 4	Range 6

can be removed as a duplicated object (here, Sample 2 has been removed). After that operation, our dataset will have a similar form to the presented in Table 4.

Now we can define a indiscernibilty relation between objects. Next, we can find out equivalence classes and, after that, we can compute core and reducts. They can be regarded as a kind of a "tip" in restriction of data taken into account in future processing. In this case (Table 4):

$$U|_{IND(A)} = U|_{IND(A-\{1007\_s\_at\})} = U|_{IND(A-\{1053\_at\})} = U|_{IND(A-\{117\_at\})} = U|_{IND(A-\{121\_at\})} = \{\{Specimen\}\{Sample1\}\{Sample3\}\}$$

As shown above, every one from the set of attributes can be removed without causing collapse of the equivalence class structure, therefore in that system Core is en empty set:

$$Core(A) = \emptyset,$$

In next step the reduct(s) - an sufficient set of attributes for representing the category structure has been searched. Because

$$\{\{Specimen\}, \{Sample1\}, \{Sample3\}\} = U|_{IND(A)} = U|_{IND\{121\_at\}} \neq U|_{IND\{1007\_s\_at\}} \neq U|_{IND\{1053\_at\}} \neq U|_{IND\{117\_at\}}\}$$

and $\{121\_at\}$ is minimal therefore

$$Red(A) = \{121\_at\}.$$

It means, that after such discretization and reduct computing, only one attribute - 121_*at* is sufficient to differentiate the species. With this information we can state, that the expression of the gene with ID=121_*at* for all samples (including reference pattern) is different. In that way we can find out all genes which differs from each other. Making selection as „reverse selection" we can obtain all undifferentiated genes. Taking into account the quantity of data (this exemplary discussion is about a few samples), similar procedure, with repetitive decreasing length division during discretization process can effects in obtaining a granular division of coexpressed genes into disjunctive sets. Additionally, just after a few first steps of preparing data (in Table 3, after discretization), we can see, that for this set of samples, fluorescence level for Probeset Id=117_at is the same in all species - so it is not a carrier of useful for us information. Attempts of application of the rest of the rough set theory in oligonucleotide microarray data analysis will be a goal in the future examination.

5 Future Tasks and Problems

For now, many different approaches are used in gene expression analysis - the most common are statistical methods, but one can find out useful some dimensiality reduction methods such as Principal Component Analysis or Projection Pursuit Regression. The other methods are from artificial intelligence field, for example - artificial neural networks (MLP for prediction or self-organizing maps) or genetic algorithms [6,7,9,12]. In this paper is presented an idea to group together coexpressed genes and then, may be after adding other biological information about tissues - make appropriate selection. Presented above exemplary computation of reduct does not reflect complexity of problems in gene expression analysis. Here, in this exemplary fragment from complete microarray reading, only one reduct, with one element appears. Example is not representative for all task - it is rather casual and unprecedented example - in real, the problem will be with a relatively large set of reducts. It may be necessary to make choice of ,,better" and ,,worse" reducts, or - to take into considerations sets of reducts. Problem which reducts it should be - is a separate task. So, yet in the very first step of data processing we have at least three possible and very important sources of errors: the first is image analysis during image processing, the second is error rising form discretization process, and the third - from necessity of selecting a subset (or subsets) of all computed attributes. Next difficulties appears when taking into consideration that gene expression process is a dynamic (expression is changeable during the time)[6], moreover - expression of a one certain gene may be depended on the other gene or other conditions [14,15,18]. Taking into account all mentioned above remarks, solving that problem may require to take into consideration not only rough set theory (to find out the most/least expressed genes or coexpressed genes), but also the data and processes mining [13,16,17,23] (see for remark in footnote at the page with the end of Section 3 of this paper) of oligonucleotide microarray data, in conjunction with biological processes and dysfunction or illness [8,10,11,14,15] for more complete and useful analysis.

However, no matters if results are correct from the point of view of computer scientist, all results of applying rough set theory in oligonucleotide microarray data analysis have to be verified and interpreted in cooperation with life- an biosciences researchers.

6 Conclusions

Initially, rough sets looks to be the helpful tool for some kind of microarray data analysis.

1. We can try to use rough sets theory to reduct quantity of dimensions in microarray data (equivalence classes or reducts for coexpressed genes instead of every one gene)
2. Discretization should rather be done taking into account quantity and fluorescence level of coexpressed genes than setting ,,a priori" boundary of partitioning. Cooperation with bioexperts in this step will be an invaluable help.
3. The problem of quantity, or apropriate choice of computed reducts should be solved (it is possible that decision should be based on groups of reducts).
4. Coexpressed genes, or differentially expressed genes can be found using the rough set theory, but all the time there will be a problem with quantity of data.
5. One can not exclude the noise in data acquisition and diversity of biological tissues (reference patterns and samples), so there can be a problem with boundary width of rough set (it is unchangeable).
6. In fact, gene expression is a dynamic, changeable in time process so it may be necessary to broaden rough sets methods to data and process mining methods.
7. All computed results ought to be verified and interpreted with help of lifean biosciences researchers, this is the only way for verifying usefulness of the method.

References

1. Düntsch, I., Gediga, G.: Rough set data analysis: A road to non-invasive knowledge discovery. Methoδos Publishers, Bangor (2000)
2. http://portalwiedzy.onet.pl/75936,,,,gen,haslo.html
3. http://www.biologia.pl/slowniczek/ekspresja_genu.php3
4. Murray, R.K., Granner, D.K., Mayes, P.A., Rodwell, V.W.: Harper's Illustrated Biochemistry, 26th edn. McGraw-Hill, New York (2003)
5. http://www.affymetrix.com
6. Ao, S.I., Ng, M.K.: Gene expression Time Series Modeling with Principal Component Analysis and Neural Network. In: Soft Comput, 10th edn. pp. 351–358. Springer, Heidelberg (2006) (published online 10 May 2005)
7. Wee-chung Liew, A., Keung Szeto, L., Tang, S.: A Computational Approach to Gene Expression Data Extraction and Analysis. Journal of VLSI Signal Processing 38, 237–258 (2004)
8. Yu, H., Gao, L., Tu, K., Gou, Z.: Broadly predicting specific gene functions with expression similarity and taxonomy similarity. Gene. 352, 75–81 (2005)
9. Shah, S., Kusiak, A.: Cancer Gene Search with Data-mining and Genetic Algorithms. Computers in Biology and Medicine 37, 251–261 (2007)
10. Lu, Y., Han, J.: Cancer classification using gene expression data. Information Systems 28, 243–268 (2003)

11. Jonsson, P., Laurio, K., Lubovac, Z., Olsson, B., Andersson, M.L.: Using Functional annotation to improve clusterings of gene expression patterns. Information Sciences 145, 183–194 (2002)
12. Leng, X., Müller, H.G.: Classification using functional Data Analysis for Temporal Gene Expression Data. Bioinformatics 22(1), 68–76 (2006)
13. Manduchi, E., Grant, G.R., McKenzie, S.E., Overton, G.C., Surrey, S., Stoeckert Jr., C.: Generation of Patterns from Gene Expression Data by Assigning Confidence to Differentially Expressed Genes. Bioinformatics 16(8), 685–698 (2000)
14. Tan, Y., Shi, L., Hussain, S.M., Xu, J., Tong, W., Frazier, J.M., Wang, C.: Integrating time-course microarray gene expression profiles with cytotoxicity for identification of biomarkers in primary rat hepatocytes exposed to cadmium. Bioinformatics 22(1), 77–87 (2006)
15. Wilczok, A.: Cytotoxicity of quinoline derivatives and metaloorganic complexes of cobalt and iron analyzed by transcriptional activity of genes associated with proliferation, apoptosis and angiogenesis in human tumor cell culture lines. In: polish: Cytotoksyczność pochodnych chinoliny oraz metaloorganicznych kompleksów kobaltu i żelaza a aktywność transkrypcyjna genów w procesach proliferacji, apoptozy i angiogenezy w hodowlach komórek ludzkich linii nowotworowych. Habilitation 33/2006, p. 192. Wydawnictwo Śląskiej Akademii Medycznej, Katowice (2006)
16. Günther, C.W., Rinderle, S., Reichert, M., van der Aalst, W.M.P.: Using Process Mining to Learn from Process Changes in Evolutionary Systems. Eindhoven University of Technology, Eindhoven (2006)
17. Cook, J.E., Wolf, A.L.: Discovering models of software processes from event-based data. ACM Transactions on Software Engineering and Methodology 7(3), 215–249 (1998)
18. Opgen-Rhein, R., Strimmer, K.: Inferring Gene Dependency Networks from Genomic Longitudinal Data: A Functional Data Approach. Statistical Journal 4(1), 53–65 (2006)
19. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data, System Theory. In: Pawlak, Z. (ed.) Knowledge Engineering and Problem Solving 9, Kluwer Academic Publishers, Dordrecht (1991)
20. Pawlak, Z., Skowron, A.: Rudiments of rough sets.Information Sciences. An International Journal 177(1), 3–27 (2007)
21. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences. An International Journal 177(1), 28–40 (2007)
22. Pawlak, Z., Skowron, A.: Rough sets and Boolean reasoning. Information Sciences. An International Journal 177(1), 41–73 (2007)
23. Pawlak, Z.: Concurrent versus sequential the rough sets perspective. Bulletin of the EATCS (48), 178–190 (1992)
24. Skowron, A., Suraj, Z.: Rough sets and concurrency. Bulletin of the Polish Academy of Sciences 41(3), 237–254 (1993)
25. Swiniarski, R., Skowron, A.: Rough set methods in feature selection and extraction. Pattern Recognition Letters 24(6), 833–849 (2003)
26. Suraj, Z.: Rough set methods for the synthesis and analysis of concurrent processes. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.) Rough Set Methods and Applications (Studies in Fuzziness and Soft Computing 56), pp. 379–488. Physica-Verlag, Heidelberg (2000)

From an Information System to a Decision Support System

Alicja Wakulicz–Deja and Agnieszka Nowak

Institute of Computer Science, University of Silesia
Będzińska 39, 41–200 Sosnowiec, Poland
{nowak,wakulicz}@us.edu.pl

Abstract. In the paper we present the definition of *Pawlak's* model of an information system. The model covers information systems with history, systems with the decomposition of objects or attributes and dynamical information systems. Information systems are closely related to rough set theory and decision support systems. The paper demonstrates how the results of *Pawlak's* research can be applied in the artificial intelligence domain.

1 Introduction

The notion of an information system formulated by *Pawlak* and developed with his co-workers, is now a well developed branch of data analysis formalisms. It is strongly related to (but different from) the relational database theory on the one hand and to fuzzy set theory on the other. In this paper we consider the connection between the theory of information and information retrieval systems with rough set theory and decision support systems. It is obvious that model of a system created by *Pawlak* makes data description and analysis simple and very reliable.

2 Information System

An information system consists of a set of objects and attributes defined on this set. In information systems with a finite number of attributes, there are classes created by these attributes (for each class, the values of the attributes are constant on elements from the class). Any collection of data, specified as a structure $S = \langle X, A, V, q \rangle$ such that X is a non-empty set of objects, A is a non-empty set of attributes, V is a non-empty set of values: $V = \bigcup_{a \in A} V_a$ and q is an *information function* of $X \times A \rightarrow V$, is referred to as an information system. The set $\{q(x, a) : a \in A\}$ is called information about the object x or, in short, a record of x or a row determined by x. Each attribute a is viewed as a mapping $a : X \rightarrow V_a$ which assigns a value $a(x) \in V_a$ to every object x. A pair (a, v), where $a \in A$, and $v \in V_a$, is called a *descriptor*. In information systems, the descriptor language is a formal language commonly used to express and describe properties of objects and concepts. More formally, an information system is a pair

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 454–464, 2007.

$\mathcal{A} = (U, A)$ where U is a non-empty finite set of objects called the *universe* and A is a non-empty finite set of attributes such that $a : U \rightarrow V_a$ for every $a \in A$. The set V_a is called the *value set* of a. Now we will discuss which sets of objects can be expressed (defined) by formulas constructed by using attributes and their values. The simplest formulas d, called *descriptors*, have the form (a, v) where $a \in A$ and $v \in V_a$. In each information system S the information language $L_S = \langle AL, G \rangle$ is defined, where AL is the alphabet and G is the grammar part of that language. AL is simply a set of all symbols which can be used to describe the information in such a system, e.g.: $\{0, 1\}$ (constant symbols), A - the set of all attributes, V - a set of all the values of the attributes, symbols of logical operations like $\tilde{}$, $+$ and $*$, and naturally brackets, which are required to represent more complex information. G - the grammar part of the language L_S defines syntax with T_S as the set of all possible forms of terms (a *term* is a unit of information in S) and its meaning (semantics). A simple descriptor $(a, v) \in T_S (a \in A$ where $v \in V_a)$. If we denote such a descriptor (a, v) as the term t, then following term formations will be also possible: $\neg t, t + t', t * t'$, where $t, t' \in T_S$. The meaning is defined as a function σ which maps the set of terms in a system S in a set of objects X, $\sigma : T_S \rightarrow P(x)$, where $P(x)$ is the set of the subsets of X. The value of σ for a given descriptor (a, v) is defined as following: $\sigma(a, v) = \{x \in X, q_x(a) = v\}$, $\sigma(\neg t) = X \setminus \sigma(t)$, $\sigma(t + t') = \sigma(t) \bigcup \sigma(t')$ and $\sigma(t * t') = \sigma(t) \bigcap \sigma(t')$ [18].

2.1 Information Table

Information systems are often represented in a form of tables with the first column containing objects and the remaining columns, separated by vertical lines, containing values of attributes. Such tables are called **information tables** (an example is presented in Table 1). The definition of this system is as follows:

Table 1. An information system - an information table

student	a	b	c
x_1	a_1	b_1	c_1
x_2	a_1	b_1	c_2
x_3	a_2	b_1	c_3
x_4	a_2	b_1	c_4
x_5	a_1	b_2	c_1
x_6	a_1	b_2	c_2
x_7	a_2	b_2	c_3
x_8	a_2	b_2	c_4

$S = \langle X, A, V, q \rangle$, where $X = \{x_1, \ldots, x_8\}$, $A = \{a, b, c\}$, $V = V_a \cup V_b \cup V_c$, $V_a = \{a_1, a_2\}$, $V_b = \{b_1, b_2\}$, $V_c = \{c_1, c_2, c_3, c_4\}$ and $q : X \times A \rightarrow V$. For instance, $q(x_1, a) = a_1$ and $q(x_3, b) = b_1$. Before we start considering the properties of an information system, it is necessary to explain what the *information* in such a system means. The information in the system S is a function ρ with the arguments on the attributes set A and its values, which belong to the set V $(\rho(a) \in V_a)$. As long as the sets of the objects, attributes and their values are finite, we know exactly how many (different) pieces of information in a given system S comprises, and the number is equal to $\prod_{a \in A} card(V_a)$. The information ρ assigns a set of the objects X_ρ that $X_\rho = \{x \in X : q_x = \rho\}$. We call them *indiscernible*, because they have the same description. If we assume that $B \subseteq A$ then each subset B of A determines a binary relation $IND_A(B)$, called an *indiscernibility relation.* By the *indiscernibility relation* determined by B, denoted by $IND_A(B)$, we understand the equivalence relation $IND_A(B) = \{\langle x, x^{'} \rangle \in X \times X : \forall_{a \in B}[a(x) = a(x^{'})]\}$.

For a given information system it is possible to define the comparison of the objects, attributes and even the whole systems. We can find some dependent and independent attributes in data, we can check whether the attributes or even objects are equivalent. An important issue in data analysis is to discover dependencies between attributes. Intuitively, a set of attributes D depends totally on a set of attributes C if the values of attributes from C uniquely determine the values of the attributes from D. If D depends totally on C then $IND_A(C) \subseteq IND_A(D)$. This means that the partition generated by C is finer than the partition generated by D.

Assume that a and b are attributes from the set A in a system S. We say that b depends on a $(a \rightarrow b)$, if the indiscernibility relation on a contains in the indiscernibility relation on b: $\widetilde{a} \subseteq \widetilde{b}$. If $\widetilde{a} = \widetilde{b}$ then the attributes are equivalent. The attributes are dependent if any of the conditions: $\widetilde{a} \subseteq \widetilde{b}$ or $\widetilde{b} \subseteq \widetilde{a}$ is satisfied. Two objects $x, y \in X$ are *indiscernible* in a system S relatively to the attribute $a \in A$ $(x_{\widetilde{a}}y)$ if and only if $q_x(a) = q_y(a)$. In the presented example, the objects x_1 and x_2 are indiscernible relatively to the attributes a and b. The objects $x, y \in X$ are *indiscernible* in a system S relatively to all of the attributes $a \in A$ $(x_{\widetilde{S}}y)$ if and only if $q_x = q_y$. In the example there are no indiscernible objects in the system S. Each information system determines unequivocally a partition of the set of objects, which is some kind of classification. Finding the dependence between attributes let us to reduce the amount of the information which is crucial in systems with a huge numbers of attributes. Defining a system as a set of objects, attributes and their values is necessary to define the algorithm for searching the system and updating the data consisted in it. Moreover, all information retrieval systems are also required to be implemented in this way. The ability to discern between perceived objects is also important for constructing various entities not only to form reducts, but also decision rules and decision algorithms.

2.2 An Application in Information Retrieval Area

The information retrieval issue is the main area of the employment of information systems. An information retrieval system, in which the objects are described by their features (properties), we can define as follows: Let us have a set of objects X and a set of attributes A. These objects can be books, magazines, people, etc. The attributes are used to define the properties of the objects. For the system of books, the attributes can be *author*, *year*, *number of sheets.* An information system which is used for information retrieval should allow to find the answer for a query. There are different methods of retrieving information. Professor *Pawlak* proposed the *atomic components method* [2,18]. Its mathematical foundation was defined in [5] and [6]. This method bases on the assumption that each question can be presented in the normal form, which is the sum of the products with one descriptor of each attribute only. To make the system capable of retrieving information it is required to create the information language (*query language*). This language should permit describing objects and forming user's queries. Naturally enough, such a language has to be universal for both the natural and system language. Owing to this, all steps are done on the language level rather than on the database level. The advantages of information languages are not limited to the aforementioned features. There are a lot of systems that need to divide the information, which is called the decomposition of the system. It allows improving the time efficiency and make the updating process easy, but also enables the organization of the information in the systems. Information systems allow collecting data in a long term. It means that some information changes in time, and because of that, the system has a special property, which is called the dynamics of the system. Matching unstructured, natural-language queries and documents is difficult because both queries and documents (objects) must be represented in a suitable way. Most often, it is a set of terms, where a*term* is a unit of a semantic expression, e.g. a word or a phrase. Before a retrieval process can start, sentences are preprocessed with stemming and removing too frequent words (stopwords).

2.3 System with Decomposition

When the system consists of huge set of data it is very difficult in given time to analyse those data. Instead of that, it is better to analyze the smaller pieces (subsets) of data, and at the end of the analysing, connect them to one major system. There are two main method of decomposition: with *attributes* or *objects*. A lot of systems are implemented with such type of decomposition.

System with object's decomposition. If it is possible to decompose the system $S = \langle X, A, V, q \rangle$ in a way that we gain subsystems with smaller number of objects, it means that $S = \bigcup_{i=1}^{n} S_i$, where $S_i = \langle X_i, A, V, q_i \rangle$, $X_i \subseteq X$ and $\bigcup_i X_i = X$, $q_i : X_i \times A \to V$, $q_i = q|_{X_i}$.

System with attributes's decomposition. When in system S there are often the same types of queries, about the same group of attributes, it means that such system should be divided to subsystems S_i in a way that: $S = \bigcup_i S_i$, where $S_i = \langle X, A_i, V_i, q_i \rangle$, $A_i \subseteq A$ and $\bigcup_i A_i = A$, $V_i \subseteq V$, $q_i : X \times A_i \to V_i$, $q_i = q|_{X \times A_i}$. Decomposition lets for optimization of the retrieval information process in the system S. The choice between those two kind of decomposition depends only on the type and main goal of such system.

2.4 Dynamic Information System and System with the History

In the literature information systems are classified according to their purposes: documentational, medical or management information systems. We propose different classification: those with respect to dynamics of systems. Such a classification gives possibility to:

1. Perform a joint analysis of systems belonging to the same class,
2. Distinguish basic mechanisms occuring in each class of systems,
3. Unify design techniques for all systems of a given class,
4. Simplify the teaching of system operation and system design principles.

Analysing the performance of information systems, it is easy to see that the data stored in those systems are subject to changes. Those changes occur in definite moments of time. For example: in a system which contains personal data: age, address, education, the values of these attributes may be changed. Thus time is a parameter determining the state of the system, although it does not appear in the system in an explicit way. There are systems in which data do not change in time, at least during a given period of time. But there are also systems in which changes occur permanently in a determined or quite accidental way.

In order to describe the classification, which we are going to propose, we introduce the notion of a *dynamic* information system, being an extension of the notion of an information system presented by *Pawlak*.

Definition 1. *A dynamic information system is a family of ordered quadruples:*

$$S = \{\langle X_t, A_t, V_t, q_t \rangle\}_{t \in T}$$

where:

- T *- is the discrete set of time moments, denoted by numbers* $0, 1, \ldots, N$*,*
- X_t *- is the set of objects at the moment* $t \in T$*,*
- A_t *- is the set of attributes at the moment* $t \in T$*,*
- $V_t(a)$ *- is the set of values of the attribute* $a \in A_t$*,*
- $V_t := \bigcup_{a \in A_t} V_t(a)$ *- is the set of attribute values at the moment* $t \in T$*,*
- q_t *- is a function which assigns to each pair* $\langle x, a \rangle$*,* $x \in X_t$*,* $a \in A_t$*, an element of the set* V_t*, i.e.* $q_t : X_t \times A_t \to V_t$*.*

An ordered pair $\langle a, v \rangle$*,* $a \in A_t$*,* $v \in V_t(a)$ *is denoted as a descriptor of the attribute* a*. We will denote by* $q_{t,x}$ *a map defined as follows:*

$$q_{t,x} : A_t \to V_t,$$

$$\forall_{a \in A_t} \forall_{x \in X_t} \forall_{t \in T} q_{t,x}(a) := q_t(a, x)$$

Definition 2. $Inf(S) = \{V_t^{A_t}\}_{t \in T}$ *is a set of all functions from* A_t *to* V_t *for all* $t \in T$*. Functions belonging to* $Inf(S)$ *will be called informations at instant* t*, similarly, the functions* $q_{t,x}$ *will be called the information about object* x *at instant* t *in information system* S*. Therefore, an information about an object* x *at instant* t *is nothing else, but a description of object* x*, in instant* t*, obtained by means of descriptors.*

Any dynamic system belongs to one of two classes of systems: **time-invariant** and **time-varying** system.

Any of systems like librarian e.g., where operations such as removing some books, or add the new one, are time-invariant. The example of the time-varying system is the system with informations about students, where some informations about them (some attributes) depend on another one, and change the value in some case. In the most situations of practice this model is more convenient then the classical (relational) model. It is due to the fact that in Pawlak's model, information about an object are given by functions, while in the classical model informations are determined by relations. This simplifies a description of systems and their analysis, which is important not only for system designing but also for teaching of system operation. The model of information system created by *Pawlak* is very useful to built and analysis in different types of retrieval information systems. The document information systems are very specific type of information systems and *Pawlak*'s model is very good to define the informations in it.

3 Decision Support Systems

When data mining first appeared, several disciplines related to data analysis, like statistics or artificial intelligence were combined towards a new topic: extracting significant patterns from data. The original data sources were small datasets and, therefore, traditional machine learning techniques were the most common tools for this tasks. As the volume of data grows these traditional methods were reviewed and extended with the knowledge from experts working on the field of data management and databases. Because of that, information systems with some data-mining methods start to be the decision support systems. Decision support system is a kind of information system, which classifies each object to some class denoted by one of the attributes, called *decision* attribute. While the information system is simply a pair of the form U and A, the *decision support system* is also a pair $\mathcal{A} = (U, A \cup \{d\})$ with distinguished attribute d. In case of decision table the attributes belonging to A are called *conditional attributes* or simply *conditions* while d is called *decision*. We will further assume that the set of decision values is finite. The i-th *decision class* is a set of objects $C_i = \{x \in U : d(x) = d_i\}$, where d_i is the i-th decision value taken from decision value set $V_d = \{d_1, \ldots, d_{|V_d|}\}$. Let us consider the decision table presented at Table 2. Having indiscernibility relation we may define the notion of reduct. In case of decision tables *decision reduct* is a set $B \subset A$ of attributes, which cannot be further reduced and $IND(B) \subseteq IND(d)$. *Decision rule* is a formula of the form $(a_{i_1} = v_1) \wedge \ldots \wedge (a_{i_k} = v_k) \Rightarrow (d = v_d)$, where $1 \leq i_1 < \ldots < i_k \leq m, v_j \in V_{a_ij}$. We can simply interpret such formula similar to natural language with *if* and *then* elements. In given decision table the decision rule for object x_1 is given as: *if* $(a = a_1)$ *and* $(b = b_1)$ *and* $(c = c_1)$ *then* $(d = T)$, the same as $(a = a_1) \wedge (b = b_1) \wedge (c = c_1) \rightarrow (d = T)$. Atomic subformulas $(a_{i_1} = v_1)$ are called *conditions, premises.* We say that rule r is applicable to object, or alternatively, the object

Table 2. Decision table

student	a	b	c	d
x_1	a_1	b_1	c_1	T
x_2	a_1	b_1	c_2	T
x_3	a_2	b_1	c_3	T
x_4	a_2	b_1	c_4	N
x_5	a_1	b_2	c_1	N
x_6	a_1	b_2	c_2	T
x_7	a_2	b_2	c_3	T
x_8	a_2	b_2	c_4	N

matches rule, if its attribute values satisfy the premise of the rule. Each object x in a decision table determines a decision rule, $\forall_{a \in C}(a = a(x)) \Rightarrow (d = d(x)))$, where C is set of conditional attributes and d is decision attribute. Decision rules corresponding to some objects can have the same condition parts but different decision parts. We use decision rules to classify given information. When the information is uncertain or just incomplete there is need to use some additional techniques for information systems. Numerous methods based on the rough set approach combined with Boolean reasoning techniques have been developed for decision rule generation.

4 Rough Sets

Rough Set theory has been applied in such fields as machine learning, data mining, etc., successfully since Professor *Pawlak* developed it in 1982. Reduction of decision table is one of the key problem of rough set theory. The methodology is concerned with the classificatory analysis of imprecise, uncertain or incomplete information or knowledge expressed in terms of data acquired from experience. The primary notions of the theory of rough sets are the approximation space and lower and upper approximations of a set. The approximation space is a classification of the domain of interest into disjoint categories. The membership status with respect to an arbitrary subset of the domain may not always be clearly definable. This fact leads to the definition of a set in terms of lower and upper approximations [8,9,10].

4.1 Lower/Upper Approximation

The *lower approximation* is a description of the domain objects which are known with certainty to belong to the subset of interest, whereas the *upper approximation* is a description of the objects which possibly belong to the subset. Any subset defined through its lower and upper approximations is called a rough set. It must be emphasized that the concept of rough set should not be confused with the idea of fuzzy set as they are fundamentally different, although in some sense complementary, notions. Rough set approach allows to precisely define the notion of concept approximation. It is based on the *indiscernibility relation* between objects defining a partition of the universe U of objects. The indiscernibility of objects follows from the fact that they are perceived by means of values of available attributes. Hence some objects having the same (or similar) values of attributes are indiscernible. Let $\mathcal{A} = (U, A)$ be an information system, then with any $B \subseteq A$ there is associated an equivalence relation $IND_{\mathcal{A}}(B)$, called the *B-indiscernibility relation*, its classes are denoted by $[x]_B$. For $B \subseteq A$ and $X \subseteq U$, we can approximate X using only the information contained in B by constructing the *B-lower*$(\underline{BX})$ and *B-upper approximations of* $X(\overline{BX})$, where $\underline{BX} = \{x : [x]_B \subseteq X\}$ and $\overline{BX} = \{x : [x]_B \bigcap X \neq \emptyset\}$. The B-lower approximation of X is the set of all objects which can be certainly classified to X using attributes from B. In the Rough set area there is also a very important problem with finding (select) relevant features (attributes), which source is denoted as so called *core* of the information system $\mathcal{A}$. *Reduct* is a minimal set of attributes $B \subseteq A$ such that $IND_{\mathcal{A}}(B) = IND_{\mathcal{A}}(A)$, which means that it is a minimal set of attributes from A that preserves the original classification defined by the set A of attributes.The intersection of all reducts is the so-called *core*. In the example both the *core* and the reduct consist of attributes b and c ($CORE(C) = \{b, c\}$, $RED(C) = \{b, c\}$).

4.2 Rule Induction

Rough set based rule induction methods have been applied to knowledge discovery in databases, whose empirical results obtained show that they are very powerful and that some important knowledge has been extracted from databases. For rule induction, lower/upper approximations and reducts play important roles and the approximations can be extended to variable precision model, using accuracy and coverage for rule induction have never been discussed. We can use the *indiscernibility function* $f_{\mathcal{A}}$, that form a minimal decision rule for given decision table [1]. For an information system $\mathcal{A} = (U, A \cup \{d\})$ with n objects, the *discernibility matrix* of $\mathcal{A}$ is a symmetric $n \times n$ matrix with entries c_{ij} defined as $c_{ij} = \{a \in A | a(x_i) \neq a(x_j)\}$ for $i, j = 1, 2, \ldots, n$ where $d(x_i) \neq d(x_j)$). Each entry consists of the set of attributes upon which objects x_i and x_j differ. A *discernibility function* $f_{\mathcal{A}}$ for an information system $\mathcal{A}$ is a boolean function of m boolean variables $a_1^*, \ldots, a_m^*$ (corresponding to the attributes $a_1, \ldots, a_m$) defined by:

$$f_{\mathcal{A}} = \bigwedge \left\{ \bigvee c_{ij}^* | 1 \leq j \leq i \leq n, c_{ij} \neq \emptyset \right\}$$

where $c_{ij}^* = \{a^* | a \in c_{ij}\}$. For given decision table we formed following set of rules:

- rule nr 1: *if* $a = a_1$ *and* $b = b_1$*then* $d = T$
- rule nr 2: *if* $b = b_1$ *and* $c = c_1$*then* $d = T$
- rule nr 3: *if* $b = b_1$ *and* $c = c_2$*then* $d = T$
- rule nr 4: *if* $c = c_3$ *then* $d = T$
- rule nr 5: *if* $c = c_4$ *then* $d = N$
- rule nr 6: *if* $b = b_2$ *and* $c = c_1$*then* $d = N$
- rule nr 7: *if* $c = c_2$ *then* $d = T$

4.3 Rough Set Theory and Decision Systems in Practise

The main specific problems addressed by the theory of rough sets are not only representation of uncertain or imprecise knowledge, or knowledge acquisition from experience, but also the analysis of conflicts, the identification and evaluation of data dependencies and the reduction of the amount of information. A number of practical applications employing this approach have been developed in recent years in areas such as medicine, drug research, process control and other. The recent publication of a monograph on the theory and a handbook on applications facilitate the development of new applications. One of the primary applications of rough sets in AI is knowledge analysis and data mining [11,12,15,16]. >From two expert systems implemented at the Silesian University, MEM is the one with the decision table in the form of the knowledge base. It is a diagnosis support system used in child neurology and it is a notable example of a complex multistage diagnosis process. It permits the reduction of attributes, which allows improving the rules acquired by the system. MEM was developed on the basis of real data provided by the Second Clinic of the Department of Paediatrics of the Silesian Academy of Medicine. The system is employed there to support the classification of children having mitochondrial encephalopathies and considerably reduces the number of children directed for further invasive testing in the consecutive stages of the diagnosis process [17,19]. The first stages of research on decision support systems concentrated on: methods to represent the knowledge in a given system and the methods of the verification and validation of a knowledge base [13]. Recent works, however, deal with the following problems: a huge number of rules in a knowledge base with numerous premises in each rule, a large set of attributes, many of which are dependent, complex inference processes and the problem of the proper interpretation of the decision rules by users. Fortunately, the cluster analysis brings very useful techniques for the smart organisation of the rules, one of which is a hierarchical structure. It is based on the assumption that rules that are similar can be placed in one group. Consequently, in each inference process we can find the most similar group and obtain the forward chaining procedure on this, significantly smaller, group only. The method reduces the time consumption of all processes and explores only the new facts that are actually necessary rather then all facts that can be retrieved from a given knowledge base. In our opinion, clustering rules for inference processes in decision support systems could prove useful to improve the efficiency of those systems[3,4]. Rough sets theory enables solving the problem of a huge number of attributes and dependent attributes removal. The accuracy of classification can be increased by selecting subsets of strong attributes, which is performed by using several classification learners. The processed data are classified by diverse learning schemes and the generation of rules is supervised by domain experts. The implementation of this method in automated decision support software can improve the accuracy and reduce the time consumption as compared to full syntax analysis[20,21].

5 Summary

Information systems and decision support systems are strongly related. The paper shows that we can treat a decision system as an information system of some objects, for which we have the information about their classification. When the information is not complete, or the system has some uncertain data - we can use rough set theory to separate the uncertain part from that, what we are sure about. By defining the *reduct* for a decision table, we can optimize the system and then, using the methods for minimal rules generation, we can easily classify new objects. We see, therefore, that *Prof. Pawlak's* contribution to the domain of information and decision support systems is invaluable.

References

1. Bazan, J.: Metody wnioskowań aproksymacyjnych dla syntezy algorytmów decyzyjnych, praca doktorska, Wydział Informatyki, Matematyki i Mechaniki, Uniwersytet Warszawski, Warszawa (1998)
2. Grzelak, K., Kochańska, J.: System wyszukiwania informacji metodą składowych atomowych MSAWYSZ, ICS PAS Reports No 511, Warsaw (1983)
3. Nowak, A., Wakulicz-Deja, A.: Effectiveness comparison of classification rules based on k-means Clustering and Salton's Metod. In: Advances in Soft Computing, pp. 333–338. Springer, Heidelberg (2004)
4. Nowak, A., Wakulicz-Deja, A.: The concept of the hierarchical clustering algorithms for rules based systems. In: Advances in Soft Computing, pp. 565–570. Springer, Heidelberg (2005)
5. Pawlak, Z.: Mathematical foundation of information retrieval. CC PAS Reports No 101, Warsaw (1973)
6. Pawlak, Z., Marek, W.: Information Storage and retrieval system-mathematical foundations. CC PAS Reports No 149, Warsaw (1974)
7. Pawlak, Z.: Rough Sets: Theoretical aspects of reasoning about data. Kluwer Academic Publishers, Boston (1991)
8. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177, 3–27 (2007)
9. Pawlak, Z., Skowron, A.: Rough sets: some extensions. Information Sciences 177, 28–40 (2007)
10. Pawlak, Z., Skowron, A.: Rough sets and Boolean reasoning. Information Sciences 177, 41–73 (2007)
11. Roddick, J.F., Hornsby, K., Spiliopoulou, M.: YABTSSTDMR - Yet Another Bibliography of Temporal, Spatial and Spatio-Temporal Data Mining Research. In: Unnikrishnan, K.P., Uthurusamy, R. (eds.) Proc. SIGKDD Temporal Data Mining Workshop, San Francisco, CA, pp. 167–175. ACM, New York (2001)
12. Roddick, J.F., Egenhofer, M.J., Hoel, E., Papadias, D., Salzberg, B.: Spatial, Temporal and Spatio-Temporal Databases - Hot Issues and Directions for PhD Research. SIGMOD Record 33(2), 126–131 (2004)
13. Simiński, R., Wakulicz-Deja, A.: Circularity in Rule Knowledge Bases - Detection using Decision Unit Approach. In: Advances in Soft Computing, pp. 273–280. Springer, Heidelberg (2004)

14. Skowron, A., Grzymała-Busse, J.: From the Rough Set Theory to the Evidence Theory. In: Yager, R.R., Fedrizzi, M., Kacprzyk, J. (eds.) Advances in the Dempster-Shafer Theory of Evidence, pp. 193–236. Wiley, New York (1994)
15. Skowron, A., Bazan, J., Stepaniuk, J.: Modelling Complex Patterns by Information Systems. Fundamenta Informaticae 67(1-3), 203–217 (2005)
16. Bazan, J., Peters, J., Skowron, A., Synak, P.: Spatio-temporal approximate reasoning over complex objects. Fundamenta Informaticae 67, 249–269 (2005)
17. Wakulicz-Deja, A.: Podstawy systemów ekspertowych. Zagadnienia implementacji. Studia Informatica 26(3), 64 (2005)
18. Wakulicz-Deja, A.: Podstawy systemów wyszukiwania informacji, Akademicka Oficyna Wydawnicza (1995)
19. Wakulicz-Deja, A., Paszek, P.: Optimalization on Decision Problems on Medical Knowledge Bases, 5th European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany (1997)
20. Wakulicz-Deja, A., Ilczuk, G.: Attribute Selection and Rule Generation Techniques for Medical Diagnosis Systems. In: Ślęzak, D., Yao, J., Peters, J.F., Ziarko, W., Hu, X. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3642, Springer, Heidelberg (2005)
21. Wakulicz-Deja, A., Paszek, P.: Applying rough set theory to multi stage medical diagnosing. Fundamenta Informaticae XX, 1–22 (2003)

Optimization for MASK Scheme in Privacy Preserving Data Mining for Association Rules*

Piotr Andruszkiewicz

Institute of Computer Science, Warsaw University of Technology, Poland
P.Andruszkiewicz@ii.pw.edu.pl

Abstract. As a result of advances in technology, large amounts of data can be collected and stored automatically. Significant development of the Internet and easier access to it have contributed to collecting large amounts of information about users' characteristics. Along with these changes, concerns about privacy of data have emerged. Several methods of preserving privacy for association rules mining have been proposed in literature: MASK scheme and its optimizations. This paper provides new solutions concerning efficiency for this scheme and considers different methods of distorting data using randomization techniques. Effectiveness of these solutions has been tested and presented in this paper.

1 Introduction

Concerns about privacy of information provided by users and collected to discover hidden knowledge lead to inaccuracy of this data. Users are afraid of revealing sensitive data. The cause to provide wrong data may be the worry that the provided data can be misused.

Considering these concerns several methods of preserving privacy for association rules mining have been proposed. The goal of preserving privacy is to encourage people to provide true information, even about sensitive values.

People have different concerns about different items (attributes). This regularity can be used to obtain higher accuracy. Thus, several solutions have been proposed to take the advantage of it.

Incorporating privacy in association rule mining in MASK (Mining Associations with Secrecy Konstraints[1]) [8] scheme results in time cost. Several optimizations of MASK were proposed, but no optimization could have broken the exponential complexity of reconstructing the original support of a set based on the distorted database in general case.

In this paper, we present a new optimization, called MMASK (Modified MASK), which breaks the exponential complexity and achieves better results for higher values of privacy.

Effectiveness and accuracy of MMASK have been tested on synthetic and real data sets and compared to Apriori and original MASK scheme.

* Research has been supported by grant No 3 T11C 002 29 received from Polish Ministry of Education and Science.

[1] Authors use *Konstraints* instead of *Constraints* to achieve abbreviation: MASK.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 465–474, 2007.

1.1 Related Work

Privacy Preserving Data Mining in association rules mining has been widely discussed recently [8] [15] [3] [4] [9] [12] [6] [5].

Proposals presented in [15] [3] [4] [9] show how to prevent the *sensitive rules* from discovering by the miner. The solution is to falsify some tuples or replace original values with unknowns. It is complementary to our work, because it preserves privacy on aggregate level, whereas our solution preserves privacy for individual values. Second difference is that this issue requires completely materialized true database as the starting point whereas our problem does not require collecting of the original data in the centralized database.

Privacy preserving for individual values in distributed data is considered in [12] [6]. In these works databases are distributed across a number of sites and each site only willing to share mining process results, but does not want to reveal the source data. Techniques for distributed database require a corresponding part of the true database at each site. Our approach collects only modified tuples, which are distorted at the client machine.

A framework for mining association rules from centralized distorted database was proposed in [8]. A scheme called MASK attempts to simultaneously provide a high degree of privacy to the user and retain a high degree of accuracy in the mining results. To address efficiency several optimizations for MASK were originally proposed in [8]. The main optimization, which reduces the number of counters to linear, requires randomization factors (i.e., probability of flipping) to be constant for all items. This is the most important disadvantage of this optimization, because it does not allow using different randomization factors for different items what helps to achieve higher accuracy [14].

Another optimization, called EMASK, was proposed in [2]. It is a powerful optimization by which the entire extra overhead of counting all the combinations generated by the distortion can be eliminated. It can be used in MMASK scheme. In general case EMASK does not break exponential complexity of reconstructing the original support.

In [14] a general framework for privacy preserving association rule mining was proposed. It allows attributes to be randomized using different randomization factors, based on their privacy levels. It was theoretically proven that the use of non-uniform randomization factors can lead to more accurate mining results than the use of one unique conservative randomization factor. Empirical experiment results also verified this claim. An efficient algorithm RE (Recursive Estimation) to mine frequent itemsets under this framework was developed also. RE algorithm uses different randomization factors, but it does not break the exponential complexity in estimating the support.

1.2 Contributions of This Paper

The new optimization proposed in this paper breaks 2^n complexity in estimating n-itemset support, which was not achieved by any other presented optimization for MASK scheme which allows every attribute to have his own randomization factors for 0's and 1's.

1.3 Organization of This Paper

The remainder of this paper is organized as follows: In Section 2, we present basic information about MASK scheme and in Section 3, we describe methods of distorting the data. Then, in Section 4, we present our new optimization of MASK algorithm for mining the distorted database. The experimental results are highlighted in Sect. 5. Finally, in Section 6, we summarize the conclusions of our study and outline future avenues to explore.

2 Basic Revision of MASK Scheme

In this Section we present basic information about the MASK algorithm for Privacy Preserving Data Mining [8].

2.1 Original Distortion Procedure in MASK Scheme

Here we present a basic randomization method for distorting the transactional data, which was used in original MASK scheme. For information about other modification methods used in Privacy Preserving Data Mining see [13].

Given a tuple which contains 0's and 1's, each item is kept with probability p or flipped with probability $1-p$ [2]. All tuples are distorted in the same manner. Distorted tuples which create a new database are supplied to the miner. Only information the miner gets is the distorted database and the value of probability p.

The randomization operator used in the scheme presented above is *item-invariant* - the distortion process is applied to each item in the transaction independently. Presented randomization is also a *per-transaction* randomization - the distortion process for transaction T_i does not use any information about transaction T_j, where $i \neq j$. Both definitions of *item-invariant* operator as well as *per-transaction* randomization can be found in [5].

Having a *per-transaction* randomization is a huge advantage, because it makes possible to distort true data (a tuple which is a customer's answer) on the client side. Collecting all the true data (all customers' tuples) is not necessary. An application on the client side distorts a tuple and sends only distorted values to the central collector. Second advantage is that additional distorted tuples can be added to the central database at any time and the data mining process repeated over the whole collected data.

2.2 Estimating n-itemset Support

Let T refer to the $True$ matrix - true dataset[2]. We denote the $Distorted$ database, obtained accordingly to the distortion procedure presented in Sect. 2.1, as D.

C_k^T, respectively C_k^D, is the number of tuples in T, respectively D, database that have binary form of k (in n digits) for the given itemset. For a 2-itemset C_0^T refers to the number of 00's and C_2^T to the number of 10's.

[2] In real applications the true database is not stored. Only distorted tuples are collected.

$$\mathbf{C}^D = \begin{bmatrix} C^D_{2^n-1} \\ \vdots \\ C^D_1 \\ C^D_0 \end{bmatrix}, \quad \mathbf{C}^T = \begin{bmatrix} C^T_{2^n-1} \\ \vdots \\ C^T_1 \\ C^T_0 \end{bmatrix}$$

The matrix **M** is defined as follows:
$m_{i,j}$ - the probability that a tuple of form C^T_j in matrix T goes to a tuple of form C^D_i in D.

$$\mathbf{M} = \begin{bmatrix} m_{0,0} & m_{0,1} & m_{0,2} & \cdots & m_{0,2^n-1} \\ m_{1,0} & m_{1,1} & m_{1,2} & \cdots & m_{1,2^n-1} \\ \vdots & & & \ddots & \vdots \\ m_{2^n-1,0} & m_{2^n-1,1} & m_{2^n-1,2} & \cdots & m_{2^n-1,2^n-1} \end{bmatrix}$$

The support of the item in the true database T based upon the support of this item in D can be estimated using the following equation:

$$\mathbf{C}^T = \mathbf{M}^{-1}\mathbf{C}^D \tag{1}$$

In general (without any assumptions about values of p, that is the value of p is the same for all items), MASK scheme needs an exponential number of counters (2^n counters for an n-itemset) and makes the process infeasible in practice [2].

3 Distorting the Data

Distorting method discussed in Sect. 2.1 is the one of the simplest randomization methods. It assumes the same value of p for all attributes.

In the more general scenario we assume that each item has a different probability which is used to decide whether to change the value (0 or 1) or not while conducting the distortion process. Then we have $p_1, p_2, ..., p_k$ parameters, where k is the number of different items in the database. Let denote these parameters as vector $\mathbf{P} = P_i$, such that $P_i = p_i$.

Randomization factors in range 0.7 - 0.9 make the number of 1's to increase in the database. As a result of this growth the time of performing a data mining process increases [2].

As the distortion is a culprit, a method with different randomization factors for 0's and 1's can be used to avoid the growth of processing time.

Let p_i denote for a given attribute the randomization factor for 0's and q_i for 1's. Having p_i and q_i as the parameters of the distortion process, M_i matrix has following elements:

$$\mathbf{M_i} = \begin{bmatrix} p_i & 1-q_i \\ 1-p_i & q_i \end{bmatrix}.$$

The last method is more general. The two prior methods are special cases of the last one. Note that in this framework the first method can be viewed as a special case of the third wherein $p_i = q_i = p$. The second method is a special case of the third when $p_i = q_i$.

4 New Optimization Proposal

Now we describe the optimization for MASK scheme which eliminates exponential complexity in estimating support of frequent sets. Instead of the $O(2^n)$ complexity we have $O(2^{rThreshold})$, where $rThreshold$ is a constant.

4.1 Reducing Number of Items in Estimating n-itemsets Support

The reduction of a number of items in estimating n-itemset support can be obtained by choosing for each candidate a subset of distorted transactions for further calculations.

Let $reductionTreshold$ ($rThreshold$) denotes the maximal number of items used in estimating the support of n-itemset.

The reconstruction algorithm performs the mining process like the algorithm presented in Sect. 2.2 until the length of the candidate is greater than $rThreshold.$ Then subset D_A of transactions from the distorted database is chosen and used to estimate the support. We choose those transactions which are supposed to support (in the true database) proper frequent set with length equal to $rThreshold.$ The reconstruction process is performed in the same way until the length of reconstructing itemset exceeds $rThreshold.$ Then subset D_B is chosen from subset D_A. The reduction is done for the second time in the same way as presented earlier.

Here is an example of the above algorithm. Let $rThreshold = 3$. We would like to estimate the support of the candidate "abcd". This candidate was generated from the frequent sets "abc" and "abd". We can use either "abc" or "abd" as a condition to reduce the transaction set. Now we assume that we choose the literally first set - "abc"[3]. Then we choose the distorted transactions which are supposed to support this set. We use an algorithm described in the next Section to achieve this goal. Having subset D_A of transactions chosen, we can estimate the support of candidate "abcd" the same way as we count the support for singletone "d", because all transactions used to estimate this support are supposed to support "abc" frequent set. We compute the support for supersets of "abc" in the same manner until the 7-itemsets candidates, for example "abcdefg", appear.

Having subset D_A we can choose the transactions which are supposed to support "def" set. Thus, we obtain the D_B subset, which contains distorted transactions. Those transactions are supposed to support "abcdef" set in the true database. Now we estimate the support of "abcdefg" set like we compute the singletone "g" support.

The subset D_A is chosen (for each candidate) only in those passes when candidates have length $k \cdot rTreshold + 1, k = 1, 2...$ In other passes we use D_A's from the latest pass in which D_A's were chosen. So, the number of D_A's is less or equal to the number of candidates in the current pass.

[3] We plan to investigate which subset should be chosen to achieve the best accuracy. See Sect. 6.

4.2 Choosing Transactions Subset Algorithm

Having vector C^D and estimated vector C^T, we know what is the estimated support of candidate C (for example candidate "abc") - $C^T_{2^n-1}$. Thus, $C^T_{2^n-1}$ transactions should be chosen to the subset D_A[4]. Randomization factors in the range $< 0.7; 0.9 >$ let us infer that it is probable that transactions which support proper set in the distorted database also support this set in the true database. Thus, we want to keep them in our subset. There are two possible cases:

- $C^T_{2^n-1} \leq C^D_{2^n-1}$ – we choose the first $C^T_{2^n-1}$ transactions which support proper set (for example "abc")[5].
- $C^T_{2^n-1} > C^D_{2^n-1}$ – $C^D_{2^n-1}$ transactions which support the proper set in the distorted database are kept in subset C_A. Then we choose i $(0 \leq i \leq 2^n - 2)$ for which there is the highest probability[6] that the true tuple which supports for example set "abc" was distorted to value i and C^D_i is greater than zero.

5 Experiments

In our experiments we evaluate two kinds of mining errors presented in [8] (Support Error, Identity Error) and one additional metric (Accuracy of Identity):

- Support Error (ρ): This metric reflects the average relative error in the reconstructed support values for those itemsets that are correctly identified to be frequent. Denoting the reconstructed support by *rec_sup* and the actual support by *act_sup*, the support error is computed over all frequent itemsets as follows:

$$\rho = \frac{1}{|F|} \Sigma_{f \in F} \frac{|rec\_sup_f - act\_sup_f|}{act\_sup_f} * 100\ [\%]$$

  We compute this metric individually for each level of itemsets, that is, for 1-itemsets, 2-itemsets, etc.
- Identity Error (σ): This metric reflects the percentage error in identifying frequent itemsets and has two components: σ^+, indicating the percentage of false positives, and σ^- indicating the percentage of false negatives. Denoting the reconstructed set of frequent itemsets with R and the correct set of frequent itemsets with F, these metrics are computed as follows:

$$\sigma^+ = \frac{|R - F|}{|F|} * 100\ [\%], \qquad \sigma^- = \frac{|F - R|}{|F|} * 100\ [\%]$$

[4] We focus only on the $C^T_{2^n-1}$ and $C^D_{2^n-1}$ values because we are interested in the support of set "abc", but not its subsets.

[5] We plan to determine the best way to choose $C^T_{2^n-1}$ transactions. See Sect. 6.

[6] We use **M** matrix to compute this probability.

- Accuracy of Identity (f): This metric reflects the accuracy in identifying frequent itemsets and shows how many sets are correctly identified to be frequent.

$$f = |F \cap R|$$

Our experiments were carried out on two databases:

- A real database, Led-24 [7], which contains information about Light Emitting Diode. There are about 3,200 tuples with 25 attributes, 24 of them are binary.
- A synthetic database generated from the IBM Almaden generator [1]. The data set was created with parameters T10I8D100kN100 (detailed naming convention can be found in [1]). It contains about 100,000 tuples with each customer purchasing about ten items on average.

5.1 Accuracy vs Privacy

Experiments were conducted on the synthetic database with distortion parameters $p = 0.5$ and $q_1 = 0.97$, $q_2 = 0.87$, $q_3 = 0.77$ and no relaxation. Only the results of the experiment with $p = 0.5$ and $q_2 = 0.87$ are shown in Table 1. The level indicates the length of the frequent itemset, $|F_0|$ indicates the number of frequent itemsets at this level, $|F_r|$ ($|F_{rm}|$) shows the number of mined frequent sets from the distorted database using MASK (modified MASK). The other columns are the metrics defined in this Section.

For $p = 0.5, q = 0.97$ MMASK has lower support error and positive error and higher negative error. As a result of these evaluations f_r is higher than f_{rm}.

Decreasing value of q (p is constant and equal to 0.5) results in increasing privacy. For $q = 0.97, 0.87$ and 0.77 Basic Privacy[7] is equal to 63.8%, 81% and 86.1%, respectively. Thus, 10% drop of q (from 0.97 to 0.87) causes Basic Privacy to increase by more than 17%.

As stated in [8], MASK performs much more worse with lower probabilities (for $p = q$). Conducted experiments confirmed this property of MASK (constant p and variable q).

MMASK performs significantly better for lower probabilities. The support error for MASK is as high as 100-300 for levels 5-6, when for MMASK does not exceed 17% for levels 4-6 with $q = 0.87$ (see Table 1).

To sum up, MMASK is significantly better with higher privacy (lower probability q). The accuracy error is always better for MMASK (for levels greater than $rThreshold$).

The results of the experiments with $p = q$ are quite similar. The modified algorithm accomplishes better results than MASK for $p = 0.8$ and $p = 0.7$.

The experiments on the real database (either with $p = q$ or different p and q) lead to the same conclusions (results of these experiments are not shown in this paper).

[7] Basic Privacy represents the probability that the original entry of a given random customer for itemset i can be accurately reconstructed from the distorted database (before the mining process). For details see [8] and [2].

Table 1. Set T10I8D100kN100, p = 0.5, q = 0.87, rThreshold = 3, minSup = 0.005

Level	$\|Fo\|$	$\|Fr\|$	ρ_r	$\sigma{-}_r$	$\sigma{+}_r$	f_r	$\|Frm\|$	ρ_{rm}	$\sigma{-}_{rm}$	$\sigma{+}_{rm}$	f_{rm}	$f_r - f_{rm}$
1	98	98	5.4	1.0	1.0	97	98	5.4	1.0	1.0	97	0
2	2522	2704	20.4	10.8	18.0	2250	2704	20.4	10.8	18.0	2250	0
3	10930	16780	37.5	25.9	79.5	8094	16780	37.5	25.9	79.5	8094	0
4	10185	22411	62.4	40.1	160.2	6098	7787	16.4	36.3	12.7	6490	-392
5	2021	5810	115.9	57.9	245.4	851	1129	16.6	57.3	13.2	862	-11
6	24	200	318.5	79.2	812.5	5	28	6.8	70.8	87.5	7	-2

Different Randomization Factors for Items. We also conducted the experiments with different randomization factors for different items for real and synthetic databases. Half of the items was distorted with parameters $p = 0.4$, $q = 0.88$ and the rest with parameters $p = 0.5$, $q = 0.87$.

The results once again show that modified algorithm is better for low probabilities. In the experiment f metric is higher for the modified algorithm (for levels greater than $rThreshold$).

5.2 Efficiency

Figure 1 shows the running time of the original algorithm based on Apriori and the modified algorithm, as compared to Apriori itself, for various settings of the minimum support parameter for real and synthetic database. Experiments with the original algorithm and MMASK were conducted on the distorted database and Apriori was used with the original database.

The figure shows that there are huge differences in running times between MASK and Apriori algorithm. Overheads between those two algorithms are larger for lower minimum support. Optimization presented in Sect. 4 makes the time of the mining process almost as fast as Apriori. This is the main advan-

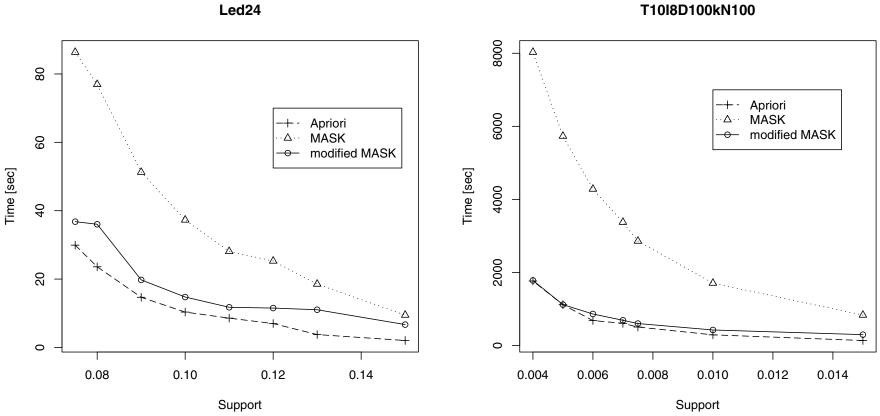


Fig. 1. Time [s] vs Support for real set Led24 and synthetic set T10I8D100kN100

tage, which makes modified MASK viable in practice (EMASK does not break exponential complexity in reconstructing the original support). The reduction in time cost could be also related to lower number of frequent items discovered by MMASK than MASK.

6 Conclusions and Future Work

We investigated the problem of efficiency in the privacy preserving MASK method and described the new optimization, which is different from those presented in literature. The main advantage of this optimization is that it breaks the exponential complexity and makes discovering association rules with preserving privacy viable in practice. Next advantage is that the proposed optimization can be used with different randomization factors for 0's and 1's. Moreover, it allows different items to have different randomization factors. Furthermore, for higher privacy it achieves significantly better results than original MASK algorithm.

Effectiveness of the new solution has been tested on synthetic and real databases and presented in this paper.

In future works, we plan to investigate the possibility of extension of our results to quantitative [11] and generalized [10] association rules.

We also plan to investigate which subset from the two possible should be chosen as a condition to reduce the transaction set to achieve the best accuracy when the candidate length exceeds $reductionThreshold$ parameter. Another possible solution is to combine the results obtained from those two sets.

We plan to determine the best way to choose $C^T_{2^n-1}$ transactions while choosing distorted transactions which are supposed to support proper set in the true database. Now the first $C^T_{2^n-1}$ transactions are chosen.

The best value for $reductionThreshold$ parameter is also an open problem to be investigated.

The false negative error component can be reduced using the relaxation technique presented in [8]. Modified relaxation, which is applied every time the reduction is performed, could also be used. Moreover, these two relaxations could be combined.

Other optimizations presented in literature can be simultaneously applied to achieve improvements in performance.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) VLDB'94., Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile, pp. 487–499. Morgan Kaufmann, San Francisco (1994)
2. Agrawal, S., Krishnan, V., Haritsa, J.R.: On addressing efficiency concerns in privacy preserving data mining. CoRR, cs.DB/0310038 (2003)
3. Atallah, M.J., Bertino, E., Elmagarmid, A.K., Ibrahim, M., Verykios, V.S.: Disclosure limitation of sensitive rules. In: Proceedings of the IEEE Knowledge and Data Engineering Workshop 1999, pp. 45–52 (1999)

4. Dasseni, E., Verykios, V.S., Elmagarmid, A.K., Bertino, E.: Hiding association rules by using confidence and support. In: Moskowitz, I.S. (ed.) Information Hiding. LNCS, vol. 2137, pp. 369–383. Springer, Heidelberg (2001)
5. Evfimievski, A., Srikant, R., Agrawal, R., Gehrke, J.: Privacy preserving mining of association rules. In: KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 217–228. ACM Press, New York (2002)
6. Kantarcioglu, M., Clifton, C.: Privacy-preserving distributed mining of association rules on horizontally partitioned data. In: DMKD (2002)
7. Oates, T., Jensen, D.: Large datasets lead to overly complex models: An explanation and a solution. In: KDD, pp. 294–298 (1998)
8. Rizvi, S., Haritsa, J.R.: Maintaining data privacy in association rule mining. In: VLDB, pp. 682–693. Morgan Kaufmann, San Francisco (2002)
9. Saygin, Y., Verykios, V.S., Elmagarmid, A.K.: Privacy preserving association rule mining. In: RIDE, pp. 151–158 (2002)
10. Srikant, R., Agrawal, R.: Mining generalized association rules. In: Dayal, U., Gray, P.M.D., Nishio, S. (eds.) VLDB, pp. 407–419. Morgan Kaufmann, San Francisco (1995)
11. Srikant, R., Agrawal, R.: Mining quantitative association rules in large relational tables. In: Jagadish, H.V., Mumick, I.S. (eds.) SIGMOD Conference, pp. 1–12. ACM Press, New York (1996)
12. Vaidya, J., Clifton, C.: Privacy preserving association rule mining in vertically partitioned data. In: KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 639–644. ACM Press, New York (2002)
13. Verykios, V.S., Bertino, E., Fovino, I.N., Provenza, L.P., Saygin, Y., Theodoridis, Y.: State-of-the-art in privacy preserving data mining. SIGMOD Record 33(1), 50–57 (2004)
14. Xia, Y., Yang, Y., Chi, Y.: Mining association rules with non-uniform privacy concerns. In: Das, G., Liu, B., Yu, P.S. (eds.) DMKD, pp. 27–34. ACM Press, New York (2004)
15. Yücel Saygin, C.C., Verykios, V.S.: Using unknowns to prevent discovery of association rules. SIGMOD Record 30(4), 45–54 (2001)

Frequent Events and Epochs in Data Stream*

Krzysztof Cabaj

Institute of Computer Science, Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
kcabaj@ii.pw.edu.pl

Abstract. Abstract. Currently used data-mining algorithms treat data globally. Nevertheless, with such methods, potentially useful knowledge that relates to local phenomena may be undetected. In this paper, we introduce new patterns in a form of local frequent events and epochs, boundaries of which correspond to discovered changes in a data stream. A local frequent event is an event which occurs in some period of time frequently, but not necessarily in the whole data stream. Such an event will be called a frequent event in a data stream. An epoch is understood as a sufficiently large group of frequent events that occur in a similar part of the data stream. The epochs are defined in such a way that they do not overlap are separated by so called change periods. In the paper, we discuss some potential applications of the proposed knowledge. Preliminary experiments are described as well.

1 Introduction

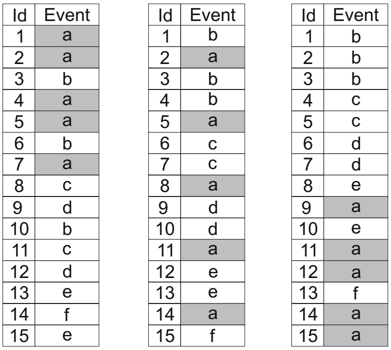
Nowadays, many organizations store huge amount of data. In many cases, the stored data is time ordered, and can be considered as a data stream. The characteristics of this data may change during time. A manual analysis of this data is time consuming; sometimes it might be infeasible to carry out. However, the data analysis may be supported by the usage of efficient data mining techniques. The most common data mining techniques treat data globally. Patterns are discovered regardless of their distribution in a data stream. It may happen that patterns occurring frequently in some parts of the data stream, but not sufficiently frequently in the whole data stream, are not discovered with techniques treating data globally. Three sample data sets are presented in Figure 1. One may note that event a occurs the same number of times in each data set. Assuming that an event is considered as frequent if it occurs at least 5 times in the whole data set, a will be found frequent in each sample data set. For threshold 6, nevertheless, a will be found frequent in none data set, even though it occurs frequently in the beginning of the first data set and at the end of the third. The approach presented in this paper enables discovering patterns that are frequent locally, although may be infrequent globally.

* Research has been supported by grant No 3 T11C 002 29 received from Polish Ministry of Education and Science.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 475–484, 2007.

Based on locally frequent events, we introduce a notion of an epoch. By an epoch we understood a sufficiently large group of frequent events that occur in a similar part of the data stream. The epochs are defined in such a way that they do not overlap are separated by so called change periods. In the paper, we discuss some potential applications of the proposed knowledge. Preliminary experiments are described as well.

The layout of the paper is as follows: The related work is described in Section 2. In Section 3, we propose new patterns - frequent events and epoch in data stream Sample applications of the new approach are presented in Section 4. Section 5, presents preliminary experimental results. Our work is concluded in Section 6.



Id	Event
1	a
2	a
3	b
4	a
5	a
6	b
7	a
8	c
9	d
10	b
11	c
12	d
13	e
14	f
15	e

Id	Event
1	b
2	a
3	b
4	b
5	a
6	c
7	c
8	a
9	d
10	d
11	a
12	e
13	e
14	a
15	f

Id	Event
1	b
2	b
3	b
4	c
5	c
6	d
7	d
8	e
9	a
10	e
11	a
12	a
13	f
14	a
15	a

Fig. 1. Sample datasets, with various distribution of element "a" (darkened in the figure)

2 Related Work

The author of this paper introduces two new patterns, namely frequent events and epochs in data stream. Those new patterns could be treated as a kind of generalization of frequent events, as introduced by Agrawal in [1]. An item set is the subset of items in an analyzed data base. It is called frequent, if it is supported by data more times than a threshold expressed by minSup. The set is frequent regardless its distribution in the analyzed data. The pattern introduced in this paper, in contrast, holds information about time, in which it is frequent. The discovered frequent set in data stream carries not only events identifier but also the range in which it appears, and its support is above a designated level. With the proposed approach we additionally gain information how events appear within a given period.

The most common approach to data stream analysis consists in discovering event sequences. The sequence is an ordered set of events, which appears in a data stream in a predefined time range. There are many algorithms for discovering this kind of information, in particular AprioriAll [2], GSP [4], PrefixSpan [5]. Another approach to data stream analysis has been described inter alia by Mannila, Toivonen and Klemettinen in [6,7]. The patterns introduced by them, episodes and episode rules, hold some information about time ranges. An episode is a collection of events which appear in a partially ordered set, in a given time window. An episode rule ascertains that if an episode appears in the data stream in a predetermined time with a probability, another event would appear later. All recalled patterns hold some information about their distribution in the analyzed data stream. This information can be useful in many situations. We try to go further and introduce new patterns in this paper.

A most similar approach to the presented one appears in [3]. The emerging pattern is an item set which supports growth between two analyzed data sets. The emerging pattern carries information that some item set appears rarely in one data set and frequently in another one. The major disadvantage of this method is that the sequencies (patterns) could be discovered only by analyzing two data sets. We introduce in this paper new patterns in the data stream,, namely frequent events and epoch. Discovering them does not have the disadvantage mentioned above. Below we describe then in more detail.

3 Definitions

Denote $\mathrm{E_0}$ as a set of events identifiers. *Event* is a pair (e, t), where $\mathrm{e_i} \in \mathrm{E_0}$ and t is a time where this event occurred. *Data stream* is an ordered triple $< \mathbf{t}_\mathrm{b}^\mathrm{DS}, \mathbf{t}_\mathrm{e}^\mathrm{DS}, \mathbf{S} >$, where $\mathbf{t}_\mathrm{b}^\mathrm{DS}$ and $\mathbf{t}_\mathrm{e}^\mathrm{DS}$ are integer numbers, greater than 0, which represents, beginning and ending moments of this data stream. S is an ordered sequence of events

$\mathrm{S} =< (\mathrm{e_1}, \mathrm{t_1}), (\mathrm{e_2}, \mathrm{t_2}), \ldots (\mathrm{e}_n, \mathrm{t}_n) >$ and for all pairs $\mathbf{t}_\mathrm{b}^\mathrm{DS} \leq \mathbf{t}_\mathrm{i} \leq \mathbf{t}_\mathrm{e}^\mathrm{DS}$ and $\mathbf{t}_\mathrm{i} \leq \mathbf{t}_\mathrm{i+1}$.

Example 1. Consider sample data stream, presented in Figure 2. For this data stream, events identifier set consist of events A, B, C and D, $\mathrm{E_0} = \{\mathrm{A, B, C, D}\}$. The data stream has the form <3, 32, S>, where some starting elements from S are presented below:

$$(\mathrm{A}, 3), (\mathrm{B}, 3), (\mathrm{B}, 4), (\mathrm{C}, 4), (\mathrm{A}, 5) \ldots.$$

This data stream will be used in next examples. □

As a window in a data stream, we define an ordered triple $\mathrm{W} =< \mathrm{t_b^W}, \mathrm{t_e^W}, \mathrm{WS} >$, where $\mathrm{t_b^W}$ and $\mathrm{t_e^W}$ are integer numbers representing the begin and the end time of the given window, if the conditions $\mathrm{t_b^{DS}} \leq \mathrm{t_b^W}$ and $\mathrm{t_e^W} \leq \mathrm{t_e^{DS}}$ are fulfiled The subsequence WS contains the all pairs $(\mathrm{e_i}, \mathrm{t_i}) \in S$, for which $\mathrm{t_b^W} \leq \mathrm{t_i} \leq \mathrm{t_e^W}$. We define the window length Wl(W) as integer calculated as a difference between the end and the begin window time, i.e. $Wl(W) = \mathrm{t_e^W} - \mathrm{t_b^W}$. For a given event e in window W, we define its support (and denote by eSup(W, e)) as the number of pairs $(\mathrm{e_i}, \mathrm{t_i})$ in the stream, where $\mathrm{e_i} = \mathrm{e}$. We say that event e is frequent in window W, when its support is eSup in given windows is greater than assumed parameter minESup, eSup(W, e)>minESup.

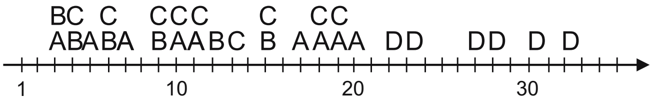


Fig. 2. Sample data stream used during all examples in this paper

Example 2. Lets consider window W=$< 20, 28, < (\mathrm{A}, 20), (\mathrm{D}, 22), (\mathrm{D}, 23), (\mathrm{D}, 24), (\mathrm{D}, 27), (\mathrm{D}, 28) \gg$ and parameter minESup set by user to value 3. In this window event D appears 5 times, which means that using this data we could say that D is a frequent event in this window. □

Interval denoted $< \mathrm{t}_\mathrm{b}^\mathrm{I}, \mathrm{t}_\mathrm{e}^\mathrm{I} >$, is a set which contains all integers t which fulfill equation $\mathrm{t}_\mathrm{b}^\mathrm{I} \leq \mathrm{t} \leq \mathrm{t}_\mathrm{e}^\mathrm{I}$, Time $\mathrm{t}_\mathrm{b}^\mathrm{I}$ is called the begin of interval and the time $\mathrm{t}_\mathrm{e}^\mathrm{I}$ is respectively called end of interval. In the following text intervals are denoted with small letters.

Event e is frequent in interval p $=< \mathrm{t}_\mathrm{b}^\mathrm{P}, \mathrm{t}_\mathrm{e}^\mathrm{P} >$, where for each $\mathrm{t} \in \mathrm{P}$, exists window W, in which e is frequent and where $\mathrm{t}_\mathrm{b}^\mathrm{W} \leq \mathrm{t} \leq \mathrm{t}_\mathrm{e}^\mathrm{W}$. Numbers $\mathrm{t}_\mathrm{b}^\mathrm{P}$ and $\mathrm{t}_\mathrm{e}^\mathrm{P}$ are called begin and end time of event frequency in this interval.

Exact interval for event e is interval, in which e is frequent and pairs $(\mathrm{e}, \mathrm{t}_\mathrm{b}^\mathrm{P})$ and $(\mathrm{e}, \mathrm{t}_\mathrm{e}^\mathrm{P})$ are in S.

Interval p is called *maximal exact interval for event e* when p is exact interval for e and there is no interval q, in which e is frequent and p is subset of q.

Example 3. Assume parameters Wl = 8 and minESup = 2. Using those parameters we find frequent intervals in the data stream presented in Figure 2. Using those constraints we can say that D is frequent in interval $\mathrm{p}_1 =< 26, 34 >$ because there is a window $\mathrm{W} =< 24, 32, \{(\mathrm{D}, 24), (\mathrm{D}, 27), (\mathrm{D}, 28), (\mathrm{D}, 30), (\mathrm{D}, 32)\} >$ in which D is frequent and each $\mathrm{t} \in \mathrm{p}_1$ encloses in this window.

p_1 is not an exact interval, because in analyzed data stream there is no pairs (D, 26) and (D, 34) which meets the case of event at begin and end time of interval. As an example of an exact interval where D is frequent we can show interval $\mathrm{p}_2 =< 27, 32 >$.

p_2 is not a maximal exact interval for event D, because we could show other interval $\mathrm{p}_3 =< 22, 32 >$, which includep_2 . This interval is maximal, because in the presented data stream we could not show other, larger interval which begins and ends with event D. □

Frequent event in data stream is such $\mathrm{e} \in \mathrm{E}_0$, which in given parameters window length (Wl) and minimal support (minESup) is frequent at least in one window. Existence of such a window implies one interval in which e is frequent.

For given parameters Wl and minESup, set of all frequent sets in analyzed data stream is called FES (Frequent Events Set).

For given parameters Wl and minESup, set of all intervals in which events are frequent is called FIS (Frequent Intervals Set).

For purpose of not loosing some important information mapping of frequent events in data stream and corresponding intervals are introduced. Event mapping, denoted as EM, is a set of pairs (e, i) where e ∈ FES and i ∈ FIS. The pair (e, i) is added to EM when event e is frequent in interval i.

Example 4. In this example we once again back to data stream presented in Figure 2. For this example we assumed that parameters have set values Wl = 8 and minESup = 2. Below are presented discovered frequent events in data stream, corresponding intervals when they are frequent and mapping set.

$$\mathrm{FES} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}$$
$$\mathrm{FIS} = \{< 3, 20 >, < 3, 15 >, < 4, 19 >, < 22, 32 >\}$$
$$\mathrm{EM} = \{< \mathrm{A}, < 3, 20 \gg, < \mathrm{B}, < 3, 15 >>, < \mathrm{C}, < 4, 19 \gg, < \mathrm{D}, < 22, 32 \gg\}$$

□

In the following part of this paper we consider only frequent events in data stream. For all next definitions we assumed that parameters Wl and minESup are set by user, who performs analysis. Earlier defined sets FES, FIS and EM are used for new definitions, too.

Define *sequence of event frequency change*, called CS (Change Sequence), as a sequence of integers representing time moments when maximal exact interval for given event start or stopped.

$CS =< \mathrm{t}_1, \mathrm{t}_2, \ldots, \mathrm{t}_\mathrm{n} >$ where $\mathrm{t}_\mathrm{i} \leq \mathrm{t}_{\mathrm{i}+1}$ and for each t_i exist such p ∈ FIS, that $\mathrm{t}_\mathrm{i} = \mathrm{t}_\mathrm{b}^\mathrm{p}$ or $\mathrm{t}_\mathrm{i} = \mathrm{t}_\mathrm{e}^\mathrm{p}$.

Example 5. For prior presented set FIS which consist of intervals $< 3, 20 >$, $< 3, 15 >, < 4, 19 >, < 22, 32 >$ a CS sequence carries values $< 3, 3, 4, 15, 19, 20,$ 22, 32 >. □

The *change period* is an interval in which many maximal exact intervals begin or stop. In such a moment, changes are no more than cwl (change window length) time units far one from other. In case of defining change period, auxiliary *sequence of changes sets* CPS (Change Period Sequence) is used.

Sequence CPS contains *change period elements CPE*,

$\mathrm{CPS} =< \mathrm{CPE1}, \mathrm{CPE2} \ldots \mathrm{CPEn} >$

where each CPE_n set is described by one of followed equality:

$$CPE_1 = \{t_{i \in} CS : \exists_{m \in N} \forall_{l \in N, l < m} t_{l+1} - t_l \leq cwl\} \tag{1}$$
$$CPE_n = \{t_{i \in} CS : \exists_{k \in N} \forall_{l \in N, k < l < m} t_{m+1} - t_m \leq cwl \wedge t_m > max(CPE_{n-1})\}.$$

Where function max returns the maximal value stored in set.

Using CPS sequence, the change period is defined as an interval $\mathrm{cp} =< \mathrm{t}_\mathrm{b}^\mathrm{cp}, \mathrm{t}_\mathrm{e}^\mathrm{cp} >$, where $< \mathrm{t}_\mathrm{b}^\mathrm{cp} = \min(\mathrm{CPE}_\mathrm{i})$ and $< \mathrm{t}_\mathrm{e}^\mathrm{cp} = \max(\mathrm{CPE}_\mathrm{i})$. Functions min and max return rightly minimal and maximal element from input set.

Example 6. For described sample data stream and earlier discovered maximal exact interval CPS $=< (3, 3, 4), (15), (19, 20, 22), (32) >$ discovered are change periods $< 3, 4 >, < 15, 15 >, < 19, 22 >, < 32, 32 >$.

The Epoch is an interval $\mathrm{E} =< \mathrm{t}_\mathrm{b}^\mathrm{E}, \mathrm{t}_\mathrm{e}^\mathrm{E} >$, where $\mathrm{t}_\mathrm{b}^\mathrm{E}$, $\mathrm{t}_\mathrm{e}^\mathrm{E} >$ are integers which represent the begin and the end time of epoch, under conditions that $\mathrm{t}_\mathrm{b}^\mathrm{DS} \leq \mathrm{t}_\mathrm{b}^\mathrm{E} \leq \mathrm{t}_\mathrm{e}^\mathrm{DS}$ and $\mathrm{t}_\mathrm{b}^\mathrm{DS} \leq \mathrm{t}_\mathrm{e}^\mathrm{E} \leq \mathrm{t}_\mathrm{e}^\mathrm{DS}$. Using CPS sequence endings of E interval could be found. If we assume that epochs are numbered from 1, then for epoch number n $\mathrm{t}_\mathrm{b}^\mathrm{E} = \max(\mathrm{CPE}_\mathrm{n})$ a $\mathrm{t}_\mathrm{e}^\mathrm{E} = \min(\mathrm{CPE}_{\mathrm{n}+1})$. Epochs and change periods in analyzed data stream contact one to other subsequently.

For each epoch associated is a set called *epoch description*, later called ED. ED is the subset of FES satisfying following equality

$$ED = \{e \in FES : \exists_{<ev,P> \in EM} \forall_{t \in <t_b^E, t_e^E>} Pt_b \geq t \wedge Pt_e \leq t \wedge ev = e\}. \quad (2)$$

□

Example 7. For given parameters cwl = 2, Wl = 8 and minESup = 2 the sample data stream presented in Figure 2 have the change periods $< 3, 4 >$, $< 15, 15 >$, $< 19, 22 >$, $< 32, 32 >$ and epochs with a corresponding to them epoch descriptions $< 4, 15 >$ and $\{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$, $< 15, 19 >$ and $\{\mathrm{A}, \mathrm{C}\}$, $< 22, 32 >$ and $\{\mathrm{D}\}$. □

Using pattern introduced before could lead to a situation, where to many patterns are discovered and presented to the person who analyzes data. For the purpose of delivery more aggregated and useful pattern *generalized epochs and change periods* are introduced. In described prior patterns even one change in frequent event's maximal exact interval generates a new change period (compare example 6 and 7, and time event 15). In generalized approach the new parameter appears - cpSup (change period support). Now, as generalized change periods are considered only those change periods in which at least cpSup changes are discovered. When the change period contains less than cpSup changes, it is not generated and is contained in corresponding epoch. To define new generalized change periods auxiliary *generalized change period sequence*, denoted GCPS is introduced. GCPS contains change period elements GCPE from CPS, which satisfy condition that $\mid \mathrm{CPE}_\mathrm{n} \mid \geq cpSup$.

Using those assumptions, generalized epoch is an interval $\mathrm{GE} =< \mathrm{t}_\mathrm{b}^\mathrm{GE}, \mathrm{t}_\mathrm{e}^\mathrm{GE}, >$, where the beginning and the ending time $\mathrm{t}_\mathrm{b}^\mathrm{GE}$ and $\mathrm{t}_\mathrm{e}^\mathrm{GE}$, are defined for epoch number n as $\mathrm{t}_\mathrm{b}^\mathrm{GE} = \max(\mathrm{GCPE}_\mathrm{n})$ and $\mathrm{t}_\mathrm{e}^\mathrm{GE} = \min(\mathrm{GCPE}_{\mathrm{n}+1})$.

The epoch description for generalized epoch is a subset of FES, which contains elements satisfying equality:

$$ED = \{e \in FES :_{<ev,P> \in EM} \exists_{t \in <t_b^E, t_e^E>} P.t_b \geq t \wedge P.t_e \leq t \wedge ev = e\} \quad (3)$$

Example 8. For the data stream presented In Figure 2, using parameters cwl = 2 and cpSup = 2, generalized epochs are discovered with corresponding epoch description sets $< 4, 19 >$ and $\{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$, $< 22, 32 >$ and $\{\mathrm{D}\}$. □

4 Application

In previous sections the idea of new knowledge discovery methods has been introduced; frequent events and epochs in data stream. These patterns could be used everywhere where knowledge about changes in data manner is needed. There are many possibilities; however, in this paragraph only the most promising ideas are described.

Today e-mails are widely used. People use it for work, official cases, and entertainment. Analyzing these e-mails could give us very valuable knowledge about the way in which an organization works or even about the person. For instance, by analyzing mails which were sent to an office, we could discover people who in some time periods send many letters. This information could be valuable for office managers, and could mean few things. It might be a signal that this person has some problems, and sends so many letters, because office workers could not solve them. It might be also a proof that this is a very annoying person, and his letters should be serviced in other way. Using introduced before patterns, such information could be delivered to manager rapidly, without time consuming analysis of all mails. This approach could be used for analyzing mails during an investigation. Officers who use those methods could easily find people with whom this person contacts.

Our mails could not be delivered without computer networks. Nonetheless, today more and more often this infrastructure is attacked.. Each element of this infrastructure produces high volume of logs information. Suspicious activity or other changes in manner leaves some signs in this data. Analyzing those data by introduced data-mining algorithms could easily discover changes. Those changes could be sign of some attack or other misconfiguration or change in user activity, which maybe need more detailed investigation.

When time parameter is slightly changed the new application appears for introduced methods. One of the most promising is connected with text analysis. In this case, time represents the position of a given word in the text. Using epochs an automatic section and a keyword discovery could be done.

5 Preliminary Experiments

During preliminary experiments three types of data is analyzed, using previously introduced techniques. As a first data set, headlines of security news from polish CERT team [9] are used. This data set was analyzed using frequent events in data stream. As a second data set electronic mails are used. On the second experiment official mails are analyzed. In this case subjects of mails are used, and epochs are used as well. The latest experiments are connected with text analysis. During the experiment a discovered epoch represents a section in the original text. The epoch description could be treated as a keyword. Experiments are carried out on NASA text which describes the new space ship Orion.

5.1 Security News Headlines

In first experiments security news headlines are analyzed. As an input set, headlines from polish CERT news [9], are used. For each article one line input was generated. At the begging of line was date when article was added, and then whole title appeared. During analysis 770 headlines, from August 2001 to September 2006 are used. Before the main algorithm steps some data preparation is done. Each word from headlines was treated as an event. The stop-list was used for deleting words which do not carry useful information (like preposition etc ...). This preprocessed data is then analyzed using few sets of parameters values. The window length parameter has set values from one week to two weeks. The support during experiments is set to value 2 and 3.

After analysis of this data some useful knowledge are discovered. All famous internet worms names are discovered as frequent events in data stream (for example Netsky, Beagle, Code Red, and Slammer etc...). Interval when those events are frequent coincides with time when those worms attack computer word. What are more important, and useful in many cases near to those worms names, names of affected software appear, for example, worms Netsky and Beagle attacks Microsoft Windows and Slammer attacks SQL server. This experiment, in our opinion, proves that presented method could be used for analyzing headlines or short descriptions. Analyst using introduced methods could easily find interesting words, and periods when those words appear. This approach could speed up time used for analyzing some stored texts in case of discovering what was often mentioned probably important in some time range.

5.2 Official Mail

In this experiment frequent events and epochs are discovered. In this case subjects of official mails are used, and treated similar to headlines described in prior section. Analyzed mails represent correspondence with students during 4 semesters (from winter semester 2004 to summer semester 2006). In this period more than nine and a half hundred of mails are received. As in before described experiments few parameters sets are used in the first phase to tune algorithms to those data set. In effect best for this data, and used for all analysis parameters are: window length two weeks, support set to value 3 and used during epoch discovery change window length set to 3 days and change period support set to 4. During analysis are discovered epochs, which describes subjects which author taught. For example in epoch 25 events SKM2 (classes name) and WiFi (classes topic) are revealed. During further analysis of this data many other similar to before described epochs are discovered. All subjects which owner of this mailbox taught are discovered. Sometimes even more precise information, for example about additional term, are explored. Using those parameters even such information that in this week is exercise of given number are discovered. But, sometimes other more mysterious epoch are discovered. For example epoch at the end of February with no events. At first sight this is some program error. More detailed analysis shows that this is winter holidays. And in this period no letters from students and other teachers are received.

5.3 Text Analysis

Introduced techniques could be used for text analysis. In this case the time parameter has slightly different meaning. In this experiment an event represents a word, and the time its position in the text calculated as number of currently analyzed sentence. The experiment is carried out the document which describes the new NASA space ship – CEV (Crew Exploration Vehicle) [8]. During analysis only first 750 sentences are used – which in the original document describe the main characteristic and mechanisms of the space ship. Data is preprocessed before epochs discovery is carried out. The simple preprocessing using a stop-list was conducted. All words which do not carry any information (like a, an, the, is, was etc ..) are removed from the text. After this preliminary work, the main analysis begins. All presented results are gain when algorithm parameters are set to corresponding values eSup = 5, windowLength = 20, cwl = 1, cpSup = 1. In this case 17 epochs, which represents a section in the text, are discovered. The average epoch length is 41 sentences; change periods between epochs have the average length of 2 sentences. A description of epoch in this case could be consider as the set of keywords, which describes what subject is discussed in the fragment The crucial thing is that sections that appear as a result of the analysis are similar to main paragraphs of this document For example, the original document starts with the description of the space ship and its mission profile. In this section we could find information that it could be used for the lunar mission and resupply ship for ISS (International Space Station). In the first epoch description we could find words: crew, exploration, vehicle, lunar, ISS, mission. When we look at the next epoch's description, the main subject of it could be rapidly derived. For example, in the 7th epoch words: RCS (Reaction Control System) control, trim, attitude, tanks are discovered. The corresponding section in original text is devoted to the description of the system which controls space ship moments in Space. Next discovered epochs are characterized by sections about the power distribution system, environment, thermal control, and landing system. Each above mentioned section appears in analyzed the original text. Using introduced in this paper patterns, automatic section and keyword generation could be easily done. An analyst using this technique could rapidly discover what the content of unknown document is.

6 Summary

In this paper new knowledge discovery patterns are introduced. Frequent events and epochs are patterns which could discover potentially useful knowledge that relates to local phenomena in data stream. Experiments proofed, that this method could be used for discovering useful information. Using proposed patterns information for example about some people behavior or subjects which are interesting for them, rapidly could be extracted from their mailbox. This allow analyst to work only on potentially interesting data without wasting time on working with all data. When time parameter meaning is slightly changed introduced patterns, could be used for automatic section and keyword generation from text, which preliminary experiments show. There are other applications in which this kind of patterns may be useful.

Acknowledgement. This Research has been supported by grant No 3 T11C 002 29 received from Polish Ministry of Education and Science. The author thanks professor Marzena Kryszkiewicz for her valuable comments especially during mathematical formulation of introduced patterns, professor Henryk Rybiski and my friend Ania for patience during punctuation and language corrections.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proc, Int. Conf. Very Large Data Bases (VLDB'94), Santiago, Chile, September, 1994, pp. 487–499 (1994)
2. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proc, Int. Conf. Data Engineering (ICDE'95), Taipei, Taiwan, March, 1995, pp. 3–14 (1995)
3. 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(SIGKDD'99), San Diego, USA, pp. 43–52 (1999)
4. Srikant, R., Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements. In: Proc. 5th Int. Conf. Extending Database Technology (EDBT'96), Avignon, France, pp. 3–17 (March 1996)
5. Pei J., Han J., i in.: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix Projected Pattern Growth. In: Proc. 2001 Int. Conf. Data Engineering (ICDE'01), Heidelberg, Germany, April 2001, pp. 215–224 (2001)
6. Mannila, H., Toivonen, H., Verkamo, A.I.: Discovering frequent episodes in sequence. In: Proceedings of the First International Conference on Knowledge Discovery and Data Mining. Montreal, Quebec, pp. 144–155 (1995)
7. Klemettinen, M.: A Knowledge Discovery Methods for Telecommunication Network Alarm Database, PhD. Thesis, University of Helsinki (January 1999)
8. NASA: NASA's Exploration Systems Architecture Study – Final Report, Part 5: Crew Exploration Vehicle, http://www.nasa.gov/mission_pages/constellation/news/ESAS_report.html
9. http://www.cert.pl/index2.html?action=show_news_arch

Memory Efficient Algorithm for Mining Recent Frequent Items in a Stream*

Piotr Kołaczkowski

Warsaw University of Technology, Institute of Computer Science
P.Kolaczkowski@ii.pw.edu.pl

Abstract. In the paper we present an improved version of multistage hashing based algorithm, used to find frequent items in a stream. Our algorithm uses low-pass filters instead of simple counters, so it concentrates more on recent items and ignores the old ones. Such behaviour is similar to sliding window based algorithms, but requires less memory and is suitable for real-time applications. The algorithm continuously gives estimates of frequencies of the most frequent items. It was tested with streams having various frequency distributions and proved to work correctly.

1 Introduction

Identifying frequent items in a stream is an important task in various fields of computing, e.g. network traffic, database workload or search engine workload analysis. The problem is complex, because it is usually not possible to store even a small fraction of the data from the stream in the memory for later offline analysis. There is usually too much data per a unit of time. It is also difficult or even not possible to attach a counter to each item category. For example if we had to analyze network flows described as a pair of IPv4 addresses each, we would have 2^{64} potential flows to monitor. Even though we would never see most of them, the number of those seen would still reach several hundred thousands [1].

Fortunately, the number of interesting frequent item categories is usually relatively small compared to the number of all item categories. They are called *heavy-hitters*. In most applications, only the heavy-hitters are taken into consideration. For instance, a database administrator would like to know the queries having the largest impact on the system's load, i.e. the most frequent and the longest ones. These queries should be optimized first and, if further optimization is not possible, then probably they are good candidate for caching. The same applies to tuning a web query engine. In network traffic monitoring it is also required to know the largest packet flows in order to prevent *denial of service attacks* (DOS).

Apart from that we observe a need for not only finding frequent items in the *whole* stream, but just in the *most recent part* of it. This is due to the fact,

* Research has been supported by grant No 3 T11C 002 29 received from Polish Ministry of Education and Science.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 485–494, 2007.

that frequencies of the items may significantly change over time. This is the case of a sudden DoS attack or when a new version of software using a database is installed. In the abovementioned applications we are usually more interested in the statistics from the last 10 minutes than from before 3 months.

2 Problem Definition

Consider a stream of items where each item belongs to one of N categories. The N can usually range from hundreds to several millions. Each item category can appear more than once. The number of items is unlimited and not known in advance. Each item can be read only once – it is not possible to rewind or restart the stream. The algorithm must answer queries about the k most frequent categories that were recently seen. It should also estimate the frequencies of these items. The queries can be sumbitted and must be immediately answered at any time. By the recently seen items we understand the items that appeared within a time window beginning at $t_q - \tau_w$ and ending at t_q, where t_q is the time of submission of the query and τ_w is a constant window size.

An ideal algorithm would give all and only k categories with their corresponding frequencies. If some frequent item is missing from the result set, it is called a *false negative*. If some non-frequent item is reported, it is called a *false positive*. Usually it is sufficient to have an algorithm that reports no false negatives and only a few false positives. False positives can be detected and removed, but this usually requires additional memory.

3 Prior Work

The problem of finding frequent items in a stream of n items has been studied for over past two decades. The earliest algorithm [2] guaranteed to find the item that occurred more often than for half of the time. This was further generalized by Misra and Gries to k items with frequencies higher than $1/(k+1)$ by using k counters [3]. These algorithms required additional second pass to estimate the exact frequencies of found items and to prune the infrequent items that could be also found in the output. This second pass could be omitted in case false positives were tolerable and no estimation of item frequencies were required. The time complexity of the Misra-Gries algorithm was further improved by usage of more sophisticated data structures [4], but the other properties of the algorithm remained the same.

Manku and Motwani presented a one-pass algorithm [5] that give guarantees on the minimal frequencies of found items while still having low space requirements $O(k \log{(n/k)})$. Their algorithm uses similar approach to that of Misra-Gries – it associates a counter with each observed item and prune the counter list according to some special algorithm. Similar results have been also achieved by sketch-based algorithm proposed by Charikar et al. [6]. Their algorithm doesn't store the individual counters, but uses a complex data structure that can estimate frequencies of the infrequent items. The infrequent items are added to the frequent items set whenever their estimated frequency exceeds some threshold.

Estan et al. [7] proposed a hash-based algorithm called a *multistage filter*. The basic idea is that the frequency counters are associated with item hashes, not the items themselves. Many items may hash to the same value, thus having a common counter. By using multiple, independent hashing functions, the algorithm assures the risk of false positives is very low, especially for real-world frequency distributions.

None of the abovementioned algorithms directly addresses the problem of *data aging*, that is the fact that the older items are not as important as the recent items. The data aging can be handled by periodically reseting the counters of a frequent item mining algorithm, so that it "forgets" the old aggregated data. This unfortunately disallows continuous monitoring, as the results should be retrieved only just before the counters are reset. Besides, the same algorithm applied to two identical streams, but having non-zero time offset between each other can produce different frequent item sets. The solution to this problem is using a sliding-window based algorithms like the one in [8]. However, for large windows, their high memory requirements are a huge disadvantage.

The problem of dynamic tracking of frequent items in streams has been recently studied by Cormode and Muthukrishnan [9]. Their work focuses on handling database inserts and deletes, while our approach addresses more general problem of changing frequency distribution.

4 Algorithm

Our algorithm uses low-pass filters to measure frequencies of events and a sketch based method, similar to that presented in [7], to filter out the infrequent items within a limited memory.

4.1 Low-Pass Filters

A low-pass filter can be used to estimate a frequency of some events. This simple idea has been borrowed from the signal theory. For each item in the stream, the filter is given a Dirac-delta-shaped impulse on the input. It can be shown that the value of the output signal of such filter is nearly proportional to the frequency of the input signal, if only the signal frequency is high enough [10]. The time constant τ of the filter controls how fast can the output signal value follow the input frequency and how accurate is the estimation. The greater the τ is, the more time is required for the filter to react to frequency changes, but the more accurate the measurement of lower frequencies is. The filter state consists of a single precision counter c that stores the output signal value and a variable t_l storing the time, when the filter's state was last updated. Whenever the input impulse of value v comes to the filter, the state is updated according to the equation:

$$c' = ce^{\frac{t_l - t}{\tau}} + v, \tag{1}$$

where t is the current time. If only the frequency of events is measured, v should always be set to 1. Setting v to e.g. the size of a network packet would cause the filter to estimate the size of a data flow, and setting it to the duration of a database query execution would estimate the average database load. In the further part of the paper we are always referring to the frequency measurement but all the described methods are equally valid for the data flow or workload estimation.

The output value can be calculated at any time from the equation:

$$c' = ce^{\frac{t_l - t}{\tau}} \tag{2}$$

Such a filter has different properties than a simple event counter. A simple event counter can be used to measure the frequency of events in some period of time, but the exact result is known only at the end of the period. If a continuous monitoring is required, then the counter must be periodically restarted. The state of the counter reflects only the number of events seen since the time the counter was restarted, so the frequency cannot be estimated with the same accuracy each time. On the contrary, the low-pass filter can estimate average event frequency at any time with the same accuracy. This enables to trace frequency changes with higher time resolution, so that new frequent events can be discovered much earlier. This is illustrated in the figure 1. Note that the waveform of the simple counter is much different after the occurrence of the second impulse than the first one, while both the sliding window (moving average) and the low-pass filter give stable results. Both impulses are of same value and length. The restart period of the counter, the size of the sliding window and the time constant of the filter were set to 1.0 in this experiment.

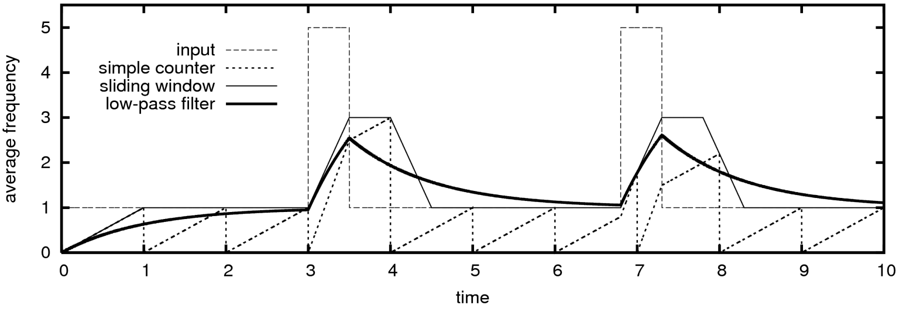


Fig. 1. Various techniques of average frequency measurement

The low-pass filter does not solve the stated problem exactly. The "time-window" is somehow "fuzzy". The most recent items are indeed more important than the previous ones, but the shape of the window is not rectangular but exponential. However, as seen in the figure 1, this would not be a large problem in practical applications. The results obtained by using a rectangular and an exponential windows are quite similar.

4.2 What Is Frequent and What Is Not

The simplest idea would be to associate an exactly one low-pass filter with a one item category and choose only the k categories with the highest frequency. Unfortunately this cannot be done, because the number of the categories N may be huge and the frequency meters would take up too much memory. Thus only the most frequent categories are associated with filters, one category with one filter. The only problem is how to tell which categories should be in this set. There must be some rules for adding new frequent categories to this set and removing the infrequent ones. Somehow the frequencies of the infrequent items should be estimated, too. Note that this estimation need not to be very accurate, as we only want to decide if the item is worth being in the frequent set or not. This estimation is done thanks to filter sharing implemented by hashing.

Consider m filters numbered from 1 to m. Each item category is hashed to a one filter from this set. We will call this data structure a *sketch*. Because $M \ll N$, more categories can hash to the same filter. Each filter in the sketch will measure a sum of the frequencies of categories that are hashed to it. By the *frequency threshold* f_{thr} we will understand the frequency above which a category is considered frequent and by the *maximum frequency level* $f_{\max}$ we will understand the frequency that could be reached by the category if no other categories were present in the stream. In certain applications $f_{\max}$ can denote a network connection capacity or a maximum system throughput. We will call a filter that reports frequency above the f_{thr} a *hot filter*. A *cold filter* is any filter that is not a hot filter. If $M \geq k$ and $f_{\text{thr}} \geq f_{\max}/k$, only k reported frequencies will have a chance to reach the high frequency level f_{thr}, for any frequency distribution. Event frequency distributions usually follow a Zipf-like distribution in real world applications. In these cases the number of categories exceeding f_{thr} will be even smaller or the f_{thr} can be set much lower.

The categories that hash to a cold filter cannot be frequent, so they are not added to the frequent set. If they were frequent, they would make the filter hot after some time. Thus false negatives are not possible. Unfortunately false positives are possible because still an infrequent event can hash accidentally to a hot filter and pass to the frequent set. To lower the risk of false positives, the number of cold filters should be high. This can be achieved by setting the total number of filters in the sketch to a value far greater than k. The only problem is, that this solution does not scale well with the increasing number of categories - the m must be proportional to the number of categories N.

The scaling problem has been solved by the parallel multistage filtering, an idea introduced in [7]. Note that the word *filter* in that article does not have anything to do with our *low-pass filters*. We will call it *multistage hashing* to avoid the name clash. Instead of hashing only once and updating the state of only one filter in the sketch, the hashing process is performed in M stages by using M independent hashing functions. States of M filters in the sketch are updated for each item in the stream. The event is considered to be frequent only if all frequencies reported by each of the M filters are greater than f_{thr}. This significantly lowers the probability for an infrequent event to be classified as a frequent one. If the probability of event hitting a hot filter in each stage is p, then the probability of hitting a hot filter M times and passing to the frequent set is p^M. Of course the size of the sketch must be now M times higher to make this probability remain on the same level as in the one-stage version. Using m only one or two orders of magnitude higher than $s * k$ an having only a few stages can yield very low false positives rates. A thorough theoretical analysis of multistage filters can be found in the original paper [7].

4.3 Managing the Frequent Set

Each entry in the frequent set consists of an item category and its corresponding low-pass filter. The low-pass filter is updated for each occurrence of the item of that category in the input stream. A category is added to the frequent set if any item belonging to that category hashes to hot filters at all stages. A category is removed from the frequent set if its filter becomes cold and so does at least one of the filters being hashed to in the sketch. Otherwise removing such category would be followed by an immediate readding. Because it would be too expensive to update all sketch filters in each iteration, the process of purging the infrequent categories is executed only before adding a new frequent category that would make the frequent set larger than the maximum past size of the frequent set or whenever the user asks for the results.

4.4 Algorithm Improvements

Estan et al. proposed many improvements to the original algorithm. Some of them can be also applied to our modified version of the algorithm.

Serial Filtering. In serial filtering the event passes to the next stage only if the values of counters at all previous stages were high enough. The event is blocked by the first low counter and the remaining counters are not checked and not modified. This keeps more counters in a low state (cold). Serial filtering has one disadvantage that makes it less usable for continuous measurements than the parallel filtering: it requires to see M times more items of the same category to detect a new frequent item. This is due to the fact that all the counters/filters in all stages must achieve enough high values and they must do it serially.

Preserving Entries. Preserving entries is a technique introduced to improve the accuracy of measurement of frequent items (called large flows in the original paper). Because the original algorithm uses simple, periodically restarted counters, clearing the frequent set after each measurement period caused that frequent items were not counted for some time at the beginning of each period. The optimisation was to leave the frequent items entries in the frequent set and clear only the counters. The problem does not exists in our algorithm as the filters *are never* restarted and the measurement is continuous. Thus the improvement does not apply.

Shielding. The shielding is an improvement consisting in not performing the sketch update on each item that is found in the frequent set. This can be easily applied in our version of the algorithm. The shielding is turned on for all items that have a hot filter in the frequent set. The main purpose of the optimisation is to make some hot filters in the sketch cold and reduce the probability of false positives. The side-effect is that if the frequent item becomes infrequent for some time, it will have to wait longer for being included in the frequent set again, as its filters in the sketch might already become cold.

Conservative Update of Counters. Instead of increasing the counters at all the stages about the full amount, only the smallest one is increased about the full amount and the rest is set to the maximum of the old value and the new value of the smallest counter. This prevents counters to increase too fast and reduces the probability of false positives even further. The technique is applicable to the low-pass filters. The value v in (1) is set appropriately at each stage just in the same way as it would be with simple counters.

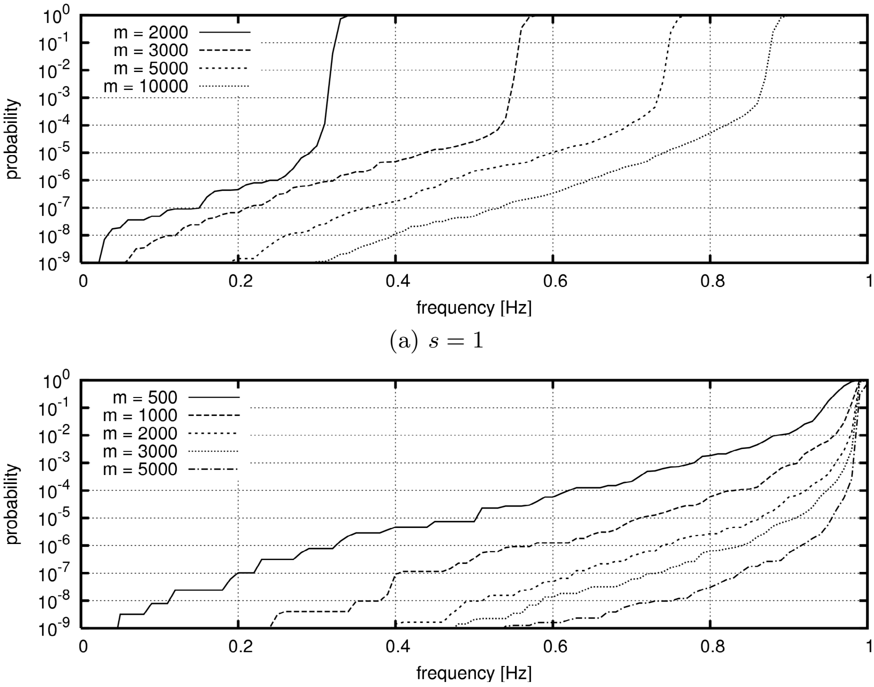
5 Experiments

To check correctness of the algorithm, we have implemented a simple random item generator that could generate a sequence of random items with a custom defined frequency distribution. Generated sequences of items were given to the input of our frequent event mining algorithm implementation. The process lasted until several millions of items were analysed and at the end the frequent category set was printed out. We also measured the maximum size $r_{\max}$ of the frequent set during the program execution and calculated statistics of filter state values in the sketch.

At first we checked if the algorithm was properly implemented and if it really detected frequent categories. The Zipf distribution with exponents $s \in \{1, 2\}$ were used. The categories were given a rank so that the most frequent had the rank 1, and the last had the rank N. The program printed out only the first few categories from the set, so we concluded it worked. By setting various parameters, we observed how the sizes of the frequent set changed. For some settings, we could notice some false positives - the categories with very high ranks, which could not become frequent by accident. Some results are shown in Table 1. Every repetition of the program run with same settings did not provide same results every time. Though the first items were always the same, various categories were reported as the categories with the lowest frequencies. This was due to the random character of the test. Because the most recent history of the stream has a high importance in our solution, a generally infrequent item could cross the threshold only for some time near the end of the experiment and be reported as a frequent one. This was proved by reporting the times, when the categories were added to the frequent set. The least frequent reported items were usually added just near the end of the experiment, while the most frequent ones – just after the beginning.

Table 1. Sample results of the test program, $n = 5 \times 10^6$, $N = 10^6$, $f_{\max} = 1000$ Hz, $\tau = 100$ s

s	f_{thr}[Hz]	m	M	$r_{\max}$	result
1	10.0	1000	5	7	(1, 2, 3, 4, 5, 6, 7)
1	5.0	1000	5	15	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13)
1	2.0	1000	5	37	(1, 2, 3, . . . , 24, 26, 25, 27, 28, 30, 29, 32, 31, 33)
1	5.0	250	2	57	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13)
1	5.0	500	3	14	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14)
1	5.0	1500	5	15	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13)
2	2.0	200	2	19	(1, 2, 3, . . . , 9, 10, 11, 12, 13, 14, 15, 16, 17)
2	1.0	200	2	26	(1, 2, 3, . . . , 17, 18, 19, 20, 21, 22, 23, 25



(a) $s = 1$

(b) $s = 2$

Fig. 2. Probability distribution of false positives obtained for a Zipf distribution of categories, for different sizes of the sketch, $n = 5 \times 10^6$, $N = 10^6$, $M = 5$, $f_{\max} = 1000$ Hz, $f_{\text{thr}} = 1$ Hz, $\tau = 100$ s

We have also investigated, how the values of certain parameters affect the results. Probabilities of false positives were estimated basing on the final state of the sketch (Fig. 2). As presumed, the probability of adding an infrequent category with a frequency only a little lower than the threshold is very high, but below some level drops very quickly. Thus the algorithm is especially well suited for distributions, where the number of items with very low frequencies can be huge. The more memory is available for the sketch, the less false positives can slip into the result. We observed a very interesting behaviour when changing the number of stages for the fixed sketch size (Fig. 3). The number of stages set too low may cause that, for the large values of N, the expected value of the number of reported infrequent categories may be high. The probability may not fall enough quickly to compensate the increasing number of categories with low frequencies. On the other hand, setting this number too high may cause that the real frequency threshold may be located far below the original, intended by the user. Note that though the probability is very high above this threshold, in general there is no guarantee it will be equal to 1.

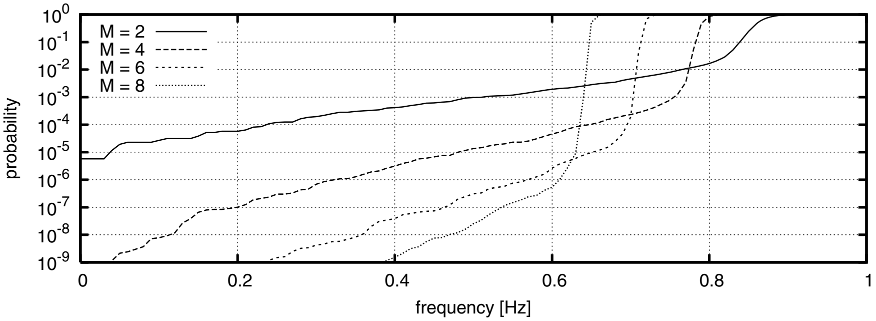


Fig. 3. Probability distribution of false positives obtained for a Zipf distribution of categories, for different number of stages, $s = 1$, $n = 5 \times 10^6$, $N = 10^6$, $m = 5000$, $f_{\max} = 1000$ Hz, $f_{\text{thr}} = 1$ Hz, $\tau = 100$ s

In the end we checked how the result set adapts to the frequency distribution changes over time. The test program significantly lowered the probability of items of one of the frequent categories after a half of iterations. It caused that category to become infrequent. The program correctly removed that category from the frequent set after some time, usually slightly exceeding τ. The same experiment was made with adding a new frequent item. In this case a new item was detected faster.

6 Conclusions

The experiments proved our improved version of the multistage filtering algorithm to work correctly. The algorithm gained ability to give stable results at any time while still using very little memory just as in the original algorithm described by Estan et al. [7]. Though it does not implement a standard, sharp-edged sliding window behaviour, the fuzzy, exponential window seems to be useful wherever there is a need to continuously monitor the frequent item set.

We think the modified algorithm is capable of being implemented in hardware, with small static associative memory, and can be employed to analyze network traffic at very high data rates, without a need for data packet sampling [11]. The exponential function in the low-pass filter would not be a problem, while good fast hardware implementations exists [12]. Unfortunately, we did not have a possibility to implement our ideas in hardware, so this remains an open research problem.

References

1. Lan, K., Heidemann, J.: A measurement study of correlations of internet flow characteristics. Comput. Networks 50(1), 46–62 (2006)
2. Boyer, R.S., Moore, J.S.: MJRTY: A fast majority vote algorithm. Technical Report 35, Institute of Computer Science, Texas University (1981)
3. Misra, J., Gries, D.: Finding repeated elements. Technical report, Cornell University, Ithaca, NY, USA (1982)
4. Demaine, E.D., López-Ortiz, A., Munro, J.I.: Frequency estimation of internet packet streams with limited space. In: Möhring, R.H., Raman, R. (eds.) ESA 2002. LNCS, vol. 2461, pp. 348–360. Springer, Heidelberg (2002)
5. Manku, G., Motwani, R.: Approximate frequency counts over data streams. In: Proceedings of the 28th International Conference on Very Large Data Bases, Hong Kong, China, August 2002 (2002)
6. Charikar, M., Chen, K., Farach-Colton, M.: Finding frequent items in data streams. In: Proceedings of the 29th International Colloquium on Automata, Languages, and Programming (2002)
7. Estan, C., Varghese, G.: New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. ACM Trans. Comput. Syst. 21(3), 270–313 (2003)
8. Chang, J.H., Lee, W.S.: estWin: Online data stream mining of recent frequent itemsets by sliding window method. J. Inf. Sci. 31(2), 76–90 (2005)
9. Cormode, G., Muthukrishnan, S.: What's hot and what's not: tracking most frequent items dynamically. ACM Trans. Database Syst. 30(1), 249–278 (2005)
10. Kołaczkowski, P.: Using low-pass signal filtering for continuous database load estimation, submitted to BDAS'07 conference, Ustroń, Poland (2007)
11. Gibbons, P.B., Matias, Y.: New sampling-based summary statistics for improving approximate query answers. In: SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data, pp. 331–342. ACM Press, New York (1998)
12. Kantabutra, V.: On hardware for computing exponential and trigonometric functions. IEEE Trans. Comput. 45(3), 328–339 (1996)

Outlier Detection: An Approximate Reasoning Approach

Tuan Trung Nguyen

Polish-Japanese Institute of Information Technology
ul. Koszykowa 86, 02-008 Warsaw, Poland
nttrung@pjwstk.edu.pl

Abstract. Outliers, defined as data samples markedly different from the rest of their kind, play an important role in modern pattern recognition and data analysis systems. Outlier treatment usually invokes reasoning about the unknown (irregular) using concepts and features pertaining to the known (regular) samples, naturally requires tools for handling uncertainty or ambiguity, incorporates multi-layered approximate reasoning structures, and often relies on an external background knowledge source. Granular Computing and Rough Set theories provide excellent methods and frameworks for such tasks. In this article, we discuss methods for the detection and evaluation of outliers, as well as how to elicit background domain knowledge from outliers using multi-level approximate reasoning schemes.

Keywords: Outlier, granular computing, rough set, approximate reasoning, concept approximation.

1 Introduction

Conceptually, *outliers/exceptions* are kind of atypical samples that stand out from the rest of their group or behave very differently from the norm [1]. These samples previously would usually be treated as bias or noisy input data and were frequently discarded or suppressed in subsequent analyses. However, the rapid development of Data Mining, which aims to extract from data as much knowledge as possible, has made outlier identification and analysis one of its principal branches. Dealing with outliers is crucial to many important fields in real life such as fraud detection in electronic commerce, intrusion detection, network management, or even space exploration.

Most popular measures to detect outliers [4] are based on either probabilistic density analysis [2] or distance evaluation [8]. Knorr made an attempt to elicit intensional knowledge from outliers through the analysis of the dynamicity of outliers' set against changes in attribute subsets [7]. However, no thorough model or scheme for the discovery of intensional knowledge from identified outliers has been established. In particular, there is almost no known attempt to develop methods for outlier analysis amongst structured objects, i.e. objects that display strong inner dependencies between theirs own features or components. Perhaps

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 495–504, 2007.

the reason for this is the fact that while many elaborated computation models for the detection of outliers have been proposed, their effective use in eliciting additional domain knowledge, as well as the elicitation of intensional knowledge within outliers, is believed difficult without support of a human expert.

In this paper, we propose a framework for outlier detection and analysis based on the Granular Computing paradigm, using tools and methods originated from Rough Set and Rough Mereology theories. The process of outlier detection is refined by the evaluation of classifiers constructed employing intensional knowledge elicited from suspicious samples. We show the role of an external domain knowledge source by human experts in outlier analysis, and present methods for the successful assimilation of such knowledge.

2 Ontology Matching and Knowledge Elicitation

A typical machine learning system attempts to build a data model that fits a provided training sample collection. This model constructing process can be facilitated with additional domain knowledge, most typically provided by an external human expert, about the samples. The knowledge on training samples that comes from an expert obviously reflects his perception about the samples. The language used to describe this knowledge is a component of the expert's ontology which is an integral part of his perception. In a broad view, an ontology consists of a vocabulary, a set of concepts organized in some kind of structures, and a set of binding relations amongst those concepts [3]. We assume that the expert's ontology when reasoning about complex structured samples will have the form of a multi-layered hierarchy, or a *lattice*, of concepts. A concept on a higher level will be synthesized from its children concepts and their binding relations. The reasoning thus proceeds from the most primitive notions at the lowest levels and work bottom-up towards more complex concepts at higher levels.

2.1 External Knowledge Transfer

The knowledge elicitation process assumes that samples, for which the learning system deems it needs additional explanations, are submitted to the expert, which returns not only their correct class identity, but also an explanation on *why*, and perhaps more importantly, *how* he arrived at his decision. This explanation is passed in the form of a rule:

$$[CLASS(u) = k] \equiv \Im(EFeature_1(u), ..., EFeature_n(u))$$

where $EFeature_i$ represents the expert's perception of some characteristics of the sample u, while synthesis operator $\Im$ represents his perception of some relations between these characteristics. In a broader view, $\Im$ constitutes of a *relational structure* that encompasses the hierarchy of experts' concepts expressed by $EFeature_i$.

The ontology matching aims to translate the components of the expert's ontology, such as $EFeature_i$ and binding relations embedded in the $\Im$ structure, expressed in the foreign language L_f, into the patterns (or classifiers) expressed in a language familiar to the learning system, e.g:

- $[FaceType(Ed) = `Square']$ ≡(Ed.Face().Width - Ed.Face().Height) $\leq$ 2cm
- $[Eclipse(p) = `True']$ ≡ (s=p.Sun())∧(m=p.Moon())∧(s∩m.Area$\geq$ s.Area·0.6)

As the human perception is inherently prone to variation and deviation, the concepts and relations in a human expert's ontology are approximate by design. To use the terms of granular computing, they are information granules that encapsulate the autonomous yet interdependent aspects of human perception. The matching process, while seeking to accommodate various degrees of variation and tolerance in approximating those concepts and relations, will follow the same hierarchical structure of the expert's reasoning. This allows parent concepts to be approximated using the approximations of children concepts, essentially building a *layered approximate reasoning scheme*. Its hierarchical structure provides a natural realization of the concept of granularity, where nodes represent clusters of samples/classifiers that are similar within a degree of resemblance/functionality, while layers form different levels of abstraction/perspectives on selected aspects of the sample domain.

On the other hand, with a such established multi-layered reasoning architecture, we can take advantages of the results obtained within the Granular Computing paradigm, which provides frameworks and tools for the fusion and analysis of compound information granules from previously established ones, in a straightforward manner. The intermediate concepts used by external experts to explain their perception are vague and ambiguous, which makes them natural subjects to granular calculi.

The translation must

- allow for a flexible matching of a variations of similar domestic patterns to a foreign concept, i.e. the translation result should not be a single patterns, but rather a collection or cluster of patterns.
- find approximations for the foreign concepts and relations, while preserving their hierarchical structure. In other words, inherent structure of the provided knowledge should be intact.
- ensure robustness, which means independence from noisy input data and incidental underperformance of approximation on lower levels, and stability, which guarantees that any input pattern matching concepts on a lower level to a satisfactory degree will result in a satisfactory target pattern on the next level.

We assume an architecture that allows a learning system to consult a human expert for advices on how to analyze a particular sample or a set of samples. Typically this is done in an iterative process, with the system subsequently incorporating knowledge elicited on samples that could not be properly classified in previous attempts.

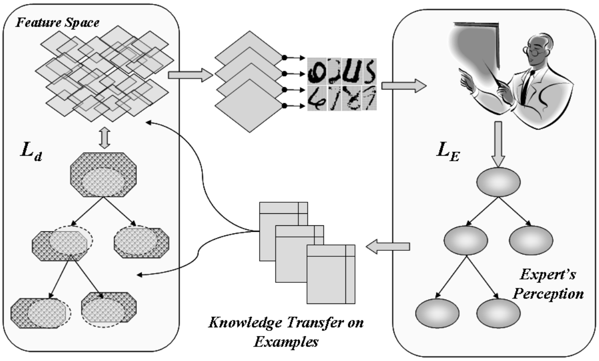


Fig. 1. Expert's knowledge elicitation

2.2 Approximation of Concepts

A foreign concept C is approximated by a domestic pattern (or a set of patterns) p in term of a rough inclusion measure $Match(p, C) \in [0, 1]$. Such measures take root in the theory of rough mereology [14], and are designed to deal with the notion of inclusion to a degree. An example of concept inclusion measures would be:

$$Match(p, C) = \frac{|\{u \in T : Found(p, u) \wedge Fit(C, u)\}|}{|\{u \in T : Fit(C, u)\}|}$$

where T is a common set of samples used by both the system and the expert to communicate with each other on the nature of expert's concepts, $Found(p, u)$ means a pattern p is present in u and $Fit(C, u)$ means u is regarded by the expert as fit to his concept C.

Our principal goal is, for each expert's explanation, find sets of patterns Pat, Pat_1,...,Pat_n and a relation $\Im_d$ so as to satisfy the following *quality requirement*:

$$\textbf{if } (\forall i : Match(Pat_i, EFeature_i) \geq p_i) \wedge (Pat = \Im_d(Pat_1, ..., Pat_n)) \\ \textbf{then } Quality(Pat) > \alpha$$

where $p_i : i \in \{1, .., n\}$ and α are certain cutoff thresholds, while the $Quality$ measure, intended to verify if the target pattern Pat fits into the expert's concept of sample class k, can be any, or combination, of popular quality criteria such as *support*, *coverage*, or *confidence* [15].

In other words, we seek to translate the expert's knowledge into the domestic language so that to generalize the expert's reasoning to the largest possible number of training samples. More refined versions of the inclusion measures would involve additional coefficients attached to e.g. $Found$ and Fit test function. Adjustment of these coefficients based on feedback from actual data may help optimize the approximation quality.

For example, let's consider a handwritten digit recognition task:

When explaining his perception of a particular digit image sample, the expert may employ concepts such as *'Circle'*, *'Vertical Strokes'* or *'West Open Belly'*. The expert will explain what he means when he says, e.g. '*Circle*', by providing a decision table (U, d) with reference samples, where d is the expert decision to which degree he considers that '*Circle*' appears in samples $u \in U$. The samples in U may be provided by the expert, or may be picked up by him among samples explicitly submitted by the system, e.g. those that had been misclassified in previous attempts.

The use of rough inclusion measures allows for a very flexible approximation of foreign concept. A stroke at 85 degree to the horizontal in a sample image can still be regarded as a vertical stroke, though obviously not a 'pure' one. Instead of just answering in a 'Yes/No' fashion, the expert may express his degrees of belief using such natural language terms as '$Strong$', '$Fair$', or '$Weak$'.

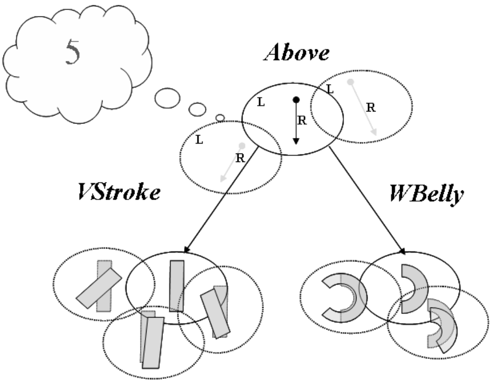


Fig. 2. Tolerant matching by expert

Table 1. Perceived features

	$Circle$
u_1	$Strong$
u_2	$Weak$
...	...
u_n	$Fair$

Table 2. Translated features

	$DPat$	$Circle$
u_1	252	$Strong$
u_2	4	$Weak$
...	...	...
u_n	90	$Fair$

The expert's feedback will come in the form of a decision table (See Table 1.):

The translation process attempts to find domestic feature(s)/pattern(s) that approximate these degrees of belief. Domestic patterns satisfying the defined quality requirement can be quickly found, taking into account that sample tables submitted to experts are usually not very large. Since this is essentially a rather simple learning task that involves feature selection, many strategies can be employed. In [11], genetic algorithms equipped with some greedy heuristics are reported successful for a similar problem. Neural networks also prove suitable for effective implementation.

Having approximated the expert's features $EFeature_i$, we can try to translate his relation $\Im$ into our $\Im_d$ by asking the expert to go through U and provide us with the additional attributes of how strongly he considers the presence of $EFeature_i$ and to what degree he believes the relation $\Im$ holds. Again, lets consider the handwritten recognition case.(See Tab. 3).

Table 3. Perceived relations

	VStroke	*WBelly*	*Above*
u_1	*Strong*	*Strong*	*Strong*
u_2	*Fair*	*Weak*	*Weak*
...	...	...	...
u_n	*Fair*	*Fair*	*Weak*

Table 4. Translated relations

	#V_S	#NES	$S_y < B_y$	*Above*
u_1	0.8	0.9	(*Strong*,1.0)	(*Strong*, 0.9)
u_2	0.9	1.0	(*Weak*, 0.1)	(*Weak*, 0.1)
...	...	...	...	...
u_n	0.9	0.6	(*Fair*, 0.3)	(*Weak*, 0.2)

We then replace the attributes corresponding to $EFeature_i$ with the rough inclusion measures of the domestic feature sets that approximate those concepts (computed in the previous step). In the next stage, we try to add other features, possibly induced from original domestic primitives, in order to approximate the decision d. Such a feature may be expressed by $S_y < B_y$, which tells whether the median center of the stroke is placed closer to the upper edge of the image than the median center of the belly. (See Tab. 4)

The expert's perception *"A '6' is something that has a 'vertical stroke' 'above' a 'belly open to the west'"* is eventually approximated by a classifier in the form of a rule:

$$\textbf{if } S(\#\text{BL\_SL} > 23) \textbf{ AND } B(\#\text{NESW} > 12\%) \textbf{ AND } S_y < B_y \textbf{ then } \text{CL='6'},$$

where S and B are designations of pixel collections, #BL_SL and #NESW are numbers of pixels with particular topological feature codes, and $S_y < B_y$ reasons about centers of gravity of the two collections.

Approximate reasoning schemes embody the concept of information granularity by introducing a hierarchical structure of abstraction levels for the external knowledge that come in the form of a human expert's perception. The granularity helps to reduce the cost of the knowledge transfer process, taking advantage of the expert's hints. At the same time, the hierarchical structure ensures to preserve approximation quality criteria that would be hard to obtain in a flat, single-level learning process.

3 Outlier Identification

As mentioned in Section 1, most existing outlier identification methods employ either probabilistic density analysis, or distance measures evaluation. Probabilistic approach typically run a series of statistical discordancy tests on a sample to determine whether it can be qualified as an outlier. Sometimes this procedure is enhanced by a dynamic learning process. Their main weakness is the assumption of an underlying distribution of samples, which is not always available in many real life applications. Difficulties with their scalability in numbers of samples and dimensions are also a setback of primary concern. Another approach to outlier detection relies on certain distance measures established between samples. Known methods are data clustering and neighborhood analysis. While this approach can be applied to data without any assumed a priori distribution, they usually entails significant computation costs.

Let C_k be a cluster of samples for class k and d_k be the distance function established for that class. For a given cut-off coefficient $\alpha \in (0, 1]$, a sample u^* of class k is considered “difficult”, “hard” or “outlier” if, e.g:

$$d_k(u^*, C_k) \geq \alpha \cdot \max\{d_k(v, C_K) : CLASS(v) = k\}$$

which means u^* is far from the “norm” in term of its distance to the cluster center, or

$$|\{v : v \in C_k \wedge d_k(u^*, v) \leq d_k(v, C_k)\}| \leq \alpha \cdot |C_k|$$

which means u^* is amongst the most outreaching samples of the cluster.

Another popular definition of outlier is:

$$|\{v : v \in C_k \wedge d_k(u^*, v) \geq D\}| \leq \alpha \cdot |C_k|$$

which means at least a fraction α of objects in C_k lies in a greater distance than D from u^*.

It can be observed that both approaches pay little attention to the problem of eliciting intensional knowledge from outliers, meaning no elaborated information that may help explain the reasons why a sample is considered outlier. This kind of knowledge is important for the validity evaluation of identified outliers, and certainly is useful in improving the overall understanding of the data. Knorr and Ng made an attempt to address this issue by introducing the notion strength of outliers, derived from an analysis of dynamicity of outlier sets against changes in the features' subsets [7]. Such analyses belong to the very well established application domain of Rough Sets, and indeed a formalization of a similar approach within the framework of Rough Sets has been proposed by [5].

Our approach to outlier detection and analysis will assume a somewhat different perspective. It focuses on two main issues:

1. Elicitation of intensional knowledge from outliers by approximating the perception of external human experts.

2. Evaluation of suspicious samples by verification the performance of classifiers constructed using knowledge elicited from these samples.

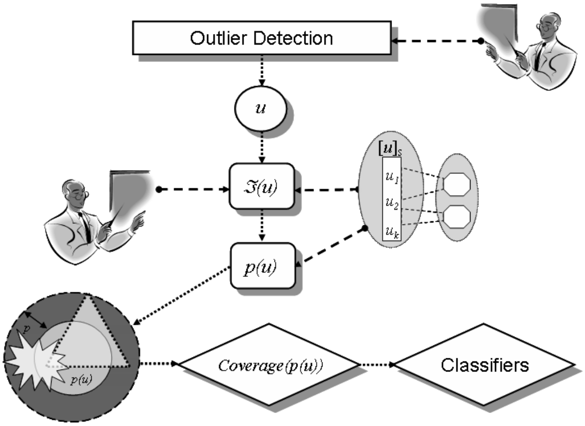


Fig. 3. Outlier analysis scheme

Having established a mechanism for eliciting expert's knowledge as described in previous sections, we can develop outlier detection tests that are completely independent from the existing similarity measures within the learning system as follows:

For a given training sample u^*,
Step 1. We ask the expert for his explanation on u^*.
Step 2. The expert provides a foreign knowledge structure $\Im(u^*)$.
Step 3. We approximate $\Im(u^*)$ under restrictive matching degrees to ensure only the immediate neighborhood of u^* is investigated. Let's say the result of such an approximation is a pattern (or set of patterns) p_u^*.
Step 4. It is now sufficient to check $Coverage(p_u^*)$. If this coverage is high, it signifies that u^* may bear significant information that is also found in many other samples. The sample u^* therefore cannot be regarded as an outlier despite the fact that there may not be many other samples in its vicinity in terms of existing domestic distance measures of the learning system.

This test shows that distance-based outlier analysis and expert's elicited knowledge are complementary to each other.

In our architecture, outliers may be detected as samples that defied previous classification efforts, or samples that pass the above described outlier test, but may also be selected by the expert himself. In this way, we can benefit from the best of both sources of knowledge.

4 Experiments

In order to illustrate the developed methods, we conducted a series of experiments on the NIST Handwritten Segmented Character Special Database 3. We compared the performances gained by a standard learning approach with and without the aid of the domain knowledge. The additional knowledge, passed by a human expert on popular classes as well as some atypical samples allowed to reduce the time needed by the learning phase from 205 minutes to 168 minutes, which means an improvement of about 22 percent without loss in classification quality. In case of screening classifiers, i.e. those that decide a sample *does not* belong to given classes, the improvement is around 60 percent. The representational samples found are also slightly simpler than those computed without using the background knowledge.

Table 5. Comparison of performances

	No domain knowledge	With domain knowledge	Gain
Total learning time	205s	168s	22%
Negative classifier learning time	3.7s	2.2s	40%
Positive classifier learning time	28.2s	19.4s	31%
Skeleton graph size	3-5 nodes	2-5 nodes	

5 Conclusion

We presented in details an approach to the problem of outlier detection and analysis in data from a machine learning perspective. We focus on the elicitation of intensional knowledge from outliers using additional background information provided by an external human expert. We described an interactive scheme for the knowledge transfer between the expert and the learning system, using methodologies and tools originated from Granular Computing paradigm, Rough Set and Rough Mereology theories, as well as techniques pertaining to the approximate reasoning schemes. Proposed approach proves capable to yield effective implementations and offers a complementary perspective compared with other existing approaches in the field of outlier detection and analysis.

References

1. Aggarwal, C.C., Yu, P.S.: Outlier detection for high dimensional data. In: Charu, C. (ed.) Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, pp. 37–46. ACM Press, New York (2001)
2. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. SIGMOD Rec. 29(2), 93–104 (2000)
3. Fensel, D.: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer, New York (2003)
4. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artif. Intell. Rev. 22(2), 85–126 (2004)
5. Jiang, F., Sui, Y., Cao, C.: Outlier detection using rough set theory. In: 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, 2005, pp. 79–87, Regina, Canada (2005)

6. Knorr, E.M.: Outliers and Data Mining: Finding Exceptions in Data. PhD thesis, University of British Columbia (April 2002)
7. Knorr, E.M., Ng, R.T.: Finding intensional knowledge of distance-based outliers. In: Edwin, M. (ed.) VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases, San Francisco, CA, USA, 1999, pp. 211–222. Morgan Kaufmann Publishers, San Francisco (1999)
8. Knorr, E.M., Ng, R.T., Tucakov, V.: Distance-based outliers: Algorithms and applications. The. VLDB Journal 8(3), 237–253 (2000)
9. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
10. Nguyen, T.T.: Eliciting domain knowledge in handwritten digit recognition. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 762–767. Springer, Heidelberg (2005)
11. Oliveira, L.S., Sabourin, R., Bortolozzi, F., Suen, C.Y.: Feature selection using multi-objective genetic algorithms for handwritten digit recognition. In: International Conference on Pattern Recognition (ICPR02), pp. I: 568–571 (2002)
12. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Norwell, MA (1992)
13. Pedrycz, W. (ed.): Granular computing: an emerging paradigm. Physica-Verlag, Heidelberg (2001)
14. Polkowski, L., Skowron, A.: Rough mereology: A new paradigm for approximate reasoning. Journal of Approximate Reasoning 15(4), 333–365 (1996)
15. Polkowski, L., Skowron, A.: Constructing rough mereological granules of classifying rules and classifying algorithms. In: Bouchon-Meunier, B., et al. (ed.) Technologies for Constructing Intelligent Systems I, pp. 57–70. Physica-Verlag, Heidelberg (2002)
16. Skowron, A.: Rough sets in perception-based computing. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, Springer, Heidelberg (2005)
17. Zadeh, L.A.: From imprecise to granular probabilities. Fuzzy Sets. and Systems 154(3), 370–374 (2005)

Discovering Compound and Proper Nouns*

Grzegorz Protaziuk[1], Marzena Kryszkiewicz[1], Henryk Rybinski[1],
and Alexandre Delteil[2]

[1] ICS, Warsaw University of Technology
[2] France Telecome R & D
hrb,mkr,gprotazi@ii.pw.edu.pl, alexandre.delteil@orange-ft.com

Abstract. The identification of appropriate text tokens (words or sequences of words representing concepts) is one of the most important tasks of text preprocessing and may have great influence on the final results of text analysis. In our paper, we introduce a new approach to discovering compound nouns, including proper compound nouns. Our approach combines the data mining methods with shallow lexical analysis. We propose a simple pattern language for specifying grammatical patterns to be satisfied by extracted compound nouns. Our method requires annotating the words with part of speech tags, thus to this extent, it is language-dependent. Based on the data mining GSP algorithm, we propose T-GSP as its modification for extracting frequent text patterns, and in particular, frequent word sequences that satisfy given grammatical rules. The obtained sequences are regarded as candidates for compound nouns. The experiments have proven very high quality of the method.

Keywords: multiword terms, compound nouns, proper nouns, frequent word sequences, frequent text patterns, text mining.

1 Introduction

Multiword units are sequences of two or more words which occur together in text and form syntactic or lexical expression. Multiword units consist of words that may appear in sentences successively without any gaps, or may be separated by one or more words. Several groups of multiword expressions may be distinguished, namely [11]:

- compound nouns, which represent notions, e.g. “information retrieval”. This group also includes proper nouns, e.g. “Warsaw University of Technology”. The expressions belonging to this category have a rigid syntactic structure.
- idioms – expressions, the meaning of which almost never can be derived from the meaning of the words constituting them. In the case of idioms there is a little possibility of modifying the syntax.
- collocations – this class consists of associated words, i.e. words that frequently co-occur in text. Usually, it is impossible to replace a term used in a collocation by its synonym without changing the meaning.

* The work has been performed within the project granted by France Telecom.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 505–515, 2007.

In many text mining tasks, the proper identification and extraction of text units may significantly influence the quality or/and usefulness of the final results of the analysis, where particularly nouns play an essential role. Having only nouns appearing in a text, one can guess a topic discussed in the text. In the classification or clustering tasks, a single proper noun could be even a better attribute than a group of several words belonging to other parts of speech. Hence, proper and compound nouns are important in building semantic dictionaries or ontologies.

In our paper, we introduce a new approach to discovering compound nouns, including proper compound nouns. Our approach combines the data mining methods with shallow lexical analysis. We propose a simple pattern language for specifying grammatical patterns to be satisfied by extracted compound nouns. Clearly, the method requires annotating the words with part of speech tags (POS), thus to this extent, it is language-dependent. Based on the data mining GSP algorithm [16], we propose T-GSP as its modification for extracting frequent word sequences that satisfy given grammatical rules. The obtained sequences are regarded as candidates for compound nouns. The experiments have proven very high quality of the method.

In Section 2 we overview briefly related work. Our method of discovering compound nouns is presented in Section 3. The experiments are reported in Section 4. Section 5 concludes the obtained results.

2 Related Work

The methods for automatic or semi-automatic multiword unit extraction can be divided into four following categories: syntactical, statistical, syntactic-statistical and, recently developed, text-mining approaches.

The first category includes methods in which deep morphological analysis of texts is performed. In order to find multiword expressions, the specific syntactic structures as noun phrases are identified and investigated. The methods allow obtaining good results [3, 8], but require highly specialized linguistic procedures, which makes them strongly language dependent, very time-consuming, and very difficult for development.

The second category methods are based on purely statistical analysis. The methods extract multiword terms from text corpora based on regularities discovered from investigated documents. Here, the preprocessing of text documents is limited merely to the extraction of single terms and splitting text into windows – text units consisting of a given number of terms and/or sentences. It makes statistical methods flexible and independent from the domain or language. In those methods, the sequences of n terms (or n-grams) are generated. Often the length of n-grams should be restricted, because of (1) the danger of combinatorial explosion of sequences, and (2) the evidence that most of the lexical relations relate words separated by at most 5 other words [15]. For the evaluation of the found collocations, in addition to the frequency threshold, various additional measures are applied. In [5], a point-wise mutual information measure was proposed. In [14], the entropy of the nearest context was used for pruning multiword units. The method of using synonyms for evaluating the strength of collocations was introduced in [9]. The "strength" of a collocation was evaluated based on a difference between the frequency of the collocation with given word and the frequency of the collocation in which the word was replaced by its synonyms. In [15], the two-phase method was presented. In the first stage, the bi-grams are selected by analyzing the distribution of the frequency of the word collocates within the 5-words neighborhood in the sentences (appearing both, before, and after the word). For pruning, the z-score measure was applied. The measure is computed based on the average frequency and standard deviation of the frequencies. In the second phase, bi-grams found in the phase 1 are expanded to n-grams by analyzing frequencies of other words in the sentences, in which given bi-gram appears. The comparison of different approach and measures was presented in [10, 12].

The third category includes methods in which both, lexical and statistical analysis are used. The approach in which deep syntax analysis of sentences and statistical tests are combined is presented in [13]. In the method, the bi-grams are extracted, based on the robust syntactic parser and then n-grams are incrementally generated from already obtained n-grams. The elimination of non-interesting collocations is performed based on the statistical test of log-likelihood ratios. In [7], the system that combines the shallow lexical analysis (part of speech tagging) and statistical methods is described. In the method, the positional n-grams are extracted from the 7 word size window context. Both, word n-grams and part-of-speech tagged n-grams are generated. The combination of mutual expectation measures concerning the word n-grams and tagged n grams are used for pruning insufficiently cohesive collocations.

The methods belonging to the fourth category involve data and text mining techniques. In [3], the problem of extracting compound nouns was investigated. The authors described a structure of company names in the form of a regular expression, then extracted three categories of elements creating such names, and prepared rules allowing the classification of a term to a proper category. The rules were annotated by the following tags: capital or lower case, and the category. The extraction of names were performed in two phases: (1) learning from the annotated terms, (2) using a pattern and extracted terms for generating names of companies. In [1, 2], methods for discovering frequent sequences have been adapted to discovering frequent word sequences. The authors described the representation of text documents to be used by the method, and provided the algorithms for discovering maximal frequent word sequences.

In our paper, we propose a new approach for extracting compound and proper compound nouns, which is also based on the data mining approach. However, we have attempted to make it with a simpler and more efficient approach, by combining efficient data mining algorithms with a shallow lexical analysis. We have adopted the GSP algorithm, which seemed to be appropriate for many text mining tasks. The main goal was to adopt GSP to extract compound (proper) nouns. We prove in the paper that it allows obtaining results of very high quality in a reasonable amount of time.

3 Proposed Approach

We start the presentation by recalling basic notions of discovering generalized sequential patterns from data sequences [16]. Each *data sequence* is an ordered list of transactions. Originally, any *transaction* consists of a set of *items*, and a *transaction time* (or *index*). A *sequence* is a list of sets of items. We say that a data sequence D *supports* a sequence S, if S occurs in D. If sequence S is supported by more data sequences than a predefined threshold, it is called *frequent.* In [16], it is alternatively called a *sequential pattern.* In [16], the authors generalize this notion by introducing time constraints and taxonomies.

3.1 Frequent Text Patterns

One can view a text as a special source of sequential patterns, from which frequent patterns of interest could be extracted. We expect that such frequent patterns could be a base for discovering compound and proper compound nouns. In the sequel, we present our framework for discovering such nouns by means of frequent text patterns.

Let $W = \{w_1, \dots, w_m\}$ be a set of words (or tokens) $w_1, \dots, w_m$, and T_W be a set of taxonomies defined over W. A *word sequence* $s =< v_1, \dots, v_k >$, where $v_i \in W$, is a list of words $v_1, \dots, v_k$. A given word may occur more than once in a word sequence. A *text pattern* $tp =< s_1, \dots, s_l >$ is a list of word sequences $s_1,\ s_2, \dots, s_l$. In our paper, we treat a sentence as a word sequence, and text repository, as a set of sentences.

Given a word sequence, one can calculate a distance between words in the sequence. The distance between two words w_1 and w_2 in sequence s, denoted as $dist_s(w_1, w_2)$, is defined as the number of words between w_1 and w_2 in s.

Example 1. Let us consider a sequence <*It is very interesting problem*>, received from a text by removing articles and punctuation marks. The distance between the words "*It*" and "*interesting*" is equal to 2, and the distance between "*is*" and "*very*" equals 0.

A word v *precedes* w, denoted by $v \geq w$, if the words are the same or v is an ancestor of w in the taxonomy T_W. Complementary, w is preceded by v, denoted by $w \leq v$, if the words are the same or w is a successor of v in the taxonomy T_W. We say that a word sequence $s_1 =< w_1, \dots, w_m >$ *contains* a word sequence $s_2 =< v_1, \dots, v_n >$, denoted as $s_1 \subseteq s_2$, if there are integers $i_1 < \dots < i_n$ such that $w_1 \leq v_{i1}, \dots, w_n \leq v_{in}$. A text pattern $tp_1 =< s^1_1, \dots, s^1_m >$ contains a text pattern $tp_2 =< s^2_1, \dots, s^2_n >$ (without constraints), if there are integers $i_1 < \dots < i_n$, such that $s^2_1 \subseteq s^1_{i_1}, \dots, s^2_n \subseteq s^1_{i_n}$.

We say that a document d *supports* a text pattern tp, if d contains tp. If tp is supported by more documents than a predefined threshold, it is called *frequent.* Table 1 summarizes equivalence between the basic notions of data patterns as defined in [16] and text patterns.

Now, we introduce *time constraints* for a frequent text pattern, namely *window size, min-gap, max-gap and maxWord-gap.* The meaning of these parameters is presented in Table 2.

Table 1. Frequent data patterns versus frequent text patterns

Frequent data pattern	*Frequent text pattern*
item	word
transaction – set of items	sentence – list of words
data sequence – list of transactions	document (or paragraph) – list of sentences
data set – set of data sequences	text repository – set of documents

Let us note that *maxWord-gap* was not used in [16]. Formally, *maxWord-gap* constraint can be expressed as follows: the sentence $snt =< w_1, w_2, \ldots, w_m >$ contains a word sequence $s =< v_1, v_2, \ldots, v_n >$, if there are integers $i_1 < \ldots < i_n$ such that $w_1 \leq v_{i_1}, \ldots, w_n \leq v_{i_n}$, and $dist_{snt}(w_i, w_{i+1}) \leq maxWord - gap$, $1 \leq i \leq n-1$.

Example 2. Let us consider a sentence *snt* =<*It is very interesting problem*>. Let *maxWord-gap* be set to 1, and *s* =<*is problem*>. As $dist_{snt}(is, problem) = 2$, which is greater than *maxWord-gap*, so *snt* does not contain *s*.

In the generalized framework, we say that a document $d =< snt_1, \ldots, snt_m >$ supports a text pattern $tp =< s_1, \ldots, s_n >$, if the constraints imposed on *window-size*, *max-gap*, *min-gap*, *maxWord-gap* for *d* and *tp* are fulfilled.

In the sequel, we focus on discovering compound nouns, including proper compound nouns. As candidates for such nouns we consider each frequent text pattern consisting of exactly one word sequence.

Table 2. Time constraints for frequent text patterns, wrt document *d*

Name of parameter	Description
window-size	It indicates the maximum number of sentences in *d* in which a given word sequence, being a part of a frequent pattern, should be present.
max-gap	This constraint is posed on every two consecutive word sequences s_1 and s_2 of a frequent text pattern. It is satisfied for s_1 and s_2 in *d*, if maximal number of sentences in *d* between sentences containing s_1 and s_2, respectively, does not exceed *max-gap*.
min-gap	This constraint is also posed on every two consecutive word sequences s_1 and s_2 of a frequent text pattern. It is satisfied for s_1 and s_2 in *d*, if minimal number of sentences in *d* between sentences containing s_1 and s_2, respectively, exceeds *min-gap*.
maxWord-gap	It defines a maximal gap between two consecutive words in a word sequence *s* being a part of a frequent text pattern. A sentence *snt* in *d* contains *s*, if snt includes all the words from *s*, and the number of words between any two consecutive words in *s* within *snt* is less than *maxWord-gap*.

3.2 Grammatical Patterns

A large number of frequent text patterns may be of little or no interest to users. In this section, we propose particular templates for specifying required properties of word sequences in the text patterns of interest. The templates we propose have a form of grammatical patterns specifying allowed parts of speech of words at each position of word sequences. We call a *grammatical pattern* a sequence $gp =< POS^1, \ldots, POS^n >$, where $POS^i = \{pos_{i1}, \ldots, pos_{ik}\}$ is a non-empty set of parts of speech $pos_{i1}, \ldots, pos_{ik}$.

A *word sequence s meets a grammatical pattern gp*, if for each word w_i in s, the part of speech of $pos(w_i) \in POS^i$.

Example 3. Let us consider the grammatical pattern $< \{noun\}, \{preposition\}, \{noun\} >$. The word sequence: <element of car> supports the specified pattern, whereas the word sequence: <*house has roof*> does not, because "has" is not a preposition.

We say that *a sentence* $snt =< w_1, w_2, \ldots, w_m >$ *supports the grammatical pattern* $gp =< POS^1, POS^2, \ldots, POS^n >$, if there is at least one word sequence contained in *snt* that meets *gp*.

Grammatical patterns are useful for finding compound (proper) nouns. Many compound (proper) nouns consist of only nouns and a preposition, e.g. phrases: "*Warsaw University of Technology*" and "*Warsaw School of Economics*" meet the pattern: $< \{noun\}, \{noun\}, \{preposition\}, \{noun\} >$. The usage of grammatical patterns, not only limits the number of discovered text patterns to those of interest, but also makes the process of finding interesting word sequences much more efficient in terms of time and quality of discovered compound (proper) nouns.

In general, the word sequences constituting compound (proper) nouns occur one by one in texts. Usually, there are no other words (gaps) between them, or there is at most one word. Let us consider the sentence *snt* = "*In Warsaw, in the buildings belonging to University there are auditoria with audio-visual equipment of advanced technology*", the grammatical pattern $gp =< \{\text{noun}\}, \{\text{noun}\}, \{\text{preposition}\}, \{\text{noun}\} >$, and the proper noun *pn* = "*Warsaw University of Technology*". Clearly, *pn* is a word sequence that is contained in *snt* and meets *gp*. This means that *snt* supports *pn*, although *pn* does not follow from *snt* logically. The reason of the incorrect reasoning was lack of a restriction on the distance between consecutive words in *pn*.

In order to avoid such situations, we extend grammatical patterns with a constraint specifying maximal gaps that are allowed between two words in a sentence, and incorporate this modification into the definition of a support. The modified definitions of a grammatical pattern and support are as follows:

We call a *grammatical pattern* a pair $[P, G]$, where $P =< POS^1, POS^2, \ldots, POS^n >$, where POS^i is a non-empty set of allowed parts of speech, and $G = \{max\_gap_1, max\_gap_2, \ldots, max\_gap_{n-1}\}$, where max_gap_i is a number greater than or equal to 0, $i = 1..n-1$.

A *sentence* $snt = < w_1, w_2, \ldots, w_m >$ *supports the grammatical pattern* $gp = [P, G]$, where $P =< POS^1, POS^2, \ldots, POS^n >$, $G = \{max\_gap_1, max\_gap_2, \ldots, max\_gap_{n-1}\}$, if there are integers $i_1 < i_2 < \ldots < i_n$, such that $pos(w_{i1}) \in POS^1, \ldots, pos(w_{in}) \in POS^n$, where $w_{ij} \in snt$, pos(w_{ij}) is the part of speech of w_{ij}, and $dist_{snt}(w_{ij}, w_{ij+1}) \leq max\_gap_{ij}$, $1 \leq j \leq n-1$.

3.3 Algorithm

In order to discover frequent text patterns, the word sequences of which satisfy the imposed grammatical patterns, we devised and implemented a T-GSP algorithm (Text Generalized Sequential Patterns algorithm). T-GSP is based on the widely known GSP algorithm. The basic idea is the same in both algorithms: the sequences consisting of n-elements are generated from previously found sequences of $n-1$ elements. However, the adaptation to dealing with text resulted in the incorporation of the following novel patterns constraints:

- the *maxWord-gap* parameter,
- grammatical patterns.

The application of grammatical patterns causes that the frequency of word sequences is not monotonic unlike the frequency of sets of items. For instance, the word sequence <*University, Technology*> may be infrequent (due to non-fulfillment of the grammatical restrictions), but the sequence <*University, of, Technology*> may be frequent. Hence, the pruning of candidate patterns in T-GSP is not based anymore on frequency, but is based on part of speech tags included in grammatical patterns. In general, T-GSP enables discovering of a general text structure in the form of frequent text patterns, composed of many word sequences. However, in this work, we concentrate on applying T-GSP for discovering compound (proper) nouns, as one word sequence frequent text patterns. For this purpose a special processing mode is available in our implementation of T-GSP.

4 Experiments

The main objective of the performed experiments was to verify the proposed data mining framework of discovering frequent text patterns from the point of view of automatic acquisition of compound (proper) nouns. In the tests performed, we looked for word sequences satisfying grammatical patterns which occur in single sentences of documents, i.e. *window-size* was set to 1. As an input data, we used documents from the Reuters repository, as well as, the scientific papers documents devoted to text mining and ontologies. The granularity for the experiments was set to the paragraph, i.e. any paragraph was considered as a document. The rationale behind using such a granularity for both repositories was as follows:

- Usually, compound (proper) nouns appear in few documents, but even if they appear in one document, they may appear in several paragraphs.

- In the case of scientific papers, we encountered the following problems: for low values of the minimal support, almost each sequence consisting of frequent words is frequent; for high values, few frequent sequences are discovered. Using paragraphs alleviated the problem.

4.1 Applied Grammatical Patterns

In the experiments, we looked for compound (proper) nouns by means of grammatical rules comprising only nouns and nouns with one preposition. In particular, we have applied the following grammatical patterns:

A. Including only nouns:

- $gp1 = [< \{noun\}, \{noun\} >, < 0 >]$ – searching for two consecutive nouns e.g. "Newcastle United",
- $gp2 = [< \{noun\}, \{noun\}, \{noun\} >, < 0, 0 >]$ – searching for 3 consecutive nouns, e.g. "information extraction system".

B. Including nouns and a preposition:

1. searching for phrases such as: "*Institute of Research*" with three various specifications of gaps between the preposition and noun.
- $gp3 = [< \{noun\}, \{preposition\}, \{noun\} >, < 0, 0 >]$,
- $gp4 = [< \{noun\}, \{preposition\}, \{noun\} >, < 0, 1 >]$,
- $gp5 = [< \{noun\}, \{preposition\}, \{noun\} >, < 0, 2 >]$.

2. searching for phrases such as: "*Warsaw University of Research*" with two various specifications of gaps between the words.
- $gp6 = [< \{noun\}, \{noun\}, \{preposition\}, \{noun\} >, < 1, 0, 1 >]$,
- $gp7 = [< \{noun\}, \{noun\}, \{preposition\}, \{noun\} >, < 2, 0, 2 >]$.

4.2 Results

In the first group of experiments, we applied the grammatical patterns gp1 and gp2. As a result, we obtained valuable compound proper nouns, *inter alia* "*President Bill Clinton*", "*Eastern Europe*", "*Columbia Pictures*", as well as, compound nouns, e.g. "*carbon dioxide*", "*credit card*", or "*information retrieval system*". We also obtained, unfortunately, sequences, which are rather not good candidate of a compound noun, e.g. "*http www*" or "*colonial rule*". However, the precision factor was very high, reaching 90%.

In the second group of tests, we have used grammatical patterns $gp3$, $gp4$, and $gp5$. In the Reuters repository, we discovered the proper nouns, such as "*Bank of Japan*", "*Union of Kurdistan*", "*Republic of China*", as well as, common phrases, such as "*thousands of people*", "*end of year*", or a bit more specialized notions, like "*barrels of oil*", or "*rate of percent*". In the case of the scientific papers repository, we found many proper nouns (or parts of them) e.g. "*Workshop on Logics*", "*University of Maryland*", "*Conference on Artificial*", as well as, multiword terms, sometimes very specific, e.g.: "*structures of ontologies*", "*specification of conceptualization*", or "*understanding of domain*".

In the last group of the experiments, we applied the grammatical patterns *gp*6 and *gp*7. The application of those patterns for investigating the Reuters documents resulted mainly in finding the proper nouns, e.g. “*Daiwa Institute of Research*”, or “*Patriotic Union of Kurdystan*”. Also some multiword terms were found, e.g. “*box office during Friday*”.

The outcome of the experiment on the scientific papers repository mainly consisted of parts of names of: (1) conferences e.g.: “*International Conference on Learning*”, “*National Conference on Artificial*”, (2) publications e.g.: “*Proceedings Workshop on Ontology*”, “*Lecture Notes in Computer*”, (3) titles of the papers or some text units within the papers, like “*Formal Ontology in Systems*”, “*Taxonomic Relations from Web*”. This results from a standard form of the papers, as they all include lists of references, and many positions are common for several lists. However, apart from finding such obvious examples, quite a number of good candidates for compound nouns were also indicated, e.g.: “*acquisition hyponyms from text*”, “*core system for german*”. The numbers of compound nouns found for applied grammatical patterns are shown in Table 3.

Table 3. Quantitative description of the experiments

Experiment	**Input data**	**Minimal support**	**Grammatical patterns**	**No. of discovered compound (proper) nouns**
Exp1.1	RPar	7	*gp*1, *gp*2	563
Exp1.2	RDoc	5	*gp*1, *gp*2	744
Exp1.3	PPar	4	*gp*1, *gp*2	1406
Exp2.1	RPar	7	*gp*3	37
			*gp*4	67
			*gp*5	77
Exp2.2	RDoc	5	*gp*3	61
			*gp*4	99
			*gp*5	137
Exp2.3	PPar	4	*gp*3	158
			*gp*4	402
			*gp*5	529
Exp3.1	RPoc	7	*gp*6	13
			*gp*7	21
Exp3.2	RDoc	5	*gp*6	20
			*gp*7	33
Exp3.3	PPar	4	*gp*6	57
			*gp*7	101

RPar – paragraphs extracted from the Reuters documents, RDoc – the Reuters documents, PPar - paragraphs extracted from the scientific papers.

5 Conclusions

We presented a new approach to extracting compound (proper) nouns. In our method, we combined a shallow lexical analysis with the data mining methods, which allowed us to discover frequent text patterns. We have introduced a simple, yet flexible, way of specifying requirements on word sequences in the form of grammatical patterns. With templates defining allowed parts of speech, we were able to extract good candidates for compound (proper) nouns in a very effective, and efficient, manner with our T-GSP algorithm. The presented experimental results related to the discovery of compound (proper) nouns, however, by means of other grammatical rules, and/or the other algorithm parameters, the method enables discovering of other categories of collocations and the document structure. The usefulness of the proposed framework will be subject to further research investigations of the team.

References

1. Ahonen-Myka, H., Doucet, A.: Data mining meets collocations discovery. In: Inquiries into Words, Constraints, and Contexts. CSLI Studies in Computational Linguistics ONLINE. Copestake, Ann, pp. 194–203 (Series Editor)
2. Ahonen-Myka, H.: Discovery of frequent word sequences in text. In: The ESF Exploratory Workshop on Pattern Detection and Discovery in Data Mining, Imperial College, London (2002)
3. Biemann, Ch., Boehm, K., Quasthoff, U., Wolff, Ch.: Automatic Discovery and Aggregation of Compound Names for the use in Knowledge Representation. J. of Universal Comp. Sci. 9(6), 530–541 (2003)
4. Bourigault, D.: Surface grammatical analysis for the extraction of terminological noun phrases. In: Proc. of the 14th conf. on Computational linguistics, vol. 3 (1992)
5. Church, K.W, Hanks, P.: Word association norms, mutual information, and lexicography. Computational Linguistics 16(1), 22–29 (1990)
6. Dias, G., Guilloré, S., Bassano, J.C., Pereira Lopes, J.G.: Combining linguistics with statistics for multiword termextraction: A fruitful association? In: Proc. Recherche d'Informations Assistee par Ordinateur (2000)
7. Dias, G.: Multiword unit hybrid extraction. In: Proc. of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment 18 (2003)
8. Nerima, L., Seretan, V., Wehrli, E.: Creating a Multilingual Collocation Dictionary from Large Text Corpora. In: 10th Conf. of EACL'03, Budapest (2003)
9. Pearce, D.: Synonymy in collocation extraction. In: NAACL 2001 Workshop: WordNet and Other Lexical Resources: Applications, Extensions and Customizations, Carnegie Mellon University, Pittsburgh (2001)
10. Pearce, D.: A comparative evaluation of collocation extraction techniques. In: Proc. 3rd Language Resources Evaluation Conf (2002)
11. Sag, I.A., Baldwin, T., Bond, F., Copestake, A., Flickinger, D.: Multiword Expressions: A Pain in the Neck for NLP. In: Proc. of 3rd Int'l Conf. Intell. Text Proc. and Comp. Linguistics, Mexico City (2002)
12. Schone, P., Jurafsky, D.: Is knowledge-free induction of multiword unit dictionary headwords a solved problem? In: Proc. Conf. on Empirical Methods in Nat. Language Proc, pp. 100–108 (2001)
13. Seretan, V., Nerima, L., Wehrli, E.: Extraction of multi-word collocations using syntactic bigram composition. In: Proc. 4th Int'l Conf. on Recent Advances in NLP, Borovets, pp. 424–431 (2003)

14. Shimohata, S., Sugio, T., Nagata, J.: Retrieving collocations by co-occurrences and word order constraints. In: 35th Conf. of the ACL, Madrid, pp. 476–481 (1997)
15. Smadja, F.: Retrieving Collocations from Text. Xtract. Comp. Linguistics 19(1), 143–177 (1993)
16. Srikant, R., Agrawal, R.: Mining Sequential Patterns: Generalizations And Performance Improvements. In: Proc. 5th Int'l Conf. EDBT (1996)

Discovering Synonyms Based on Frequent Termsets*

Henryk Rybinski[1], Marzena Kryszkiewicz[1], Grzegorz Protaziuk[1], Adam Jakubowski[1], and Alexandre Delteil[2]

[1] ICS, Warsaw University of Technology
[2] France Telecome R & D
hrb,mkr,gprotazi@ii.pw.edu.pl,a.jakubowski@elka.pw.edu.pl,
alexandre.delteil@orange-ft.com

Abstract. Synonymy has been of high importance in information retrieval and automatic indexing. Recently, in the view of special needs for domain ontology building and maintenance, the problem returns with a higher demand. In the presented paper, we present a novel text mining approach to discovering synonyms or close meaning terms. The offered measures of closeness of terms (or their contexts) are expressed by means of data mining notions; namely, frequent termsets and association rules. The measures can be calculated by using data mining techniques, such as the well known Apriori algorithm. The approach is domain-independent and large-scale. It is, however, restricted to the recognition of parts of speech. In that sense the approach is language dependent, up to the language dependency of the parts of speech tagging process. The experimental results obtained with the approach are presented.

Keywords: association rules, frequent termsets, close meaning terms, synonyms.

1 Introduction

One can distinguish two main approaches in extracting semantic information from text corpora – knowledge-rich and knowledge-poor ones, according to the amount of knowledge they presuppose [6]. Knowledge-rich approaches require some sort of previously built semantic information, domain-dependent knowledge structures, semantic tagged training copora, or semantic dictionaries, thesauri, ontologies, etc. (see e.g. [9], [17]). This requirement is the main limitation in using these approaches. There is therefore a need for finding a knowledge-poor methodology that would give satisfactory results, especially for the cases of limited lexical resources.

In this context, the role of using text mining techniques for discovering semantic information from text corpora has been widely recognized (see e.g. [13], [2], [16], [8], [4]). One of the important fields of applications is to use the TM

* The work has been performed within the project granted by France Telecom.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 516–525, 2007.

methodology (text mining methodology) for building and maintaining ontologies ([13], [2], [16]).

In the paper, we continue research in this direction. We concentrate on verifying and proving usefulness of text mining approach for discovering synonyms. The work presented here is a part of an industrial project for France Telecom. The goal is to develop a text mining platform supporting domain specialist in building and maintaining ontologies. In particular, a dedicated platform has been built in order to verify novel TM algorithms in discovering synonyms, and homonyms, taxonomy relationship, compound term, and association relationships between terms.

For high importance of synonymy in information retrieval and automatic indexing, the problem of automatic discovering synonyms was a subject of research already in sixties of the last century ([3], [14], [10]). Recently, in the view of special needs for domain ontology building and maintenance, the problem returns with a higher demand.

In the presented paper, we present a novel text mining approach to discovering close meaning terms. The approach is domain-independent and large-scale. It uses basic notions of data mining techniques, such as finding frequent termsets, based on the well known *Apriori* algorithm [1]. The approach can be classified as knowledge-poor, though we use shallow text analysis. It is restricted to the recognition of parts of speech (POS). In that sense the approach is language dependent, up to the language dependency of the POS tagging process.

The rest of the paper is organized in the following manner. Section 2 briefly presents the main idea of the approach. In Section 3, the discovering procedure is described in details. Then, in Section 4, we discuss the results obtained with the approach. Section 5 concludes the paper.

2 Basic Idea and Concepts

The idea of discovering synonyms from text corpora presented in early papers (e.g. [14], [10]) was based on viewing language as a statistical phenomenon, and the aim was to define quantitatively measures of associations between words (alternatively, between the searching terms) using some function of the frequency with which words occur or co-occur within a document collection. To an extent, similar assumptions lie behind the text mining approach. As the results reported in [10] were satisfactory, we have decided to analyse how we can improve the approach using nowadays text mining methods.

In [10], the authors dealt with the derivation of a statistical measure of a relationship expressing the meanings "*equivalent*" or "*contrary*", thus enabling them to build pairs of the terms being synonyms, antonyms, but also related by means of the relationships "*broader-narrower*". The statistical measures of synonymy described in [10] are based on the following hypothesis:

Hypothesis 1: If two terms are synonymous, then they very infrequently, or never, co-occur in the same sentence, but they tend to have similar contexts in their separate occurrences.

This general and intuitive rule will be also applied in our approach. The main difference though is in defining the notion of *context* of two terms, and the measures of their similarities. Another essential difference, nota bene resulting from the notion of *context*, is in using the well known *Apriori* algorithm [1].

Now let us introduce basic definitions that will show how the approach relates to the standard data mining task. We see a text corpus as a set T of information units. An information unit is a set of terms. The more definitive notion of the information unit depends on the granularity level of the text mining process. In the implemented platform such a unit can be a set of terms extracted from a sentence or a set of terms extracted from a paragraph or else a set of terms extracted from a document. For the process of synonymy discovering, we accept sentence as the granularity level. Depending on the context, in the sequel the notion of a *sentence* will be used either as a set of terms, as extracted from a grammatical sentence or just as a grammatical sentence.

Let dictionary $D = \{t_1, t_2, \ldots, t_m\}$ be a set of distinct literals, called *terms*. In general, any set of terms is called a *termset*. A termset consisting of k terms will be called *k-termset*. So, the set T is a set of *sentences*, where each sentence s is a set of terms such that $s \subseteq D$. An *association rule* is an expression of the form:

$$X \rightarrow Y, \text{ where } \emptyset \neq X, Y \subset D \text{ and } X \cap Y = \emptyset.$$

Statistical significance of a termset X is called *support* and is denoted by $sup(X)$. $sup(X)$ is defined as the percentage of sentences in T that contain X. Statistical significance (*support*) of a rule $X \rightarrow Y$ is denoted by $sup(X \rightarrow Y)$ and is defined as follows:

$$sup(X \rightarrow Y) = sup(X \cup Y).$$

Additionally, an association rule is characterized by *confidence*, which expresses its strength. The confidence of an association rule $X \rightarrow Y$ is denoted by $conf(X \rightarrow Y)$ and is defined as follows:

$$conf(X \rightarrow Y) = sup(X \rightarrow Y)/sup(X).$$

We postulate that only those two terms (compound or single words) X and Y are likely to be synonyms if they are frequent in T, but do not co-occur frequently together; that is

$$sup(X) > minSup_1, sup(Y) > minSup_1 \text{ and } sup(XY) \leq minSup_2,$$

where $minSup_1$ is used to define meaningful terms, whereas $minSup_2$ is used to define *infrequent* pairs of terms as candidates for synonymy checking. The postulate above can be considered as a particular interpretation of Hypothesis 1.

Moreover, by Hypothesis 1 we expect that similar terms should occur in similar contexts. We introduce the definition of a *context* of the termset X, denoted by $context(X)$, by means of all frequent termsets being proper supersets of X as follows:

$$context(X) = \{Z \backslash X \mid X \subset Z \wedge sup(Z) > minSup_1\}.$$

In our approach, the necessary condition for similarity of two terms is nonemptiness of the intersection of their contexts. In the sequel, we denote the common context of X and Y by $CC(X,Y)$, i.e.:

$$CC(X,Y) = context(X) \cap context(Y).$$

We define *context similarity measure* for two termsets X and Y, denoted by $CSIM(X,Y)$ as follows:

If $context(X) \cap context(Y) = \emptyset$, then $CSIM(X,Y) = 0$, otherwise

$$CSIM(X,Y) = \frac{\mid CC(X,Y) \mid}{\mid context(X) \cup context(Y) \mid}.$$

$CSIM(X,Y)$ can be understood as a normalized measure of the intersection of the contexts of X and Y.

Let us look closer at the notions above by considering an example. Given terms *student, capital, rector, lecturer, professor* we find out frequent termsets and receive the following contexts:

Contex(student) = {{*warsaw*}, {*university*}, {*warsaw, university*}, {*exams*}, {*passed*}, {*passed, exams*}}
Contex(capital) = {{*warsaw*}, {*university*}}
Contex(rector) = {{*warsaw*}, {*university*}, {*warsaw, university*}}
Contex(lecturer)= {{*warsaw*},{*university*},{*warsaw, university*},{*exams*},{*performed*}, {*performed, exams*}}
Contex(professor)={{*warsaw*},{*university*},{*warsaw,university*},{*exams*},{*performed*}, {*performed, exams*}}

As one can see, the definition of the context by means of frequent termsets influences the similarity measure, as we take into account not only single terms, but also termsets (n-grams), which results in essential differentiating contexts of some pairs, see e.g. the pairs ({rector}, {capital}) or ({professor}, {student}). One can note that the context definition in [10] is based on frequent co occurrences of pairs, instead of frequent termsets.

The measure $CSIM$ expresses not only synonymy, because the similarity of two contexts may occur also for such relationships like *broader-narrower* or *category-instance.* In particular, if X and Y are related by the relationship *broader-narrower* or *category instance*, we will probably have $context(X) \subset context(Y)$, which still gives $CSIM = 1$.

If, on the other hand, a term (termset) X is a homonym, but one of its meanings is close to the meaning of the term (termset) Y, we may have $CC(X,Y) \neq \emptyset$, and $CSIM \ll 1$. Even more complicated may be a case when X and Y are both homonyms. For this case, we may still expect that if $CC(X,Y) \neq \emptyset$, it is that part of the contexts where the meanings of X and Y could be similar, whereas $context(X) \backslash CC(X,Y)$ and $context(Y) \backslash CC(X,Y)$ refer to other meanings. Therefore, we additionally check, if the confidences of the rules $X \rightarrow Z$ and $Y \rightarrow Z$, $Z \in CC(X,Y)$, are similar. If so, presumably X and Y are interchangeable within their common context $CC(X,Y)$. To this end, we introduce a measure which reflects association similarity of X and Y relative to the common context.

Association similarity measure: If there is no $Z \in CC(X, Y)$ such that $|Z| > 1$, then $ASIM(X, Y) = 0$, otherwise

$$ASIM(X,Y) = \frac{\sum\limits_{Z \in CC(X,Y)} (SConf(X \rightarrow Z, Y \rightarrow Z))}{\sum\limits_{Z \in CC(X,Y)} \mid Z \mid}$$

where

$$SConf(X \rightarrow Z, Y \rightarrow Z) = \min\left(\frac{conf(X \rightarrow Z)}{conf(Y \rightarrow Z)}, \frac{conf(Y \rightarrow Z)}{conf(X \rightarrow Z)}\right).$$

As one can notice, if there many rules $X \rightarrow Z$ and $Y \rightarrow Z$, $Z \in CC(X, Y)$, with similar confidences, the measure $ASIM(X, Y)$ tends to 1.

The usefulness of the introduced measures has been verified experimentally. All the experiments have been performed on the text mining platform TOM that has been implemented within the project. The platform uses a variety of open source software. In the next section, we describe the particular processing phases in more detail.

3 Discovering Procedure

Text preprocessing phase
The TOM platform provides options to define the granularity of the text mining process. In particular, TOM allows viewing the whole corpus as a set of documents, paragraphs or sentences. For the experiments of discovering synonyms, we set the granularity at the sentence level. Thus, the first step was to generate a set of sentences from all the documents in the repository. It means that the context of particular terms is restricted to the sentences.

Then we have used the Hepple tagger [7] for part-of-speech tagging of the words in the sentences. In TOM, the tagger is a wrapped code of the Gate part of speech processing resource [5]. As we seek for close terms within the same part of speech classes (verbs with verbs, nouns with nouns, etc.), we carry out the POS tagging step. Additionally, special filters can be used to filter out some parts of speech from all the sentences. For the synonymy discovering, we remove only stop words (adverbs, articles, prepositions).

Conversion into "transactional database"
The next step is to convert the text corpora into "transactional database" (in terms of [1]). This conversion makes it possible to use the classical *Apriori* based data mining algorithms. So, every unit of text (i.e. every sentence) is converted into a set of terms identifiers. This leads to speeding up all the data mining operations. Further on, the identifiers of all terms that do not have required minimum support are deleted from all the transactions.

Finding frequent termsets
Having transactional representation, we find frequent termsets with an *Apriori-like* algorithm which has been adopted for text mining. Once all frequent termsets are mined, we store them in the Lucene index [15] in order to provide means for fast viewing the contexts.

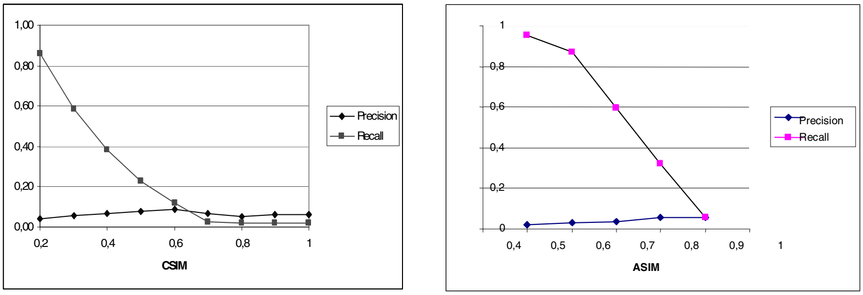
Identification of the close meaning pairs
Having the frequent termsets, we identify non-frequent pairs of terms belonging to the same POS class (our experiments have been performed for the pairs (*noun, noun*)). Then for all non-frequent pairs, we calculate common contexts and the similarity measures, as proposed in Section 2.

4 Experiments

We have performed a number of experiments in order to verify how the proposed method works. In particular, we wanted to check if it is possible to find synonyms by analyzing frequent termsets. We also wanted to check how different values of minimum support may influence the results.

It was expected that the method would yield a list of words that are very likely to be synonyms. Obviously, we were aware of the fact that comparing contexts of words may lead to the pairs of words that are not synonyms.

The experiments were performed on the repository that was built of a collection of about 120 scientific papers concerning text mining and ontology issues. This repository was quite difficult to process (the texts were in various forms, mainly PDF, sometimes Word). In addition, it is expected to be rather difficult for finding synonyms, as the language of the scientific papers is more restrictive and, as a rule, the authors try to be consistent and strict as much as possible in using the scientific terminology. The experiment was run for the pairs (*noun, noun*). The support threshold was set to $Sup_o = 0.08\%$. For this value, 6449 non-frequent pairs (*noun,noun*) were detected, which then were subject to consecutive steps (we have included *gerund* to the class *noun*).



(a) (b)

Fig. 1. Recall and Precision for (a) *CSIM* and (b) *ASIM*

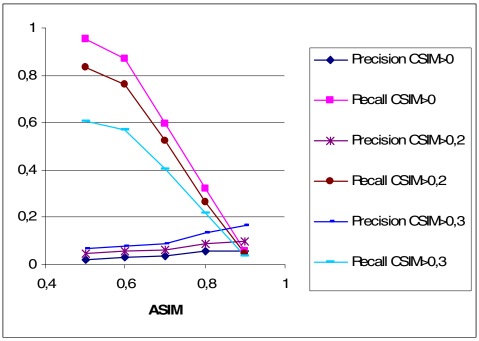


Fig. 2. Recall and precision for $ASIM$ with varying $CSIM$

Within all the pairs we found 109 pairs of terms having close meaning, including similarity between singular and plural forms. With the parameter $CSIM \geq 0.8$ we receive 31 pairs of which 2 are very close (like {*building, construction*}, {*methodology, study*}), whereas the remaining 29 pairs are irrelevant. If additionally we restrict the pairs to $ASIM \geq 0.7$, we receive the same close meaning pairs within 12 pairs. As many terms have various meanings, it turns out that even with $CSIM$ close to 0.2 some close meanings can be found. On Fig. 1, we can see how the recall and precision depend on the applied measures $CSIM$ and $ASIM$. As one can see, even for $CSIM \geq 0.2$ still interesting pairs may happen.

Table 1. $ASIM$ and $CSIM$ for found candidates for synonyms

Term 1	**Term 2**	$ASIM$	$CSIM$	**Term 1**	**Term 2**	$ASIM$	$CSIM$
algorithm	component	0,74	0,667	relationship	hierarchy	0,845	0,333
application	framework	0,87	0,455	relationship	link	0,834	0,667
application	task	0,828	0,5	representation	structure	0,766	0,3
approach	study	0,95	0,4	research	approach	0,872	0,333
building	construction	0,853	1	research	method	0,745	0,333
building	design	0,836	0,6	research	analysis	0,744	0,308
design	modeling	0,864	0,667	sources	texts	0,707	0,667
design	construction	0,792	0,6	structures	texts	0,905	0,5
mappings	methods	0,72	0,4	structures	languages	0,834	0,333
methodology	study	0,932	1	structures	algorithms	0,792	0,333
methodology	approach	0,886	0,4	table	list	0,709	0,4
methodology	methods	0,851	0,4	taxonomy	Wordnet	0,852	0,333
models	modeling	0,736	0,667	taxonomy	hierarchy	0,836	0,333
name	value	0,895	0,5	taxonomy	relation	0,797	0,4
Paper	documents	0,808	0,444	technique	tool	0,913	0,286
Paper	texts	0,714	0,444	term	word	0,711	0,3
Query	search	0,975	0,333	tools	methods	0,634	0,5
relation	similarity	0,861	0,333	type	types	0,795	0,417

In the case of using the $ASIM$ measure, most of the interesting pairs are covered by $ASIM \geq 0.6$. Still though, using only one parameter (either CSIM or ASIM) give rise to a large number of pairs, which have to be checked manually. It results from the fact that most of the terms, even those domain-specific, have various meanings, depending of the context. Examples of such terms are "access", "analysis", "application", "approach". We, therefore, have experimented with using the two measures for filtering the relevant pairs. In this experiment with the thresholds $CSIM \geq 0.2$ and $ASIM \geq 0.7$ we receive 508 pairs, 10% of which are interesting (and cover 50% of all the interesting pairs). Some of the pairs are not trivial, and it would be rather difficult to find them out manually (see Table 1).

5 Conclusions and Future Works

The experiments have proven that the *Apriori* approach combined with the $CSIM$ and $ASIM$ measures is very useful for finding similar meaning terms in text corpora. With the use of $CSIM$, some very strong synonyms can be detected, like (*building, construction*). However, as it has been also shown, by applying only the measure $CSIM$ many interesting relations are lost. With the use of $ASIM$, we were able to find also pairs of terms, which are synonyms in a restricted context but may have other various, more specific, meanings. Here, good examples are the "synonyms" (*taxonomy, hierarchy*) or (relationship, link), which could not be found with $CSIM$, whereas having $ASIM$=0.836, can be properly classified. By combining the two parameters we can filter out a reasonable set of irrelevant pairs, reaching 10% of precision, and 90% of completeness.

We have performed experiments with various minimum support parameters. They show that the lower the minimum support is, the better the results are. The results show that the discovered relationships reflect not only the synonymy, but also other close-meaning relationships. Namely, if any two terms do not co-occur, but have a very similar context, the following cases may hold:

1. the terms are very close synonyms (i.e. having close meaning in most of their contexts), e.g. (*building, construction*), or are much more dependent on a particular context, e.g. (*paper, text*);
2. the terms are related by the *broader-narrower* relation (previously observed in [10]), e.g. (*machine, tool*);
3. one of the term is a name of a category, whereas another one is an instance of that category, e.g. (*relationship, taxonomy*) or (*relationship, similarity*) (*language, rdf*);
4. two terms are instances of the same category e.g. (*rdf, oil*) or (*rdf, daml*);
5. the terms that are related by an ontology association relationship, e.g. (*description, schema*), (*implementation, task*), (*conceptualization, entity*).

The proposed approach finds instances of all the types of such close meaning terms. Unfortunately, the number of irrelevant pairs discovered as candidates for synonyms is still too high. This problem we are going to address in our future work. We anticipate that removing "noisy" termsets from the contexts should help considerably. We intend to examine the usefulness of applying concise representations of termsets [11, 12]. Additionally, we plan to elaborate an interactive effective method, as a post processing step, for selecting appropriate close meaning terms.

References

1. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: Proc. of the 20th Int'l Conf. on Very Large Databases, Santiago, 1994, pp. 487–499. Morgan Kaufmann, San Francisco (1994)
2. Ahonen-Myka, H.: Discovery of frequent word sequences in text. In: The ESF Exploratory Workshop on Pattern Detection and Discovery in Data Mining, Imperial College, London (2002)
3. Baxendal, P.B.: An empirical model for computer indexing. In: Machine Indexing, American U., Washington, D.C, pp. 207–218 (1962)
4. Delgado, M., Martin-Bautista, M.J., Sanchez, D., Amparo Vila Miranda, M.: Mining Text Data: Special Features and Patterns. In: Hand, D.J., Adams, N.M., Bolton, R.J. (eds.) Pattern Detection and Discovery. LNCS (LNAI), vol. 2447, pp. 140–153. Springer, Heidelberg (2002)
5. General Architecture for Text Engineering. http://gate.ac.uk/projects.html
6. Grefenstette, G.: Evaluation Techniques for Automatic Semantic Extraction: Comparing Syntatic and Window Based Approaches. In: Boguraev, B., Pustejovsky, J. (eds.) Corpus processing for Lexical Acquisition, pp. 205–216. MIT Press, Cambridge (1995)
7. Hepple, M.: Independence and commitment: Assumptions for rapid training and execution of rule-based POS taggers. In: Proc. of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000) (2000)
8. Hotho, A., Maedche, A., Staab, S., Zacharias, V.: On Knowledgeable Unsupervised Text Mining. In: Proc. of the DaimlerChrysler Workshop on Text Mining, Ulm (2002)
9. Hamon, T., Nazarenko, A., Gros, C.: A step towards the detection of semantic variants of terms in technical documents. In: Proc. of the 36th Ann. meeting of ACL (1998)
10. Lewis, P.A.W., Baxendale, P.B., Bennett, J.L.: Statistical Discrimination of the Synonymy/Antonymy Relationship Between Words. J. of the ACM 14(1), 20–44 (1967)
11. Kryszkiewicz, M.: Concise Representation of Frequent Patterns based on Disjunction-Free Generators. In: Proc. of the 2001 IEEE International Conference on Data Mining (ICDM), pp. 305–312. IEEE Computer Society, Los Alamitos (2001)
12. Kryszkiewicz, M., Gajek, M.: Concise Representation of Frequent Patterns based on Generalized Disjunction-Free Generators. In: Chen, M.-S., Yu, P.S., Liu, B. (eds.) PAKDD 2002. LNCS (LNAI), vol. 2336, pp. 159–171. Springer, Heidelberg (2002)
13. Maedche, A., Staab, S.: Mining Ontologies from Text. In: Dieng, R., Corby, O. (eds.) EKAW 2000. LNCS (LNAI), vol. 1937, pp. 189–202. Springer, Heidelberg (2000)

14. Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. Comm. ACM 8(10), 627–633 (1965)
15. Stevens, J.S., Husted, T., Cutting, D., Carlson, P.: Apache Lucene Overview (2006) http://lucene.apache.org/java/docs/index.pdf
16. Velardi, P., Fabriani, P., Missikoff, M.: Using text processing techniques to automatically enrich a domain ontology. In: Proc. of FOIS, pp. 270–284. ACM Press, New York (2001)
17. Wu, H., Zhou, M.: Optimizing Synonym Extraction Using Monolingual and Bilingual Resources. Ann. Meeting of the ACL. In: Proc. of the 2nd Int'l workshop on Paraphrasing, vol. 16, pp. 72–79 (2003)

A Summary Structure of Data Cube Preserving Semantics

Zhibin Shi and Houkuan Huang

School of Computer and IT, Beijing Jiaotong University, Beijing 100044, China
shizb@nuc.edu.cn hkhuang@center.bjtu.edu.cn

Abstract. The semantic relations among cells in data cube are more important for efficient query and OLAP. Normally the size of a data cube is very huge and relations among cells are very complicated so the semantic data cube is difficult to be realized. Based on quotient cube, Semantic Data Cube (*SDC*) structure is put forward in this paper. In *SDC* the lattice of cells is expressed as tree-hierarchy structure and each cell in lattice is replaced with its upper bound. The *SDC* depicts the lattice of cells concisely and preserves all the semantic relations among cells. Applying semantics to query answering and maintaining incrementally in *SDC*, the time of response and the cost of updating can be reduced greatly. Algorithms of constructing *SDC*, answering a query and maintaining incrementally in *SDC* are given. The experimental results show that the *SDC* is effective.

Keywords: data cube; semantic; incremental maintenance.

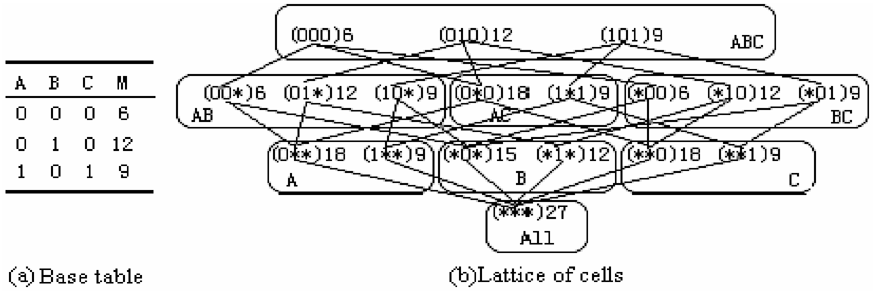
1 Introduction

The data cube[1] is an import operator for OLAP. Researchers have brought forward various methods to obtain data cubes which have different memory sizes and query answering time. The technology of computing data cubes is studied by [2,3,4,5], the technology of materializing and material selecting in data cube are worked by [6,7], literature [8] researches on compressed technique in data cube. In recent years, workers study how to discover and preserve the semantic relations among cells in data cube. [9,10,11] are representative literatures. They proposed some compact data structures which can preserve partial semantics in data cube.

Given the basic relation $R(A_1, A_2, \ldots, A_n, M)$, A_i means i^{th} attribute with n dimension attributes and M is the measure attribute. The data cube based R contains 2^n views through group-by operator. Each group-by is a view corresponding to one kind of granularity of data. Each view contains a set of cells. All views of data cube form a lattice which contains some partial orders and expresses basic semantic relations.

Two basic semantic relations of a data cube are the drill-down relation and the roll-up relation. Figure 1(b) is the lattice cell educed from the base table showed in Figure 1(a). The symbol '*' express the special value 'All' in [1], and the aggregation function is sum. It expresses the relations of drill-down and roll-up among concrete cells in data cube. For example, cell (0**) can be rolled up to (***), (0**) can be drilled down to (00*), (01*), (000), (010).

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 526–533, 2007.



A	B	C	M
0	0	0	6
0	1	0	12
1	0	1	9

Fig. 1. Lattice of cells educed from base table which has 3 tuples

In practice, the relations among cells are more important and more detailed for query and OLAP. Applying semantic relations to query, incremental maintenance and data analysis, the correlative operation can be simplified and directed.

Conventional techniques can only find and store the semantic relations among views but not among cells. In those approaches cells in same views are put into same files. The semantic relations among cells can't be preserved and roll-up and drill-down operators among cells can't be done. The naive idea to obtain fully semantics among cells is to store entirely the lattice of cells. But the lattice cell is very big normally and the relations among cells are very complicated. So the semantic data cube is difficult to be realized.

On the whole for semantic data cube there are some crucial technologies to be studied such as how to discover latent and pivotal semantics, how to select which kinds of semantics to be stored, how to store semantics and how to apply semantic relations for query, incremental maintenance and OLAP.

Dwarf[9] and Quotient cube[10,11] are data cubes that can keep semantics of cells.

Dwarf takes the technologies of prefix sharing and suffix coalescing. In the process of searching same prefixes and suffixes, the semantic relations among cells can be detected and then be preserved. Dwarf store data cube with complicated graph structure and semantic relations are not clear.

Quotient cube takes an elegant technique and partitions cells into several equivalent classes. Each cell in same class covers same tuple set in base table. So aggregations of these cells in one class are same. In Quotient cube, the lattice cell is expressed by relations of classes. But Quotient cube and *QC-Tree* structure have several problems. Firstly roll-up operator can not be done. So it is not fully semantic data cube. Secondly the semantic relations are not clear in quotient structure. In addition, *QC-Tree* is realized through complicated graph structure so it is difficult to be implemented.

According to above analysis, we proposed Semantic Data Cube (*SDC*) structure based on quotient cube. The *SDC* is a fully semantic cube and preserves all semantics among cells. And it can express semantic relations compactly and clearly.

2 *SDC*: Semantic Data Cube

Given the basic relation $R(A_1, A_2, \ldots, A_n, M)$. We let C be the data cube derived from R and c_i is one cell in C. Let $\prec$ be the partial order in cube lattice.

Definition 1.(cover and base table set) A cell c covers another cell t whenever there exists a roll-up path from t to c, i.e., $c \prec t$ in the cube lattice. The base table set of c called **BTS** concisely is the set of tuples in the base table covered by c.

For example, the BTS of cell (0**) is {(000), (010)}.

Definition 2.(upper bound of cell) $c_i \in C$, B is the base table set of c_i, if cell ub $(a_1, a_2, \ldots, a_n)$ satisfies the following conditions, we call ub is the **upper bound** of c_i.

1. If all the tuples in B have same value v in dimension A_i, then $a_i = v$;
2. otherwise, $a_i = *$.

At the same time we assume the **lower bound** of c_i is the cell itself.

For example, the BTS of (0**) is {(000), (010)}, and all the tuples in this set have the same value in A_1 and A_3. So the upper bound of (0**) is (0*0) and its lower bound is itself (0**).

Lemma 1. The aggregation of cell is equal to the aggregation of its upper bound.

From definition 2, we can see that the cell and its upper bound have a same BTS. So their aggregations are same.

Lemma 2. The upper bound of cell is unique and the lower bound of cell covers the upper bound.

From definition 2 we can obtain that.

2.1 The Structure of *SDC*

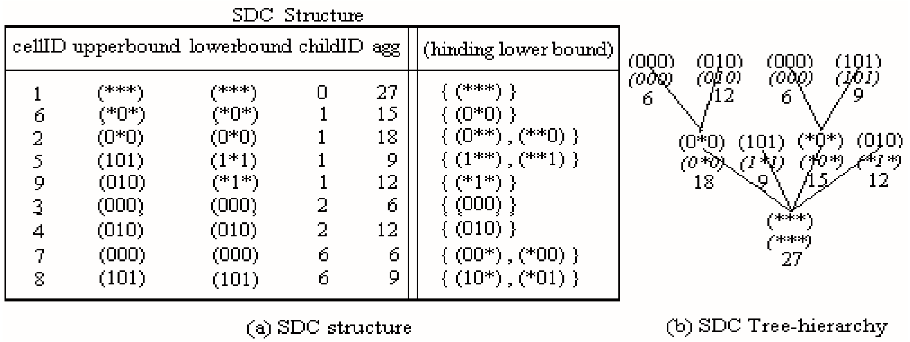
From above definitions and lemmas we can see that the cell and its upper bound have same BTS and have same aggregation. On second thoughts the upper bound of cell contains not only all the information of cell but also has more detailed information. In fact, the upper bound of cell is the most detail granularity in cover set of this cell. So we try to substitute cell's upper bound for cell itself in lattice cell. Thus some cells in lattice can be omitted and the relations are reduced accordingly. Then the lattice of cell can be simplified. But in this way the relations among cell may be jumbled and the partial order may be confused. So we record not only the upper bound but also the lower bound of cell. And we must record the partial order where exit order of $\prec$.

The *SDC* structure has the following properties:

1. For each cell in *SDC* there are 5 pieces of information should be recorded: the upper bound, the lower bound, aggregation of cell, *cellID* and *childID*. The *cellID* record the id of cell. The *childID* record the relations among cells. Suppose c and d are cells, if c directly drills down to d in *SDC*, the *childID* of d is the *cellID* of c. We called that c is child of d.
2. In *SDC* the lattice of cells is expressed as tree-hierarchy structure. There is only root whose upper and lower bound are fixed $(*, *, \ldots, *)$, called 0th level. There are at most n levels, where n is the number of dimensions.
3. Started from 1th level, every cell in original lattice is replaced with its upper bound. Thus some cells may be cut down as well as their upper cells.

4. If there are same upper bounds in same level, the only one cell is reserved to save space. For the sake of avoiding losing semantics, the lower bound of these cells must be recorded entirely and be stored into one lower bound.

Figure 2 shows the *SDC* structure deduced from base table in Figure 1(a). Each bound can be represented as a string w.r.t. a given dimension order. Figure 2(a) is actual storage in memory and cells in *SDC* are sorted according to *childID*. In *SDC* one lower bound may contain several cells. For example, (1**) and (**1) are all in 1th level and their upper bound is all (101). So we record only one cell (101) in *SDC*. At the same time we combine two lower bounds into one lower bound with (1*1).



SDC Structure

cellID	upperbound	lowerbound	childID	agg	(hinding lower bound)
1	(***)	(***)	0	27	{ (***) }
6	(*0*)	(*0*)	1	15	{ (0*0) }
2	(0*0)	(0*0)	1	18	{ (0**) , (**0) }
5	(101)	(1*1)	1	9	{ (1**) , (**1) }
9	(010)	(*1*)	1	12	{ (*1*) }
3	(000)	(000)	2	6	{ (000) }
4	(010)	(010)	2	12	{ (010) }
7	(000)	(000)	6	6	{ (00*) , (*00) }
8	(101)	(101)	6	9	{ (10*) , (*01) }

(a) SDC structure

(b) SDC Tree-hierarchy

Fig. 2. *SDC* Tree-hierarchy

Figure 2(b) shows tree-hierarchy corresponding to Figure 2(a). The branches show the order $\prec$ between cells. The roll-up and drill-down relations can be done through those branches. For example, from the cell whose *cellID* is 1, i.e. root (***), through searching such cells whose *childID* is 1, we can drill down from (***) to cells (*0*), (0**), (**0), (*1*), (1**) and (**1) which their lower bound are (*0*), (0*0), (*1*) and (1*1) respectively. On the other hand, the *childID* of cell (0**) is 1, i.e. its child is (***), so from (0**) we can roll up to (***).

2.2 Construction of *SDC*

The algorithm of constructing *SDC* is given below. It is similar to BUC[5] and takes searching strategy of Depth-First. The input is the base table *B* and cell (*,*,. . . ,*). *B* is partitioned on dimension and the value of ’*’ in input is assigned with concrete value. When we get a new cell, we calculate its aggregation and compute its upper bound. Then we let the upper bound act as the input into the next Depth-First process. Lastly we sort cells according to *childID*.

```
Algorithm: [Construct $SDC$]
Input: base table $B$;
Output: $SDC$;
1. $b = (*, \ldots, *)$;
```

```
2. call $DFS(b, B, 0, 0, 1)$;
3. Sort the cells according to childID and when childID is same, '*' precedes other
  values;
4. Return;
Function $DFS(c, Bc, k, chdID, clsID)$
1. if ($chdID == 0$)
    upper bound $d = (*, \ldots, *)$;//root's ub be $(*, \ldots, *)$;
  else compute the upper bound d of the cell c;
2. if there exits cell s in SDC s.t. $s.childID == chdID$ and $s.upperbound == d$
     combine lower bound of s with c's value in kth dim;
  Return;
3. Compute aggregation of cell c;
4. Record cell in SDC: lower bound c, upper bound d , aggregation of c, childID
  with chdID, cellID with clsID, then let chdID be clsID, clsID be clsID++;
5 for each $d[j] =' *'$ do
     for each value x in dimension j of base table
       let $d[j] = x$;
       if partition Bd is not empty, call $DFS(d.Bd, , j, chdID, clsID)$ ;
6. Return;
```

Theorem 1. *SDC* can contain all the semantic relations of lattice cell.

Theorem 1 shows that the *SDC* is full semantic data cube. From defines and lemmas above we can see that the upper and lower bound of cells contain all information of data cube. In addition we record all the partial orders between cells in *SDC*. Thus all the relations in original lattice of cells are expressed directly or indirectly in *SDC*. Then we can preserve the all relations among cells.

Compared with *QC-Tree* of quotient cube, *SDC* occupies more memory size and has some duplicate cells. But *SDC* cuts down more the size of memory compared with the full cubes proposed previously. Further more, *SDC* can express the lattice cell clearly and the operators of drill-down and roll-up can all be done between any cells.

3 Point Query

Query in data cube can be classified into point query, range query and iceberg query.

We propose efficient algorithm to answer point queries in *SDC*. The key idea for query answering efficiently in *SDC* comes from the fact that the *SDC* keep the semantics of cells, thereby query answering can be done according to the semantics.

A point query is that given a cell q, find its aggregate value(s). Query answering in *SDC* only visits the upper bound of cells.

Suppose that the upper bound of cell is $ub(x_1, \ldots, x_n)$, the cell to be queried is $q(y_1, \ldots, y_n)$, let $ub \wedge q = (z_1, \ldots, z_n)$, define as $z_i = x_i$, if $x_i = y_i$, otherwise $zi =' *', (1 \le i \le n)$.

We start at root c in 0^{th} level and carry out the operation $c.ub \wedge q$, following several situations may happen.

Case 1: $c.ub \wedge q = q$. It indicates that the aggregation of q has been found, and the aggregation of q is the aggregation of c, the query is end.

Case 2: $c.ub \wedge q = c.ub$. It indicates that cell c covers q and we need drill down to c's parent cell and continue to above operate in upper level.

Case 3: $c.ub \wedge q \neq c.ub \neq q$. It indicates that we don't find q and c does not cover q, we need search other c's sibing cells and continue to above operate.

After searching in same level if we can't fit case 1 and case 2, it indicate that the data cube does not contain q, then search is over and we return that q is not found.

In *SDC*, from cell in i^{th} level the most times of operator $c.ub \wedge q$ is $(n - i + 1)d$ (in there n is the dimension number, d is the cardinality of each dimension). In fact when we construct *SDC* we replace the cell with cell's upper bound, so the number of cells in i^{th} level must be smaller than $(n - i + 1)d$. So query in *SDC* is more effective.

4 Incremental Maintenance of *SDC*

When the base table changed, fast maintenance of *SDC* is important. In this section, we discuss the methods of insertion and deletion of single tuple.

4.1 Insertion

When a new tuple is inserted into base table, some upper bound of cells in original lattice may change. Then we must update *SDC* along with this change. we only seek and update such cells whose upper bound change.

Let T be the *SDC* and t be one cell in T and s be new tuple to be inserted. Let ub be the upper bound and lb be the lower bound of cell t. We compare s with cell's lower bound. Firstly we expand the lower bound of cell in *SDC* so that each lower bound denotes only one cell. When inserting new tuple the aggregation of root cell $(*, \ldots, *)$ must be changed and we add the value of s on it. Then the operation of insertion can be done from 1^{th} level. Several situations may appear as follows:

Case 1: $s \wedge t.lb = t.lb$. It indicates that t covers s, two more situations may appear:

a) $s \wedge t.ub = t.ub$. Firstly we add the aggregation of t with the value of s and then there are may be two more situations:

a.1) If the upper bound of t has two or more value of '*', then we drill down to t's child cell and recursion above operations;

a.2) If the upper bound of t has only one value of '*', we insert s and let its *childID* be *cellID* of t;

b) $s \wedge t.ub \neq t.ub$. We insert $s \wedge t.ub$ as the new upper bound of t, then we gather all the cells whose *childID* is t' *cellID* and s as set of data Bc and we process *SDC*'s *DFS* algorithm. Then we obtain several new cells and we insert them in *SDC*;

After finishing operation in one level if each dimension of s is dealt with, then the insertion operation is finished, otherwise we need insert s into this level, its upper bound is s itself and its low bound need be obtained through calculation.

4.2 Deletion

When a tuple is deleted from base table the upper bound of cells may change too.

Let s be the tuple to be deleted. Compare with s with t's upper bound from 0^{th} level. Many situations may happen:

Case 1: $s \wedge t.ub = t.ub \neq s$. It indicate that cell t covers s and we need update the aggregation of t, then we drill down to its parent cell in upper level;

Case 2: $s \wedge t.ub = s$. We delete s. If this level and the lower levels have only one cell, we must merge cell;

Case 3: $s \wedge t.ub \neq t.ub \neq s$. We continue to search next sibing cells in this level.

Compared with algorithms proposed before, in which one or more search must be done from end to end in incremental maintenance processing, *SDC* needn't scan all cells. When we insert or delete a tuple we compare it with the lower and upper bound in *SDC*. Only cells which cover cell are processed, otherwise we skip these cells as well as all those parent cells in upper level. From the processes of insertion and deletion we can discover that the cost of incremental maintenance in *SDC* is low.

5 Experimental Result and Analysis

In this section, we use synthetic and real-world data sets to evaluate *SDC*.

We compare *SDC* in memory size and constructing time with *BUC*[5] and *DFS* in *QC-Tree*[11] using synthetic data. Those three algorithms of constructing data cube all use sequential access to store data cube. *BUC* ends the recursive course when the number of tuples in data set is one. *DFS* only create temporary table in which record

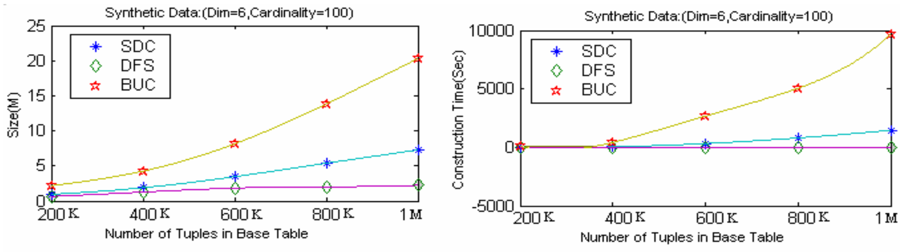


Fig. 3. Storage Size and Running Time on synthetic data

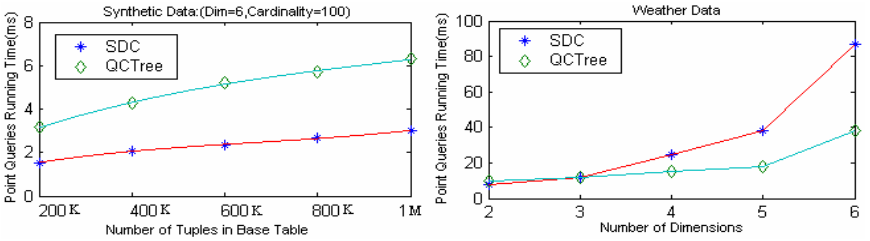


Fig. 4. Query performance

cell's lower and upper bound. Results are shown in Figure 3. *DFS* take the optimized method when recursion process (but only be able to preserve partial semantics). The speed of constructing *SDC* is slower than the *DFS* but is faster than the *BUC*. The memory size of *SDC* is larger than *DFS*'s but is smaller then *BUC*'s.

The synthetic and real-world data sets are used to examine the capability of query of *SDC*. The real dataset of weather conditions on land for September 1985[12] contains 1,015,367 tuples and 9 dimensions. We generate 5 datasets with 2 to 6 dimensions by projecting the weather dataset on the first k dimensions($1 \leq k \leq 9$). Results are shown in Figure 4. It shows that although the speed of query of *SDC* is slower than *QC-Tree* in big sets but is fairish closed to *QC-Tree*.

6 Conclusions

Based on quotient cube, Semantic Data Cube(*SDC*) structure is put forward in this paper. We discuss the methods of constructing, answering a query and incrementally maintaining in *SDC*. The contribution of this paper is that *SDC* can express the lattice cell concisely and can preserve all semantic relations of lattice cell. We can apply semantic relations for query answering and incrementally maintaining. The time of query answering and the cost of updating are low. The experiments show that although memory size of *SDC* may be large but *SDC* can query and update efficiently.

References

1. Gray, J., Bosworth, A., Layman, A., Pirahesh, H.: Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. In: ICDE, pp. 152–159 (1996)
2. Agarwal, S., Agrawal, R., Deshpande, P.M., Gupta, A., Naughton, J.F., Ramarkrishman, R., Sarawagi, S.: On the computation of multidimensional aggregates. In: ICDE, pp. 506–521 (1996)
3. Ross, K.A., Srivastava, D.: Fast computation of sparse data cubes. In: ICDE, pp. 116–125 (1997)
4. Zhao, Y., Deshpande, P.M., Naughton, J.F.: An array-based algorithm for simultaneous multidimensional. In: SIGMOD, pp. 159–170 (1997)
5. Beyer, K., Ramakrishnan, R.: Bottom-up computation of sparse and iceberg cube. In: SIGMOD, pp. 359–370 (1999)
6. Harinarayan, V., Rajaraman, A., Ullman, J.D.: Implementing data cubes efficiently. In: SIGMOD, pp. 205–216 (1996)
7. Shukla, A., Deshpande, P.M., Naughton, J.F.: Materialized view selection for multidimensional datasets. In: VLDB, pp. 488–499 (1998)
8. Wang, W., Lu, H.J., Feng, J.L., Yu, J.X.: Condensed cube: An effective approach to reducing data cube size. In: ICDE, pp. 155–165 (2002)
9. Sismanis, Y., Deligiannakis, A., Roussopoulos, N., Kotidis, Y.D.: Shrinking the PetaCube. In: ACM SIGMOD, pp. 464–475 (2002)
10. Lakshmanan, L.V.S., Pei, J., Han, J.W.: Quotient cube: How to summarize the simantics of a data cube. In: VLDB, pp. 778–789 (2002)
11. Lakshmanan, L.V.S., Pei, J., Zhao, Y.: QC-Trees: An efficient summary structure for semantic OLAP. In: ACM SIGMOD, pp. 64–75 (2003)
12. Hahn, C., et al.: Edited synoptic cloud reports from ships and land stations over the globe, 1982-1991 (1994). cdiac.est.ornl.gov/ftp/ndp026b/SEP85L.Z

Mining Association Rules with Respect to Support and Anti-support-Experimental Results

R. Słowiński[1,2], I. Szczęch[1], M. Urbanowicz[1], and S. Greco[3]

[1] Inst. of Computing Science, Poznań University of Technology,60-965 Poznań, Poland
{Roman.Slowinski, Izabela.Szczech}@cs.put.poznan.pl
[2] Inst. for Systems Research, Polish Academy of Sciences, 01–447 Warsaw, Poland
[3] Faculty of Economics, University of Catania, Corso Italia, 55, 95129 Catania, Italy
salgreco@mbox.unicit.it

Abstract. Evaluating the interestingness of rules or trees is a challenging problem of knowledge discovery and data mining. In recent studies, the use of two interestingness measures at the same time was prevailing. Mining of Pareto-optimal borders according to support and confidence, or support and anti-support are examples of that approach. Here, we consider induction of "*if..., then...*" association rules with a fixed conclusion. We investigate ways to limit the set of rules non–dominated wrt support and confidence or support and anti-support, to a subset of truly interesting rules. Analytically, and through experiments, we show that both of the considered sets can be easily reduced by using the valuable semantics of confirmation measures.

Keywords: Association rules, Induction, Support, Anti–support, Confirmation, Confidence, Pareto–optimal border.

1 Introduction

In data mining and knowledge discovery, the discovered knowledge patterns are often expressed in a form of "*if..., then...*" rules. They are consequence relations representing correlation, association, causation etc. between independent and dependent attributes. In order to increase the relevance and utility of selected rules and, thus, also limit the size of the resulting rule set, quantitative measures, also known as interestingness measures, have been proposed and studied (e.g. confidence, support, gain, conviction, lift). Among widely studied interestingness measures, there is, moreover, a group of Bayesian confirmation measures, which quantify the degree to which a piece of evidence built of the independent attributes provides "evidence for or against" the hypothesis built of the dependent attributes [4]. Another approach to evaluation of generated rules concentrates on the use of two different interestingness measures. In this paper, we show a way to limit the set of rules generated with respect to pairs of measures: support–confidence and support–anti–support, by filtering out the rules for which the premise does not confirm the conclusion. This proposition is based on imposing the confirmation perspective on the analyzed two–dimensional evaluations.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 534–542, 2007.

The paper is organized as follows. In section 2, there are preliminaries on rules and their quantitative description. In section 3, we investigate the idea and advantages of mining only rules with positive confirmation from Pareto–optimal border with respect to support and confidence. Section 4 concentrates on the proposal of limiting the set of rules generated with respect to support and anti–support. Theoretical considerations are supported by experimental results. The paper ends with conclusions.

2 Preliminaries

Since discovering rules from data is the domain of inductive reasoning, its starting point is a sample of larger reality often given in a form of a data table. Formally, a *data table* is a pair $S = (U, A)$, where U is a nonempty finite set of objects called *universe*, and A is a nonempty finite set of *attributes* such that $a : U \rightarrow V_a$ for every $a \in A$. The set V_a is a domain of a. A *rule* induced from S is denoted by $\phi \rightarrow \psi$ (read as "*if* ϕ*, then* ψ"). It consists of antecedent ϕ and consequent ψ, called *premise* and *conclusion*, respectively. In this paper, similarly to [2], we consider evaluation of rules with the same conclusion.

2.1 Partial Preorder on Rules in Terms of Two Measures

Let us denote by $\preceq_{qt}$ a partial preorder given by a dominance relation on a set X of rules in terms of any two different interestingness measures q and t, i.e. for all $r_1, r_2 \in X r_1 \preceq_{qt} r_2$ if $r_1 \preceq_q r_2$ and $r_1 \preceq_t r_2$. Recall that a *partial preorder* on a set X is a binary relation R on X that is reflexive and transitive. The partial preorder $\preceq_{qt}$ can be decomposed into its asymmetric part $\prec_{qt}$ and its symmetric part $\sim_{gt}$ in the following manner: given a set of rules X and two rules $r_1, r_2 \in X, r_1 \prec_{qt} r_2$ if and only if $q(r_1) \leq q(r_2) \wedge t(r_1) < t(r_2)$, or $q(r_1) \leq q(r_2) \wedge t(r_1) < t(r_2)$, moreover, $r_1 \sim_{qt} r_2$ if and only if $q(r_1) = q(r_2) \wedge t(r_1) = t(r_2)$. If for a rule $r \in X$ there does not exist any rule $r' \in X$, such that $r \prec_{qt} r'$ then r is said to be non–dominated (i.e. Pareto–optimal) wrt interestingness measures q and t. A set of all non–dominated rules wrt q and t is also referred to as an *q–t Pareto–optimal border*.

2.2 Monotonicity of a Function in Its Argument

Let x be an element of a set of rules X and let $g(x)$ be a real function associated with this set, such that $g : X \rightarrow \mathbf{R}$. Assuming an ordering relation $\succ$ in X, function g is said to be monotone (resp. anti–monotone) in x, if for any $x, y \in X$, relation $x \succ y$ implies that $g(x) \geq g(y)$ (resp. $g(x) \leq g(y)$).

2.3 Support, Confidence and Anti–support Measures of Rules

Among measures very commonly associated with rules induced from information table S, there are *support* and *confidence*. The *support* of condition ϕ, denoted as $sup(\phi)$, is equal to the number of objects in U having property ϕ. The support of rule $\phi \rightarrow \psi$, denoted as $sup(\phi \rightarrow \psi)$, is the number of objects in U having property ϕ and ψ.

The *confidence* of a rule (also called *certainty*), denoted as $conf(\phi \rightarrow \psi)$, is defined as: $conf(\phi \rightarrow \psi) = \frac{sup(\phi \rightarrow \psi)}{sup(\phi)}$, $sup(\phi) > 0$.

Anti–support of a rule, denoted as $anti - sup(\phi \rightarrow \psi)$, is equal to the number of objects in U having the property ϕ but not having the property ψ. Thus, anti–support is the number of counter–examples, i.e. objects for which the premise ϕ evaluates to true but which fall into a class different than ψ. Note that anti–support can also be regarded as $sup(\phi \rightarrow \neg\psi)$.

2.4 Bayesian Confirmation Measures

Bayesian confirmation measures constitute a group of interestingness measures that quantify the degree to which a premise ϕ provides “support for or against” a conclusion ψ [4]. Under the “closed world assumption” adopted in inductive reasoning, and because U is a finite set, a confirmation measure denoted by $c(\phi \rightarrow \psi)$ is required to satisfy the following definition:

$$c(\phi \rightarrow \psi) = \begin{cases} > 0 \text{ if } conf(\psi \rightarrow \phi) > sup(\psi)/|U|, \\ = 0 \text{ if } conf(\psi \rightarrow \phi) = sup(\psi)/|U|, \\ < 0 \text{ if } conf(\psi \rightarrow \phi) < sup(\psi)/|U|. \end{cases} \qquad (1)$$

For the confirmation measures a desired property of monotonicity (M) was proposed in [5]. This monotonicity property says that, given an information system S, a confirmation measure is a function non–decreasing wrt $sup(\phi \rightarrow \psi)$ and $sup(\neg\phi \rightarrow \neg\psi)$, and non–increasing wrt $sup(\neg\phi \rightarrow \psi)$ and $sup(\phi \rightarrow \neg\psi)$. Among confirmation measures that have property (M) there is e.g. confirmation measure f [4] defined as:
$f(\phi \rightarrow \psi) = \frac{conf(\psi \rightarrow \phi) - conf(\neg\psi \rightarrow \phi)}{conf(\psi \rightarrow \phi) + conf(\neg\psi \rightarrow \phi)}$.

2.5 A Brief Description of a Dataset and Experiments

For the purpose of these experiments we used a dataset *adult* [7]. The number of analyzed instances reached 32 561. They were described by 9 nominal attributes differing in domain sizes. Missing values were substituted by the most frequently appearing one. Two experiments were conducted: one generating rules wrt support and confidence, and the second one generating rules according to support and anti-support. Both of them proceeded in a two step Apriori-like framework:

- firstly, all conjunctions of elementary conditions (i.e. itemsets) that exceeded the minimum rule support threshold (i.e. frequent itemsets) were found;
- secondly, those frequent itemsets were used to generate association rules having either confidence or anti-support not smaller than the user’s defined threshold.

The detailed description as well as the efficiency comparison of the applied algorithms (based on [1,6]) can be found in [9]. Throughout the experiment, the value of support was expressed as a relative value between 0 and 1. During the frequent itemset generation phase, only itemsets that exceeded 0.15 support threshold were approved. No confidence nor anti-support thresholds were applied in order to show the complete Pareto-optimal border exceeding the support threshold.

3 Support–Confidence Pareto–Optimal Border

Bayardo and Agrawal [2] proposed evaluation of the set of rules in terms of two popular interestingness measures being rule support and confidence. They have proved that for a class of rules with fixed conclusion, the support–confidence Pareto–optimal border includes optimal rules according to several different interestingness measures, such as gain, lift, conviction, etc. Thus, by solving an optimized rule mining problem wrt rule support and confidence one can identify a set of rules containing most interesting (optimal) rules according to several interestingness measures. However, despite those valuable features of the support–confidence Pareto–optimal border, one cannot, in general, claim that the set of dominated rules is without interest. It can be e.g. due to the fact that in order to cover the analyzed concept (decision class) one has to use both dominated and non–dominated rules. Of course, a user can set some thresholds both on rule support and confidence, but still taking under the consideration both dominated and non–dominated rules can result in a large, difficult to analyze set of rules. Hence, we propose a way to limit the set of the analyzed rules by using the valuable semantic of confirmation measures.

3.1 The Confirmation Perspective on the Support–Confidence Evaluations

The advantages of semantic utility of confirmation measures in general over confidence have been widely studied in [3,5]. Thus, we find it valuable to impose the confirmation perspective on the analyzed support–confidence evaluations and limit in this way the set of rules to be analyzed. It has been analytically proved in [3] that for a fixed value of rule support, confidence is monotone wrt any confirmation measure having the desired property of monotonicity (M) proposed in [5].

Let us observe that according to definition (1) of $c(\phi \to \psi)$, we have:

$$c(\phi \to \psi) > 0 \Leftrightarrow conf(\phi \to \psi) > \tfrac{sup(\psi)}{|U|} \tag{2}$$

Since, we limit our consideration to rules with the same conclusion, then $|U|$ and $sup(\psi)$ should be regarded as constant values. Thus, (2) shows that rules laying under a constant, expressing what percentage of the whole dataset is taken by the considered class ψ, are characterized by negative values of confirmation

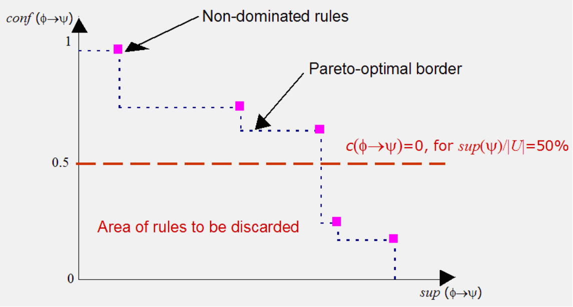


Fig. 1. An example of a constant line representing $c(\phi \rightarrow \psi) = 0$ in a support–confidence space. Rules laying under it should be discarded from further analysis.

(see Fig. 1). For those rules ψ is satisfied less frequently when ϕ is satisfied rather than generically.

It is also interesting to investigate a more general condition $c(\phi \rightarrow \psi) \geq k$, $k \geq 0$, for some specific confirmation measures. In the following, we consider confirmation measure $f(\phi \rightarrow \psi)$.

Theorem 1. (See proof in [8])

$$f(\phi \rightarrow \psi) \geq k \Leftrightarrow conf(\phi \rightarrow \psi) \geq \frac{sup(\psi)(k+1)}{|U|-k(|U|-2sup(\psi))} \tag{3}$$

3.2 Experiments with Rule Induction with Respect to Support and Confidence

On Fig. 2 we show association rules generated, according to mentioned thresholds for the conclusion: workclass='Private'. This class contains information about people working in a private sector. Rules are presented in a support–confidence space.

This experiment makes it evident that in practice even rules with high value of confidence (exceeding even 0.7) can be found useless as their premise disconfirms the conclusion (those rules are marked by solid circles). It is therefore clear, that the semantic scale of the confidence measure is not enough and that confirmation measures are very much needed. Sometimes even rules from the Pareto–optimal border need to be discarded from further analysis as their value of confirmation is non–positive. On Fig. 2 a constant line was placed separating the rules with positive confirmation (situated above the line) from those with non–positive confirmation (situated below the line). Fig. 2 visualizes result (2) and says how big (in comparison to the whole dataset) is the considered class of rules for the analyzed conclusion workclass='Private'. Illustrations for other classes can be found in [8,9]. By imposing the confirmation perspective, the number of rules to be analyzed by the domain expert can be significantly reduced. For the conclusion being worklass='Private', 41 out of 84 rules had to be discarded for disconfirming the conclusion. Tab. 1 shows results for other conclusions that we have considered.

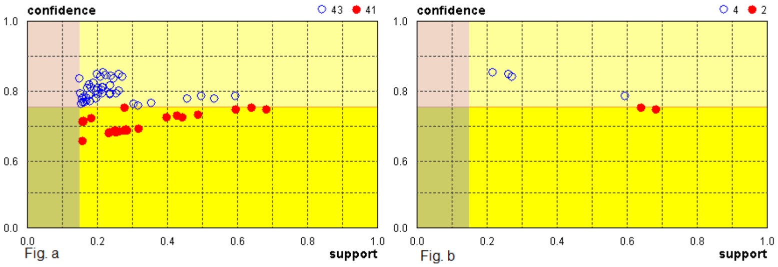


Fig. 2. Rules generated for a conclusion workclass='Private' with positive (empty circles) and non–positive confirmation measure value (solid circles) in a support–confidence space. Fig. a – all generated rules, Fig. b – the Pareto-optimal border only.

Table 1. Information about the percentage of rules with non-positive confirmation in the set of all generated rules for different conclusions

Considered conclusion	**No. of all rules**	**No. of all rules with non–positive confirm.**	**Reduction percentage**
workclass='Private'	84	41	49%
sex=Male	85	24	28%
income<=50kUSD	87	43	49%

Table 2. Information about the percentage of rules with non-positive confirmation laying on the support–confidence Pareto–optimal border for different conclusions

Considered conclusion	**No. of all rules on Pareto border**	**No. of all rules with non–positive confirm.**	**Reduction percentage**
workclass='Private'	6	2	33%
sex=Male	6	1	17%
income<=50kUSD	5	1	20%

Tab. 2 shows how many rules with non–positive confirmation laid on the support–confidence Pareto–optimal border for different considered conclusions. Even Pareto–optimal borders, i.e. objectively the best sets of rules, contain rules that are misleading. In some cases, the support–confidence Pareto–optimal border could be reduced by even 33%, like for the conclusion workclass='Private'.

4 Support–Anti–support Pareto–Optimal Border

Presentation of association rules in dimensions of rule support and anti–support was proposed in [3]. The idea of combining those two dimensions came from a critical remark towards support–confidence Pareto–optimal border. In [3], it was proved that a rule maximizing a confirmation measure satisfying the property (M) is on the support–confidence Pareto–optimal border only if a specific condition is satisfied. Thus, in general, not all rules maximizing such a measure are on the support–confidence Pareto–optimal border. However, due to valuable semantics of confirmation measures, mining all rules that maximize confirmation measures with (M), became an interesting problem. The solution is support–anti–support Pareto–optimal border. It was proved in [3] that the best rule according to any of confirmation measures with (M) must reside on the support–anti–support Pareto–optimal border. Moreover, it was pointed out in [3] that the Pareto–optimal border of support–anti–support contains the support–confidence Pareto–optimal border. Despite all good characteristics of the support–anti–support Pareto–optimal border, one can still remain interested in the set of dominated rules. Thus, analyzing whether one can limit the set of rules, by imposing a confirmation perspective on the spport–anti–support evaluations, is interesting.

4.1 The Confirmation Perspective on the Support–Anti–support Evaluations

It has been analytically proved in [3] that for a fixed value of rule support, any confirmation measure $c(\phi \to \psi)$ having the desired property of monotonicity (M) is anti–monotone (i.e. non–decreasing) wrt anti–support. Let us observe that a simple transformation of definition (1) leads to the following result:

$$c(\phi \to \psi) \geq 0 \Leftrightarrow anti - sup(\phi \to \psi) \leq sup(\phi \to \psi) \left[\frac{|U|}{sup(\psi)} - 1 \right] \qquad (4)$$

Having limited our consideration to rules with the same conclusion, $|U|$and $sup(\psi)$ should be regarded as constant values. Thus, the result (4) shows that a simple linear function bounds rules that are characterized by positive values of confirmation from those with non–positive confirmation values (see Fig. 3).

It is also interesting to investigate a more general condition $c(\phi \to \psi) \geq k, k \geq 0$. Let us consider again $f(\phi \to \psi)$.

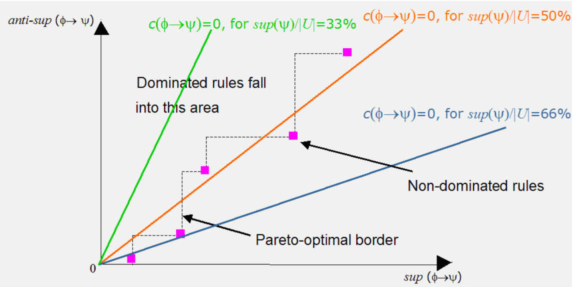


Fig. 3. Three examples of linear functions representing $c(\phi \to \psi) = 0$ in a support–anti–support space. Lines were drawn according to a set of rules for conclusions different in cardinality. Rules laying above them should be discarded from further analysis.

Theorem 2. (See proof in [8])

$$f(\phi \to \psi) \geq k \Leftrightarrow anti - sup(\phi \to \psi) \leq sup(\phi \to \psi)(U - sup(\psi))\frac{1-k}{(1+k)sup(\psi)} \quad (5)$$

4.2 Experiments with Rule Induction with Respect to Support and Anti–support

On Fig. 4, we show association rules generated, according to mentioned threshold, for the conclusion: workclass='Private'.

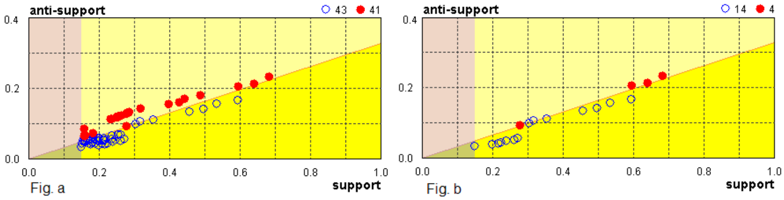


Fig. 4. Rules generated for a conclusion workclass='Private' with positive (empty circles) and non–positive (solid circles) confirmation measure value in a support–anti–support space. Fig. a – all generated rules, Fig. b – the Pareto-optimal border only.

Table 3. Information about the percentage of rules with non-positive confirmation laying on the support–anti–support Pareto–optimal border for different conclusions

Considered conclusion	**No. of all rules on Pareto border**	**No. of all rules with non–positive confirm.**	**Reduction percentage**
workclass='Private'	18	4	22%
sex=Male	8	3	38%
income<=50kUSD	15	4	27%

This experiment makes it clear, that despite the valuable properties of support–anti–support Pareto–optimal border, it is necessary to take under consideration also the information brought by the sign of the confirmation measures. Within the Pareto-optimal set presented on Fig. 4, 22% of rules need to be discarded as their value of confirmation is non–positive. On Fig. 4, a linear function was placed separating the rules with positive confirmation (situated under the line) from those with non–positive confirmation. Fig. 4 visualizes result (4). Tab. 3 presents the percentage of rules to be discarded from the support-anti-support Pareto-optimal border. In the conducted experiment the set of rules to be analyzed could be reduced by e.g. about 22% (workclass='Private').

5 Conclusions

In this paper, we investigated rules induced for a fixed conclusion and evaluated in spaces of support–confidence and support–anti–support. The Pareto–optimal borders of those spaces have some valuable features. However, these worthy features, do not assure that the number of induced rules would not exceed the human user capabilities to analyze them. Inspired by the strength of the semantics of confirmation measures, we show that it is reasonable to limit the set of rules by eliminating those that are characterized by non–positive or small values of confirmation. We have shown analytically that a simple constant line imposed on the support–confidence space bounds the rules with positive values of confirmation measure from those with non–positive confirmation values. This is a very practical result allowing to limit the set of analyzed rules only to those with positive confirmation values, without actually calculating the value of a particular confirmation measure for each of the induced rules. Analogous analysis has been conducted for rules in support–anti–support space. We have shown that a simple linear function separates the rules with positive and non–positive values of confirmation. Again, this is an easy approach to limit the set of analyzed rules. Experimental results show how big the reduction can be.

References

1. Agrawal, R., Imielinski, T.: Mining associations between sets of items in massive databases. In: Proc.of ACM-SIGMOD Int'l Conf. on Management of Data (1993)
2. Bayardo, R.J., Agrawal, R.: Mining the most interesting rules. In: Proc. of 5th ACM-SIGKDD Int'l Conf. on Knowledge Disc.and Data Mining, pp. 145–154 (1999)
3. Brzezińska, I., Greco, S., Słowiński, R.: Mining pareto-optimal rules with respect to support and anti-support. Eng. Applic.of Artif. Intelligence (to appear)
4. Fitelson, B.: Studies in Bayesian Confirmation Theory. MSc thesis, University of Wisconsin, Madison (2001)
5. Greco, S., Pawlak, Z., Słowiński, R.: Can bayesian confirmation measures be useful for rough set decision rules? Eng. Applic.of Artif. Intelligence 17, 345–361 (2004)
6. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proc. ACM SIGMOD Conference on Management of Data, pp. 1–12 (2000)
7. Kohavi, R.J.: Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In: Proc. of the 2nd Int'l Conf. on Knowledge Discovery and Data Mining (1996)
8. Słowiński, R., Szczech, I., Urbanowicz, M., Greco, S.: Experiments with induction of assoc. rules wrt support,anti-support. Technical Report RA-018/06, II-PP (2006)
9. Urbanowicz, M.: Induction of association rules with given support, confidence and confirmation. PhD thesis, Poznań University of Technology (2007)

Developing Data Warehouse for Simulation Experiments

Janusz Sosnowski, Przemysław Zygulski, and Piotr Gawkowski

Institute of Computer Science, Warsaw University of Technology,
ul. Nowowiejska 15/19, Warsaw 00-665, Poland
jss@ii.pw.edu.pl

Abstract. The paper deals with the problem of creating a specialized data warehouse for collecting and analyzing experimental results, which relate to system dependability evaluation using fault injections into running programs. The developed data warehouse with embedded data mining capabilities facilitates to identify factors influencing fault susceptibility of analyzed applications. The paper presents the developed system, and illustrates its usefulness with a sample of experimental results.

1 Introduction

System robustness (correct operation in the presence of faults) and fail-safe operation (avoiding wrong results or output signals) are becoming common requirements to modern civilisation. Hence, an important issue is the evaluation of system dependability and the analysis of the system behaviour in the presence of faults. For this purpose various fault injection techniques have been proposed. In the literature two approaches to fault injection are presented: applied to the existing systems or to their models. The first approach bases on pin-level fault injection, heavy-ion irradiation, electrical disturbances, laser fault injection and software implemented fault injection [1,2,8,9]. Fault injection applied to the system models can be targeted at specific levels of the system or the circuit e.g. electrical or high-level description (RTL, VHDL models [1,2]). This approach is especially useful to characterize fault propagation from physical to higher levels. Software implemented fault injection (SWIFI) assures checking fault susceptibility of the applications in the real system environment e.g. [2,5,9].

We have developed several flexible SWIFI tools [8,9], which were used in many experiments to analyse fault susceptibility of program applications as well as to verify the effectiveness of various fault detection and tolerance mechanisms. We have observed a large dispersion of experimental results. The main goal of the presented study was identification of factors influencing test results. For this purpose we have developed a specialised data warehouse with data exploration capabilities. We are pioneers in using this approach for dependability analysis. Basing on our experience with fault injection techniques (section 2) we selected appropriate attributes (types and value ranges), and defined an original data model for the created data warehouse (section 3). The usefulness of this approach

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 543–552, 2007.

has been illustrated (section 4) with data exploration results, which show various aspects related to fault susceptibility of the analysed applications.

2 Main Features of Fault Injectors

The software implemented fault injector (SWIFI) simulates faults in the system environment by disturbing the states of processor registers or RAM locations (storing program code and data). The fault type (bit flip, bit setting, resetting and bridging), the fault duration, triggering moments and location can be specified explicitly or generated pseudorandomly according to the experiment configuration.

SWIFI compares the results of the analysed application disturbed by a fault with those found during the golden (referential) run. It registers the exit code, results and all generated events and exceptions in a test. In general, we distinguish 5 classes of test results: C - correct result, INC - incorrect result, S - fault detected by the system, T - time-out, U user messages (generated by the program if an error is detected). Detailed qualification of test results is also possible. System exceptions (S) are mostly generated by special hardware mechanisms embedded in contemporary COTS (commercial off-the-shelve) systems. Microprocessors signal such exceptions as: access violation (within RAM), in page error, array bounds exceeded, data type misalignment (wrong word boundaries), illegal instruction, etc. [9]. SWIFI delivers also various statistics on distribution of fault injections, system resource activity etc.

In fault injection experiments, it is important to specify test scenarios, which cover fault distribution in time and space (fault localization and triggering), fault types, input data profiles etc. The performed experiments were targeted at transient faults (bit flips) injected into registers (specified CPU or FPU registers, or all of them), the code or data area of the memory. By concentrating on specified system resources (or code segments) we can perform deeper analysis and tune appropriate fault handling mechanisms. For each application, we choose a representative set of input data to assure high coverage of the code, decisions etc. This selection is based on the analysis of some coverage measures [9]. The number of injected faults is sufficiently large (typically $10^4 \div 10^6$ faults) to assure statistical significance of the obtained results.

We have improved application robustness using various fault-handling mechanisms targeted at fault detection and fault correction or masking. Fault detection techniques are based on duplication of variables, duplication of calculations, checking various assertions and checking the program control flow. Techniques tolerating faults are based on software redundancy, error detection and fault recovery. They range from duplication to triplication of the code and data (including various diversities). We also handle exceptions generated by the system. We use software-implemented schemes proposed in the literature and our own improved approaches [5,9]. Results of fault injection experiments related to standard (basic) applications as well as applications with software implemented fault detection (FD) and fault tolerance (FT) are given in tab. 1. Here we can observe

Table 1. Fault injection result distribution (%)

	Faults injected into data, code and registers			Faults injected into CPU registers (basic applications)					
	Basic	FD	FT	eax	ebx	esp	ebp	esi	eip
C	2-76	14-63	30-100	5-50	0-84	0-9	0-72	0-75	1-6
S	0-66	0-57	0-64	0-52	0-54	57-100	4-97	2-61	4-94
T	0-12	0-0.5	0-7	0-3	0-1	0-1	0-1	0-1	0-0.3
INC	4-97	1-40	0-36	36-95	14-65	0-34	3-94	22-68	5-90

various levels of fault leakage (INC results) and correct results (C). In particular this depends upon the applied dependability improvement technique. Some detailed results are given in [5,6].

Each column of tab. 1 gives ranges of test results over many applications (pseudorandom distribution of faults). The big dispersion of results is caused by application specificities, different fault susceptibility of code, data area and registers, different activity of used computer resources, etc. Fault hardened applications (FD and FT) showed also in some cases excessive percentage of incorrect results (INC) due to some inefficiency of used mechanisms. Hence a systematic analysis of factors influencing application susceptibility to faults is needed.

Fault injection experiments are performed for a specified application and input data set. Each experiment comprises many tests, a single test relates to one injected fault. Results of the experiment tests are stored in a file. Test result qualification (C, INC, T, S) is performed by comparison with the result of the golden run (the execution of non disturbed application). The golden run log comprises dynamic image of the program execution (e.g. mnemonics of executed instructions, register states, exit code). Experiment files comprise also details on injected faults. The golden run logs as well as the experiment files are used to fill the developed data warehouse. The data model for this data warehouse is based on the attributes, which specify various aspects of the executed tests and application properties. We have defined 6 classes of attributes:

- *specification of the tested program*: program name, program version, input data set, name of the main module,
- *fault localization in space*: resource area or group, resource item,
- *fault specification*: type (stuck-at-0 or 1, coupling, bit-flip), fault mask (specification of disturbed bits), number of disturbed bits, range of disturbed bits, specification of disturbed bytes,
- *disturbed instruction (DIC)*: mnemonic and code of the result of code corruption,
- *fault triggering instruction (FTI)*: static address of the instruction during which the fault is injected, program module, FTI appearance (i.e. its execution number or loop iteration - it defines dynamic address), FTI code and mnemonic, FTI code group, FTI length in bytes,
- *test result*: correct (C), incorrect (INC), system exception (S), time-out (T).

This list can be extended with parameters specifying application profiles and resource activity (percentage of execution time for which the resource holds some data for the later use [6]). The specified attributes facilitate data exploration in the data warehouse to find critical dependability issues, identify fault susceptibility correlations with various test or application features, etc.

3 Structure of the Developed Data Warehouse

The developed data warehouse (DWS) integrates the following functions: collecting the source data from the experiments (with fault injectors), data transformation and loading the main database (DBS), data aggregation and exploration processes, result presentation. The whole system has been realized using standard data warehouse framework (MS SQL platform) and adapting it to the considered problem by specifying appropriate metadata and data models. The user can interact with the data warehouse by means of standard dimension browser, OLAP, data exploration and visualization tools.

The most important is the definition of the data model, which comprises facts and dimensions. The collected data from the experiments is grouped in topics, which create different hierarchies. These groups are defined by the so-called dimensions, which relate to the attributes listed in section 2. The structure of the main database is correlated with the dimensional data model. In particular the database tables correspond to model entities. Table columns relate to attributes. Each table row corresponds to a data record comprising fields consistent with attribute columns. We distinguish the fact table and dimensional tables. Each record of the fact table represents a single test. Dimensional tables store information on test attributes and their hierarchies. The fact table comprises columns related to the so-called summarizing attributes i.e. describing test results (C, INC, S, T). For the specified test only one column related to the summarizing attributes can hold value 1 (denoting the class of the test result), the remaining hold value 0. These columns allow us to find summarizing reports e.g. related to the percentage of tests of a specified result class etc. Moreover, appropriate relations combine the fact table with dimensional tables. Hence each test in the fact table has also specified attribute values. Relations between table entries are specified by additional columns comprising so-called primary (PK) and foreign keys (FK). The attributes correspond to fault localization, specification, fault triggering instruction (FTI) in the golden run table GRI, its appearance (section 2) etc. The GRI table comprises the specification of instructions executed during the golden run (program, data, address, number of executions, the pointer to FTI attributes etc.).

The list of dimensional tables is defined by the analytical model. This model may comprise some hierarchies of attributes. The hierarchy allows us to define different detail levels of a considered dimension. For example the dimensions corresponding to the fault triggering moment (FTI) has three levels: instruction mnemonic, instruction group and instruction class. Each of these levels is an attribute of tests. Instruction mnemonics relate to a single assembler level instruction mnemonics (e.g. *mov ax bx*). Instruction groups relate to instructions of similar functionality e.g. *mov* (all move instructions), *arithm* (all arithmetic instructions), *branch*, *logic* operations etc. We have defined 11 instruction groups for CPU and 6 for Floating Point Unit. We define 3 instruction classes (CPU instructions, FPU instructions and System instructions). Higher hierarchy levels comprise fewer details. Another example of natural hierarchy is fault localization e.g. CPU registers, their groups (segment, arithmetic and control) and explicitly specified individual registers within these groups. Similarly, we define hierarchy for FPU registers. Within memory space we distinguish data, stack and code subareas. These subareas can be further partitioned. In the case of attributes with no natural hierarchy and a large set of possible values it is useful to introduce an artificial hierarchy levels. Hence, for the purpose of the analysis we define address ranges (specified relatively or explicitly).

The list of dimensional tables and relations between them can be prepared on the basis of the analytical model by assigning for each dimension hierarchy level an entity (table) in the database. This solution introduces many joints, so it is not effective during processing. Hence we have decided to use tables covering the whole dimension hierarchy e.g. a single localization table combining the whole hierarchy. This table comprises records with three columns (attributes): resource localization, subarea and area localization. In a similar way we can also combine tables of different hierarchies e.g. hierarchies of some similarity such as FTI and DIC instruction hierarchy. In practice it is reasonable to create dimensional tables of no more than several hundred thousands of records [11].

4 Statistical and Data Mining Results

We can generate various reports on collected results using standard OLAP tools (e.g. Cube Browser - CuB). These reports are presented in the form of tables. Creating a table we select attributes corresponding to table rows and columns. Table entries comprise numbers related to the selected measures. The measure can be any summarizing attribute e.g. the number of the executed tests, the number (or percentage) of test results within the specified class (C, INC, S, T) compare section 2. To make the reports more readable we can use slicing which narrows the analysis to a specified subset of tests defined by appropriate conditions (expressed in function of various attribute values). Various statistics over individual applications or specified classes of applications can be easily generated from the data warehouse.

For an illustration we give a statistic of test terminated by timeouts (T) in relevance to fault triggering instructions (over many applications):

flag - 8.4%, fcom 8.0%, fcterl 7.6%, lpgic 6%, farithm 3.7%, misc 2.6%, branch 1.9%, arithm 1%, for the remaining instructions T < 1%.

The main database (DBS) is used in data mining to find some rules facilitating to understand fault susceptibility in function of various application features, fault location etc. Due to the large number of attributes we uses data mining based on decision trees. The decision tree comprises one root node, internal nodes and leaves. Each internal node and the root are related to a specified conditional attribute. Leaves relate to the decision attributes. Each branch coming out from a node relates to a specified value of its attribute. In our system each node relates to a set of tests, which fulfill the condition on the path from the root to the considered node. The leaf, in addition, comprises values of the decision attributes i.e. the category of test result (C, INC, S, T). In the ideal classification the leaf should comprise tests with results belonging to one class. In the case of fault injection experiments it is better to deal with rough tree fitting in which a leave may comprise a set of tests corresponding to various result classes. The class with maximal cardinality (or percentage) is called dominating. In such trees we identify rules with the confidence level below 100%. The trees are generated according to the algorithm, which uses Buntine measure to find optimal node partitioning [11]. The algorithm used in SQL Server has two parameters, which has to be defined by the user: the number of minimum test cases in the leaf and the tree complexity penalty (fraction in the range [0,1]). Moreover, we have to select the attributes and specify their types. We use nominal and ordered types. The nominal type is used if the set of the values assumed by the attribute is finite. The algorithm applies only operator $=$ and $\neq$ to define comparison on the tree branches. Repeating attributes in a path we can perform more complex comparisons. We use nominal type for attributes of fault localization and mnemonic of instruction FTI. The ordered attribute type relates to the set of ordered values (e.g. instruction execution number - integer value) and here we can define value intervals using relations $<, >, \geq, \leq$.

The data mining process involves generation of various decision trees and finding rules on fault susceptibility. The most interesting are the rules with the high confidence level. We create separate decision trees for test results of 4 classes: C (correct), INC (incorrect), S (system exceptions) and T (time-outs) by selecting appropriate binary decision attribute. In this case the tree leaves relate to the considered result class or its complement (the one which is dominant). It is also possible to generate the decision tree related to all result classes (multi-valued decision attribute), where different leaves may comprise different dominating result classes. While analyzing the trees we trace paths leading to leaves with dominating tests or leaves comprising a small number of tests for the considered result class. For example, nodes with dominating C category indicate fault tolerance capabilities and those with a small number of C category results indicate fault tolerance problems. An important issue is the selection of conditional attributes, their types and ranges of values. For an illustration in tab. 2 we give possible partitions of a node in a tree (over many applications). Partition *b* for attribute FTI (section 2) instruction length is less interesting than partition *a* because it gives results closer to the root node. Partition *a* gives leaves with dominant tests for C and S results. Partition *c* (for the attribute specifying the number of injected faults in test) is also interesting due to dominant classes C and INC in the leaves.

Table 2. A sample of decision tree partitions

parent node, (C, INC, S, T) = (60, 25, 15, 0) in percents					
FTI instruction length attribute L (bytes)				Number of faults N	
Partition *a*		Partition *b*		Partition *c*	
L<4	L≥4	L<5	L≥5	N=1	N=2
$(70, 28, 2, 0)$	$(30, 15, 55, 0)$	$(65, 27, 8, 0)$	$(45, 19, 36, 0)$	$(65, 15, 20, 0)$	$(45, 55, 0, 0)$

Table 3. Test result distribution according to fault localization attribute

Test	Fault localization							
results	alu	regs	seg	fregs	fst	code	data	stack
C	92%	26%	32%	82%	59%	33%	45%	74%
INC	4%	5%	0%	15%	41%	30%	34%	7%
S	4%	69%	68%	3%	0%	32%	21%	19%
T	0%	0%	0%	0%	0%	5%	0%	0%

Data mining was successfully used in identifying factors influencing fault susceptibility, error detection effectiveness (e.g. related to exceptions), finding the most critical points in fault hardened applications, etc. In the sequel we illustrate this with a sample of results. The first example relates to the analysis of fault injection results of 3 programs: P1 (Taylor series), P2 (bublesort) and P3 (LZW compression). The decision tree with decision attribute S was generated by DMMB program. Test result distribution for the root node was: (C, INC, S, T) = (48%, 22%, 28%,2%). The first conditional attribute was fault localization. The distribution of test results in the nodes related to this attribute is given in tab. 3. It confirms the significance of the selected attribute. The embedded fault detection mechanisms generating system exceptions (S) are most effective for fault disturbing specialized (regs) and segment (seg) registers (about 70% detected faults). Time-outs (T) practically appear for faults injected into program code. The next conditional attribute in the tree just after the node with fault localization related to code was program. It generated two child nodes related to program P3 and the remaining programs. For P3 we have got S=5% and other test results 95%; for the remaining programs it was 40% and 60%, respectively. It is interesting to note that for P3 correct results (C) contributed 90% (faults injected into code), which confirms its relatively high fault tolerance.

The generated decision trees we analyze in a systematic way to identify interesting conditions. We do this in some heuristic and recursive way by tracing paths, starting from the root we select subsequent node if the percentage of the considered test class is higher than in the parent node. Finding conditions leading to interesting behaviors e.g. dominating INC class we face the problem of selecting conditions with appropriate generality. This problem we illustrate for a tree generated for time-out results (T) of programs P1-P3 (193 100 tests in the root n1 node). Systematic tracing of the tree paths leads to the path with high time-out percentage, shown in tab. 4.

Table 4. Example of a selected path in time-out decision tree (programs P1−P3)

Node	n1-root	n2	n3	n4	n5	n7
attribute	-	code	P1	[247,250]	msb$\leq$ 11	msb$\neq$ 5
T	2%	5%	10%	25%	34%	38%

The presented path relates to the following attributes (and their values): fault localization (code), program (P1), relative address range of disturbed code ([247,250]), disturbed bits in the code (any bits from 0 to 11-for node n5 and any bits except bit number 5 for node n6). This path related to the highest percentage of time-outs (T). In fact the specified condition (including 5 attributes) is too detailed. Practicality, we can skip the last attribute (msb$\neq$ 5), which in fact improves slightly result classification from 34% (2600 tests) to 38% (2400 tests). Here it is worth noting that timeouts occur rarely, so the classification levels over 30% can be considered as satisfactory. Much higher level of classification confidence can be achieved for other test results e.g. correct results (C). The decision tree generated for C results of programs P1-P3 shows the following path with good classification: root (C=45%), fault localization: the main module of program P1 (C = 54%) and fault localization universal register (C = 92%).

An interesting issue is finding critical code areas in the application. For a matrix multiplication program with embedded row and column checksums (to detect and correct faults) the generated tree identified code subareas of different fault leakage (INC). The tree showed 5 subsequent code subareas (specified in brackets, which give the number of comprised bytes B) with the following distribution of INC results:

(507 B): 0%, (151 B); 11%, (23 B): 27%, (2937 B): 3%, (8 B): 0%

Hence, we identified code bytes 507-680 (constituting less than 5% of the code) as most sensitive to faults ($11\% \leq INC \leq 27\%$). This code can be improved. For a bubble sort application the MS Analysis Server generated a partition of 4 code ranges (specified by relative addresses) with the following percentage of time-outs (T):

[0,26]: 6%, [27,65]: 0%, [66,82]: 25%, [83,182]: 6%

For the whole application we obtained on average T < 5%, and a small code area [66-82] is much more susceptible to timeouts (27%). The relative ranges can be mapped into real addresses (748 bytes of the whole code), so the considered area [66-82] comprises 86 bytes (11% of the application code). The decision tree generated with CuB browser is a little bit different - shown in tab. 5. From this table we can find that faults injected closer to the end of the program generate the lowest percentage of incorrect results (INC) and more correct results. The percentage of correct results is significantly higher for the second half of the program, generated system exceptions practically are not correlated with the address ranges. Some address areas are more susceptible to time-outs. For comparison we give result distribution for the whole program: (C, INC, S, T) = (15%, 29%, 51%, 5%).

In this way we have analyzed many applications taking into account various aspects related to fault susceptibility, application specificity, configuration of fault injection experiments etc. The available standard tools, allow us to drill out sources of identified anomalies in application behavior.

Table 5. Address partitioning for Bubblesort program provided by CuB browser

Test	Address ranges of FTI instructions						
results	[0,31]	[32,48]	[49,65]	[66,86]	[87,112]	[113,145]	[146,182]
C	5%	13%	14%	13%	36%	36%	35%
INC	38%	37%	39%	15%	4%	5%	7%
S	52%	50%	47%	51%	60%	53%	47%
T	5%	0%	0%	21%	0%	6%	11%

5 Conclusion

Developing data warehouse the most important issue is the definition of the data model. This allows the user to generate the required system basing on standard data base and data mining platforms. This system has to be supported with appropriate data loading, and transformation procedures. The developed system was targeted at the analysis of fault simulation results generated in many experiments by various tools. In the data mining processes we based on decision trees and found them useful in identifying some critical points in the analyzed applications. The results were not trivial (difficult to find in a manual analysis.). Here it is worth noting that the data exploration process is not fully automated. Practically the user is provided with appropriate tools and should act in an interactive way to find interesting rules, relations etc. To facilitate data exploration we have proposed an original scheme of tracing decision trees.

As far as new experiments with different applications are performed the data warehouse is filled systematically with new data and the exploration process covers larger spectrum of dependability features. This may result in a useful knowledge database for this domain. Recently, we have initiated a new research, which takes into account other data mining techniques. Rough set approach will be considered also.

References

1. Arlat, J., et al.: Comparison of physical and software implemented fault injection technique. IEEE Trans. on Computers 52(8), 1115–1133 (2003)
2. Benso, A., Prinetto, P.: Fault injection techniques and tools for embedded systems reliability evaluation. Kluwer Academic Publishers, Dordrecht (2003)
3. Bradley, P., Fayyad, U., Reina, C.: Scaling EM clustering to large databases. Microsoft Research, November 1998, rev. (October 1999)
4. Chaudhuri, S., Fayyad, U.M., Bernhardt, J.: Scalable Classification over SQL Databases, pp. 470–479 (1999) ICDE 1999: ftp://ftp.research.microsoft.com/users/surajitc/icde99.pdf
5. Gawkowski, P., Sosnowski, J.: Dependability evaluation with fault injection experiments. IEICE Trans. Inf. & Syst. E86-D(12), 2642–2649 (2003)
6. Gawkowski, P., Sosnowski, J., Radko, B.: Analyzing the effectiveness of fault hardening procedures. In: Proc. of the 11th IEEE Int'l On-Line Testing Symp, pp. 14–19 (2005)
7. Han, J., Kamber, M.: Data mining, concepts and techniques. Morgan Kaufman Pub, San Francisco (2001)
8. Karnik, T., Hazucha, P., Patel, J.: Characterization of soft errors caused by single event upsets in CMOS processes. IEEE Trans. on Dependable and Secure Comp. 1(1), 128–143 (2004)
9. Sosnowski, J., Gawkowski, P., Lesiak, A.: Software implemented fault inserters. In: Proc. of IFAC PDS 2003, Workshop, Pergamon, pp. 293–298 (2003)
10. Analysis services, Data mining. Business Intelligence and data warehousing in SQL, Server 2005, Microsoft TechNet. http://www.microsoft.com/technet/prodtechnol/sql/2005/default.mspx
11. DMFAQ Algorithms. Creating solutions with SQL Server Technologies. MSDN 2005 (2005)

Classification of Complex Structured Objects on the Base of Similarity Degrees

Piotr Hońko

Department of Computer Science
Białystok University of Technology
Wiejska 45A, 15-351 Białystok, Poland
phonko@ii.pb.bialystok.pl

Abstract. In the paper, we propose an algorithm for classification of complex structured objects. The objects, expressed in a first-order logic (FOL) language, are positive and negative examples of a target relation. In the process of searching for a classification pattern, a similarity measure and some notions of rough set theory are applied. We search for the pattern being a similarity degree of examples and satisfying two conditions: the number of positive examples similar at least to the degree to other positive examples is highest and the number of negative examples similar at least to the degree to positive examples is lowest. The obtained set of similar examples corresponds to the lower or to the upper approximation of a set of all positive examples. The found similarity degree is applied in classification of new examples. An example is classified as positive if it belongs to the approximation computed with respect to the degree, and it is classified as negative, otherwise.

Keywords: first-order logic, rough sets, similarity measures.

1 Introduction

In a lot of applications aimed at classification of complex structured objects expressed in a FOL language, an inductive logic programming (ILP)[2,4] approach has been used. In ILP, there are given a set E^+ of positive and a set E^- of negative examples of a target relation, and a background knowledge B usually consisting of Horn clauses. The task is to find a hypothesis H (Horn clauses) such that H covers all positive examples and does not cover any negative example. Our approach is related to the task described above, but it is not within an ILP framework. We consider sets E^+, E^- and B consisting of literals without variables. In construction of a classification algorithm, we apply some similarity measure and notions of rough set theory [5,6,7,8]. In our approach, the classification pattern is a similarity degree of examples. In order to find the degree, we consider a set of candidates being real numbers in the range of 0 to 1. For each of them, we check how many positive and negative examples are similar to other positive examples. We select a degree for which the number of similar positive (negative) examples is highest (lowest). The obtained set of similar examples

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 553–563, 2007.

corresponds to the lower or to the upper approximation of set E^+. The type of approximation can be fixed by a user. New examples are classified on the base of the degree found over a training dataset. An example is classified as positive if it belongs to the approximation computed with respect to the degree, and it is classified as negative, otherwise.

The rest of the paper is organized as follows: in Sect. 2, we present a general way of defining of similarity of examples. Section 3 presents some notions of rough set theory applied in our approach. In Sect. 4, an algorithm computing a similarity degree and classifying new examples is proposed. Results of experiments, performed by applying the approach proposed in the paper, are presented in Sect. 5.

2 Similarity Degree of Examples

We apply some similarity measure to target examples in order to compare them. As a result, we obtain a similarity degree of examples that is a real number in the range of 0 to 1. The examples can be then said to be similar to each other at least to the degree. If the degree is:

- 0, then examples are not similar (i.e., they are dissimilar);
- 1, then examples are totally similar.

Similarity of examples is defined by using *supporting sets*. Let $term(l)$ denote the set of terms of a literal l.

Definition 1. *An example e is supported by a literal l if and only if*

$$term(e) \cap term(l) \neq \emptyset.$$

We assume target examples to be as similar as background literals supporting the target examples. When comparing two examples, we first compute their supporting sets (i.e., a set of literals supporting the example). Let $supp(e)$ denote the set of literals supporting an example e. In our approach, relations among unstructured objects (represented by terms) are more important than their concrete real or symbolic values. In order to capture these relations, we generalize literals of supporting sets by replacing constants with variables. In the process of generalization, we obtain the set $supp_{gen}(e)$ of literals, where each term is a variable. The generalization can be carried out according to the algorithm presented below. Let $term(e) = \{t_1, t_2, ..., t_n\}$, for any example e.

```
$supp_{gen}(e)$
begin
  $T := term(e); n := card(T); m := n; S := \emptyset$;
  for $i := 1$ to $n$ do associate a variable $v_i$ with a term $t_i$;
  for each literal $l \in supp(e)$ do
  begin
    for each $t \in term(l)$ do
    begin
      if $t \notin T$ then
      begin
        $m := m + 1; t_m := t; T := T \cup \{t_m\};$
        associate a variable $v_m$ with a term $t_m$;
      end;
      replace $t$ in $l$ with $v$ associated with $t$;
    end;
    $S := S \cup \{l\};$
  end;
  return $S$;
end;
```

Finally, by computing similarity of examples, we compare literals of the generalized supporting sets. For example, we can compute a similarity degree of examples e, e' by applying the following function.

Definition 2. *A similarity degree $e\_sim\,(e, e')$ of examples e, e' is defined by*

$$e\_sim\,(e, e') = \begin{cases} \frac{card(S \cap S')}{card(S \cup S')} & \text{if } S \cap S' \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

where $S = supp_{gen}\,(e)$ and $S' = supp_{gen}\,(e')$.

One can consider more advanced similarity measures, for example by applying distance measures to literals.

3 Similarity Degree in Construction of Approximation Space

In this section, we recall the general definition of an approximation space [7], [8] and also propose a way of application of approximations in our approach. Let $P\,(U)$ denote the set of all subsets of a non-empty set U.

Definition 3. *A parameterized approximation space is a system $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$, where*

- *U is a non-empty set of objects,*
- *$I_{\#} : U \rightarrow P\,(U)$ is an uncertainty function,*
- *$\nu_{\$} : P\,(U) \times P\,(U) \rightarrow [0, 1]$ is a rough inclusion function.*

For every object, the uncertainty function defines a set of similarly described objects. A set $X \subseteq U$ is *definable in $AS_{\#,\$}$* if and only if it is a union of some values of the uncertainty function.

The rough inclusion function defines the degree of inclusion of a set X in a set Y, where $X, Y \subseteq U$. We consider two rough inclusion functions in our approach, namely the standard rough inclusion and the rough inclusion of the variable precision rough set model (VPRSM)[10].

Definition 4. *The standard rough inclusion $\nu_{SRI}(X,Y)$ of a set X in a set Y is defined by*

$$\nu_{SRI}(X,Y) = \begin{cases} \frac{card(X \cap Y)}{card(X)} & \textit{if } X \neq \emptyset \\ 1 & \textit{if } X = \emptyset \end{cases}$$

Definition 5. *The rough inclusion $\nu_{l,u}(X,Y)$ of a set X in a set Y is defined by*

$$\nu_{l,u}(X,Y) = f_{l,u}(\nu_{SRI}(X,Y)),$$

$$\textit{where } f_{l,u}(t) = \begin{cases} 0 & \textit{if } 0 \leq t \leq l \\ \frac{t-l}{u-l} & \textit{if } l < t < u \\ 1 & \textit{if } \quad t \geq u \end{cases}$$

$$\textit{and } 0 \leq l < u \leq 1.$$

Note that if $l = 0$ and $u = 1$, then the rough inclusion $\nu_{l,u}$ is equivalent to the standard rough inclusion ν_{SRI}.

The lower and the upper approximations of subsets of U are defined as follows.

Definition 6. *For an approximation space $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$ and any subset $X \subseteq U$, the lower and the upper approximations are defined by*

$$LOW\left(AS_{\#,\$}, X\right) = \{x \in U : \nu_{\$}(I_{\#}(x), X) = 1\},$$
$$UPP\left(AS_{\#,\$}, X\right) = \{x \in U : \nu_{\$}(I_{\#}(x), X) > 0\}, \textit{ respectively.}$$

Approximations of concepts (sets) are constructed on the base of background knowledge. Symbols $\#, \$$ denote vectors of parameters which can be tuned in the process of concept approximation.

We adapt the lower and the upper approximations for subsets of the set of target examples. Let $U = E^{+} \cup E^{-}$.

Proposition 1. *For a similarity measure e_sim of examples, a degree d and any example $x \in U$, the uncertainty function is defined by*

$$I_d^{e\_sim}(x) = \{y \in U : e\_sim(x,y) \geq d\}.$$

Proposition 2. *For an approximation space $AS_d = \left(U, I_d^{e\_sim}, \nu_{l,u}\right)$ and any subset $X \subseteq U$, the lower and the upper approximations are defined by*

$$LOW(AS_d, X) = \left\{x \in U : \nu_{l,u}\left(I_d^{e\_sim}(x), X\right) = 1\right\},$$
$$UPP(AS_d, X) = \left\{x \in U : \nu_{l,u}\left(I_d^{e\_sim}(x), X\right) > 0\right\}, \textit{ respectively.}$$

4 Similarity Degree as Classifier

In this section, we present an algorithm for classification of examples on the base of a similarity degree. The degree is found over a training dataset. In our approach, we consider the lower (or the upper) approximation of set E^{+}. The type of approximation can be fixed by a user. Belonging of an example to the approximation depends on a degree under consideration. The idea is to find a degree for which the number of positive (negative) examples belonging to the approximation is highest (lowest). Each example of the approximation is then treated as positive. For practical reasons, we consider the set $I_d^{e\text{-}sim}(e) \setminus \{e\}$ instead of the set $I_d^{e\text{-}sim}(e)$. In order to increase the lower approximation and to decrease the upper one, we apply the rough inclusion of the VPRSM. Therefore, we consider two parameters l and u, called *precision control parameters.*

Let $X = (E, B)$ be a pair representing a dataset, where $E = E^+ \cup E^-$ and B is a background knowledge. The function computing the lower and the upper approximations with respect to a degree d is defined in the following way.

```
$Compute\_App\,(X, X', e\_sim, d, l, u)$
begin
  $LOW := \emptyset; UPP := \emptyset;$
  if $X' \neq \emptyset$ then $E := X'.E$; else $E := X.E$;
  for each $e \in E$ do
  begin
    $t := \nu_{SRI}\left(I_d^{e\text{-}sim}(e) \setminus \{e\}, X.E^+\right);$
    if $t > l$ then $UPP := UPP \cup \{e\};$
    if $t \geq u$ then $LOW := LOW \cup \{e\};$
  end;
  return $(LOW, UPP)$;
end;
```

The function can be applied to a training dataset and also to the test one. In both cases, X is a training dataset. If we compute approximations for the training dataset, then X' is the empty one. By computing approximations for a test dataset, X' is the test one. In case of the training dataset, (LOW, UPP) is a pair of the lower (i.e., LOW) and the upper (i.e., UPP) approximations of E^+. In case of the test dataset, set LOW (UPP) consists of test examples similar with respect to parameter u (l) to examples of E^+.

We propose the following method generating a classifier (i.e., a similarity degree) by applying function *Compute_App*.

```
$Generate\_Classifier\,(X, e\_sim, C, l, u, app)$
begin
  $v' := 0; C' := \emptyset;$
  for each $c \in C$ do
  begin
    $APP := Compute\_App\,(X, \emptyset, e\_sim, c, l, u);$
    if $app = lower$ then $A := APP.LOW$; else $A := APP.UPP$;
    $pos := \nu_{SRI}(X.E^+, A); neg := \nu_{SRI}(X.E^-, A);$
    $v := pos + (1 - neg);$
    if $v = v'$ then $C' := C' \cup \{c\};$
    else if $v > v'$ then begin $C' := \{c\}; v' := v$; end;
    end;
    return $\frac{min(C')+max(C')}{2}$;
end;
```

Here X is a training dataset, and C is a fixed set of candidates to be the classifier. One can fix which examples are treated as positive, namely the examples belonging to the lower approximation (i.e., $app = lower$) or to the upper one (i.e., $app = upper$). Here function ν_{SRI} is applied to compute the degree of inclusion of the set of positive (negative) examples in the approximation.

We illustrate the notions presented above by means of an example.

Example 1. Given a dataset:

$E^+ = \{e_1, e_2, e_3, e_4\}, E^- = \{e_5, e_6, e_7, e_8\}$,
where $e_1 = father\,(6,5), e_2 = father\,(8,1), e_3 = father\,(1,3)$,
$e_4 = father\,(2,4), e_5 = father\,(6,7), e_6 = father\,(4,1)$,
$e_7 = father\,(8,2), e_8 = father\,(7,6)$;
$B = \{male\,(1), male\,(2), male\,(3), male\,(5), male\,(6), male\,(8), male\,(9)$,
$parent\,(1,3), parent\,(2,4), parent\,(4,8), parent\,(6,5), parent\,(7,3)$,
$parent\,(8,1), parent\,(9,7)\}$.

We compute the supporting sets of the examples. For examples e_1 and e_7, we obtain:

$supp\,(e_1) = \{male\,(5), male\,(6), parent\,(6,5)\}$,
$supp\,(e_7) = \{male\,(2), male\,(8), parent\,(2,4), parent\,(4,8), parent\,(8,1)\}$.
Replacing constants with variables, we obtain:
$supp_{gen}\left(e_{1(v_1,v_2)}\right) = \{male\,(v_1), male\,(v_2), parent\,(v_1,v_2)\}$,
$supp_{gen}\left(e_{7(v_1,v_2)}\right) = \{male\,(v_2), male\,(v_1), parent\,(v_2,v_3), parent\,(v_3,v_1)$,
$parent\,(v_1,v_3)\}$,
where $(v_1,\ldots,v_n)$ in $e_{(v_1,\ldots,v_n)}$ is a tuple of variables associated with terms of an example e.

We obtain the following similarities for all examples:

$e\_sim\,(e,e')$	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_1	1	0.6	0.6	0.5	0.17	0.14	0.34	0.17
e_2	0.6	1	0.67	0.6	0.29	0.43	0.67	0.5
e_3	0.6	0.67	1	0.34	0.29	0.43	0.43	0.29
e_4	0.5	0.6	0.34	1	0.4	0.14	0.34	0.17
e_5	0.17	0.29	0.29	0.4	1	0.5	0.5	0.34
e_6	0.14	0.43	0.43	0.14	0.5	1	0.67	0.8
e_7	0.34	0.67	0.43	0.34	0.5	0.67	1	0.8
e_8	0.17	0.5	0.29	0.17	0.34	0.8	0.8	1

Let $l = 0.3, u = 0.7$ and $C = \{0.5, 0.6\}$. We compare the qualities of candidates $c_1 = 0.5$ and $c_2 = 0.6$. For c_1, we obtain:

$I_{c_1}^{e\_sim}(e_1) = E^+, I_{c_1}^{e\_sim}(e_2) = E^+ \cup \{e_7, e_8\}, I_{c_1}^{e\_sim}(e_3) = \{e_1, e_2, e_3\}$,
$I_{c_1}^{e\_sim}(e_4) = \{e_1, e_2, e_4\}, I_{c_1}^{e\_sim}(e_5) = \{e_5, e_6, e_7\}, I_{c_1}^{e\_sim}(e_6) = \{e_5, e_6, e_7, e_8\}$,

$I_{c_1}^{e\_sim}(e_7) = \{e_2, e_5, e_6, e_7, e_8\}, I_{c_1}^{e\_sim}(e_8) = \{e_2, e_6, e_7, e_8\}.$
For examples e_1, e_7, we obtain $\nu_{l,u}\left(I_{c_1}^{e\_sim}(e_1) \setminus \{e_1\}, E^+\right) = 1 > u$,
$\nu_{l,u}\left(I_{c_1}^{e\_sim}(e_7) \setminus \{e_7\}, E^+\right) = 0.25 < l$. Hence, $e_1 \in LOW$ and $e_7 \notin UPP$.

By considering the lower approximation (i.e., $app = lower$), we obtain:
$LOW = \{e_1, e_3, e_4\}, pos_1 = \frac{card(LOW)}{card(E^+)} = 0.75, neg_1 = \frac{card(LOW)}{card(E^-)} = 0.$
Hence, $v_1 = pos_1 + (1 - neg_1) = 1.75$.
In case of the upper approximation (i.e., $app = upper$), we obtain:
$UPP = E^+ \cup \{e_8\}, pos_1' = \frac{card(UPP)}{card(E^+)} = 1, neg_1' = \frac{card(UPP)}{card(E^-)} = 0.25.$
Hence, $v_1' = pos_1' + (1 - neg_1') = 1.75$.
For c_2, we obtain:

$I_{c_2}^{e\_sim}(e_1) = \{e_1, e_2, e_3\}, I_{c_2}^{e\_sim}(e_2) = E^+ \cup \{e_7\}, I_{c_2}^{e\_sim}(e_3) = \{e_1, e_2, e_3\},$
$I_{c_2}^{e\_sim}(e_4) = \{e_2, e_4\}, I_{c_2}^{e\_sim}(e_5) = \{e_5\}, I_{c_2}^{e\_sim}(e_6) = \{e_6, e_7, e_8\},$
$I_{c_2}^{e\_sim}(e_7) = \{e_2, e_6, e_7, e_8\}, I_{c_2}^{e\_sim}(e_8) = \{e_6, e_7, e_8\}.$
For examples e_1, e_7, we obtain $\nu_{l,u}\left(I_{c_2}^{e\_sim}(e_1) \setminus \{e_1\}, E^+\right) = 1 > u$,
$\nu_{l,u}\left(I_{c_2}^{e\_sim}(e_7) \setminus \{e_7\}, E^+\right) = 0.34 > l$. Hence, $e_1 \in LOW$ and $e_7 \in UPP$.

By considering the lower approximation, we obtain:

$LOW = \{e_1, e_2, e_3, e_4\}, pos_2 = \frac{card(LOW)}{card(E^+)} = 1, neg_2 = \frac{card(LOW)}{card(E^-)} = 0.$
Hence, $v_2 = pos_2 + (1 - neg_2) = 2$.

In case of the upper approximation, we obtain:

$UPP = E^+ \cup \{e_7\}, pos_2' = \frac{card(UPP)}{card(E^+)} = 1, neg_2' = \frac{card(UPP)}{card(E^-)} = 0.25.$
Hence $v_2' = pos_2' + (1 - neg_2') = 1.75$.

For the lower approximation, we obtain that $v_1 = 1.75 < v_2 = 2$. Therefore, c_2 is a better candidate for the classifier than c_1. For the upper approximation, we obtain that $v_1' = v_2' = 1.75$. Therefore, we take $c_3 = \frac{c_1+c_2}{2} = 0.55$ as the classifier.

New examples (i.e., test examples) are classified on the base of the classifier found over the training set. An example is classified as positive if it belongs to the approximation (computed with respect to the classifier) of set E^+ and it is classified as negative, otherwise. We classify new examples on the base of the following algorithm.

```
$Classify\_Examples(X, X', e\_sim, C, l, u, app)$
begin
   $c := Generate\_Classifier(X, e\_sim, C, l, u, app)$;
   $APP := Compute\_App(X, X', e\_sim, c, l, u)$;
   if $app = lower$ then
   begin $POS := APP.LOW; NEG := X'.E - POS$; end;
   else begin $POS := APP.UPP; NEG := X'.E - POS$; end;
   return $(POS, NEG)$;
end;
```

Here X is a training dataset and X' is the test one. The algorithm returns a pair of sets POS and NEG consisting of examples classified as positive and negative, respectively.

Example 2. Consider the training dataset given in Example 1, the same parameters $l = 0.3, u = 0.7$ and generated classifiers, namely $c_2 = 0.6$ for the lower approximation and $c_3 = 0.55$ for the upper one. We classify examples of the following test dataset:

$E' = \{e_9, e_{10}\}, B' = B$, where $e_9 = father\,(5, 2), e_{10} = father\,(9, 7)$.

We obtain:

$supp\,(e_9) = \{male\,(2), male\,(5), parent\,(2, 4), parent\,(6, 5)\}$,
$supp\,(e_{10}) = \{male\,(9), parent\,(7, 3), parent\,(9, 7)\}$.

Hence:
$supp_{gen}\left(e_{9(v_1,v_2)}\right) = \{male\,(v_2), male\,(v_1), parent\,(v_2, v_3), parent\,(v_3, v_1)\}$,
$supp\left(e_{10(v_1,v_2)}\right) = \{male\,(v_1), parent\,(v_2, v_3), parent\,(v_1, v_2)\}$.

We obtain the following similarities for examples e_9, e_{10}:

$e\_sim\,(e, e')$	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
e_9	0.4	0.8	0.5	0.4	0.34	0.5	0.8	0.6
e_{10}	0.5	0.6	0.34	1	0.4	0.14	0.34	0.17

If $app = lower$, then we use classifier c_2 and obtain:

$I^{e\_sim}_{c_2}(e_9) = \{e_2, e_7, e_8, e_9\}$ and $I^{e\_sim}_{c_2}(e_{10}) = \{e_2, e_4, e_{10}\}$.
We have: $\nu_{l,u}\left(I^{e\_sim}_{c_2}(e_9) \setminus \{e_9\}, E^+\right) = 0.34 < u$,
and $\nu_{l,u}\left(I^{e\_sim}_{c_2}(e_{10}) \setminus \{e_{10}\}, E^+\right) = 1 > u$. Hence, $e_9 \notin LOW$ and $e_{10} \in LOW$. Therefore $POS = \{e_{10}\}$ and $NEG = \{e_9\}$.

If $app = upper$, then we use classifier c_3 and obtain:

$I^{e\_sim}_{c_3}(e_9) = \{e_2, e_7, e_8, e_9\}$ and $I^{e\_sim}_{c_3}(e_{10}) = \{e_2, e_4, e_{10}\}$.
We have: $\nu_{l,u}\left(I^{e\_sim}_{c_3}(e_9) \setminus \{e_9\}, E^+\right) = 0.34 > l$,
and $\nu_{l,u}\left(I^{e\_sim}_{c_3}(e_{10}) \setminus \{e_{10}\}, E^+\right) = 1 > l$. Hence $e_9, e_{10} \in UPP$. Therefore $POS = \{e_9, e_{10}\}$ and $NEG = \emptyset$.

5 Experiments

In this section, we present results of some experiments performed by our algorithm. It is implemented in C++ language and executed on a PC with CPU 1.84 GHz and 256 MB RAM. We use two datasets in our experiments. The first one is related to the document understanding (DocUnd) [3] and describes components of single page documents. Target predicates (i.e., target relations) are related to the following components: sender, receiver, date, logo and references. Background predicates describe properties of the components and relationships with other components. Since there is more than one target predicate, we consider in each step literals of one predicate as positive examples and literals of other predicates as the negative ones.

The second dataset is related to the family relations and it is created on the base of the Family dataset provided with the FORTE system [1]. We consider the following target predicates: wife, husband, mother, father, daughter, son, sister and brother. Background predicates describe other relations between persons. In case of this dataset, each target predicate has positive and negative examples.

In experiments, we apply two pair of precision control parameters, namely $u = 1, l = 1 - u = 0$ and $u = 0.7, l = 1 - u = 0.3$. In both cases, we take the lower and next the upper approximation as the set of positive examples. Table 1 presents the results of experiments, total time of execution of the algorithm (i.e., time of generation of pattern and time of classification of objects), as well as valuations of classification. The first valuation is related to parameter and approximation, namely $val_1 = \frac{p+(100-n)}{2}$, where p (n) is the percentage of positive (negative) examples classified as positive. The second one is related to approximation, namely $val_2 = \frac{val_1^1 + val_1^{0.7}}{2}$, where val_1^u is val_1 for parameter u. As experiments show, the results depend on both the factors i.e., parameter and approximation. In general, if the parameter u is lower, then the result (i.e., val_1) is better or the same. This difference is more visible in case of the lower approximation. The general result (i.e., val_2) of the approximation is insignificantly better in the case of the upper one.

Table 1. Percentage of positive and negative examples classified as positive

dataset	DocUnd				Family			
app	LOW		UPP		LOW		UPP	
parameter	u=1	u=0.7	u=1	u=0.7	u=1	u=0.7	u=1	u=0.7
positive	84.29	96.31	92.07	93.77	76.54	93.24	95.75	95.75
negative	0	2.79	03.26	04.31	1.53	6.50	7.01	7.01
time(sec.)	0.88	0.88	0.89	0.88	0.42	0.44	0.44	0.43
val_1	92.14	96.76	94.41	94.73	87.50	93.37	94.37	94.37
val_2	94.45		94.57		90.44		94.37	

6 Conclusions and Future Research

In the paper, we propose an algorithm for classification of complex structured objects understood as target examples. Notions of rough set theory are adapted in construction of the algorithm. A similarity degree, computed by the algorithm over a training dataset, is treated as a classifier in case of the test one. The algorithm is designed for datasets characterized by the following features: consideration of values of literal terms is not essential; background knowledge, restricted to literals of supporting sets, is sufficient to distinguish positive examples from the negative ones.

An essential advantage of the approach is the size of the generated pattern. Regardless of a dataset, the pattern is one real number in the range of 0 to 1. While classifying test objects, referring to a training dataset may be some inconvenience. On the base of the results of experiments, one can observe that effectiveness of our approach depends on selection of the precision control parameters and the type of approximation. A measure, applied to compute a similarity degree, may has a significant influence on the results of experiments. More advanced similarity measures will be considered in future research. The second subject of future research is another way of partition of a set of considered objects. In the approach presented in the paper, we treat complex structured objects as examples of a target relation. If a dataset includes more than one target relation, we divide a set of examples into two sets in the following way. The first set consists of positive examples of the considered target relation, while the second one consists of positive examples of other relations (i.e., negative examples of the target relation). One can apply another partition of the set of objects. Namely, each relation, understood as a set of objects belonging to the relation, is considered separately. Therefore, we compute a classifier for each relation. New objects are classified to the most similar relation.

Acknowledgements

The research has been supported by Bialystok University of Technology grant S/WI/5/03.

References

1. Bradley, L., Mooney, R., Mooney, R.J.: Refinement of first-order Horn-clause domain theories. Machine Learning 19(2), 95–131 (1995)
2. Dzeroski, S.: Inductive logic programming and knowledge discovery in databases. In: Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.) Advances in Knowledge Discovery and Data Mining, AAAI Press / The MIT Press, pp. 117–152 (1997)
3. Esposito, F., Malerba, D., Semerano, G., Pazzani, M.: A Machine learning approach to document understanding. In: Michalski, R. S., Tecuci, G. (eds.) Proceedings of the Second International Workshop on Multistrategy Learning, pp. 276–292 (1993) ftp://ftp.mlnet.org/ml-archive/general/data/doc-understanding/
4. Muggleton, S., De Raedt, L.: Inductive logic programming: theory and methods, Journal of Logic Programming, Special Issue on Ten Years of Logic Programming (1994)
5. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
6. Polkowski, L., Skowron, A. (eds.): Rough Sets in Knowledge Discovery 1 and 2. Physica-Verlag, Heidelberg (1998)
7. Skowron, A., Stepaniuk, J.: Tolerance Approximation Spaces. Fundamenta Informaticae 27, 245–253 (1996)

8. Stepaniuk, J.: Knowledge discovery by application of rough set models. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.) Rough Set Methods and Applications, New Developments in Knowledge Discovery in Information Systems, pp. 137–233. Physica-Verlag, Heidelberg (2000)
9. Stepaniuk, J., Hońko, P.: Learning first-order rules: A rough set approach. Fundamenta Informaticae 61(2), 139–157 (2004)
10. Ziarko, W.: Variable precision rough sets model. Journal of Computer and Systems Sciences 46(1), 39–59 (1993)

Application of Parallel Decomposition for Creation of Reduced Feed-Forward Neural Networks

Jacek Lewandowski[1], Mariusz Rawski[2], and Henryk Rybinski[1]

[1] ICS, Warsaw University of Technology
[2] IT, Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
j.lewandowski,hrb@ii.pw.edu.pl,rawski@tele.pw.edu.pl

Abstract. In this paper a method of creating layers of feed-forward neural network that does not need to be learned is presented. Described approach is based on algorithms used in synthesis of logic circuits. Experimental results presented in the paper prove that this method may significantly decrease the time of learning process, increase generalization ability and decrease a probability of sticking in a local minimum. Further work and goals to achieve are also discussed.

Keywords: feed-forward neural network, logical circuit, parallel decomposition, argument reduction, pattern recognition, learning algorithm.

1 Introduction

Feed-forward neural network is a widely used tool in the area of artificial intelligence. It is mainly used in various applications of pattern recognition and prediction. However, feed-forward neural network has to be learned before it is used, and the learning process seems to be the most important step during the development of the network. There are few essential problems which can be met during the learning process. Most of algorithms use optimization methods to adjust weights of neurons - the network acts a parameterized function. The function used in the optimization process - most often based on mean square error – has a large number of local minimums and regions with very low slope. Therefore the algorithms can stuck in such local minima, and the learning process has to be restarted. Moreover, the larger the network is, the more neurons it contains, and with each single neuron n weights are associated, where n is the number of neurons in the previous layer. So, the number of weights grows much faster than the number of neurons and the learning algorithm has to perform significantly more calculations. On the other hand the structure of the neural network is also very important, because if there are too many neurons in the layer, the generalization abilities may decrease.

Since the neural network may also act as a logic circuit, functional decomposition based methods can be used to solve the problems mentioned above. The

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 564–573, 2007.

methods similar to those used in logic synthesis have already been used in knowledge discovery [1]. The first attempts in application of functional decomposition to neural networks were presented in [2,3] and then in [4]. In this approach the serial decomposition was used to shatter a large neural network into a couple of small networks connected with each other. The parallel decomposition and argument reduction were used to decrease the initial number of inputs. In [4] it has been shown that this approach has some major disadvantages – the generalization ability may be lower, or may decrease faster with the increase of noise in data. Another disadvantage is that this method can be used if the neural network is created for the binary or just quantized input and output data. It is useless for data that consists of real numbers.

In [4], another application of functional decomposition in feed-forward neural networks has been mentioned, but it was neither explored, nor experimentally verified. In this paper we attempt to fill this gap. In the presented approach, actually the created neurons are already learned, and the parallel decomposition or argument reduction is used to reduce number of them to the necessary minimum. The experimental results show very interesting capabilities of the method, in particular the learning time has been reduced for up to 30 times, without loss of the generalization abilities.

In Section 2 we present some basic notions. Then in Section 3 the considered approach is described. Section 4 presents experimental results. Finally conclusions are presented in Section 5.

2 Basic Notions

2.1 Feed-Forward Neural Network

A typical feed-forward neural network consists of few layers. Each layer contains a number of neurons which are not connected to any other neuron in the same layer, but each one is connected to all neurons in the previous and next layers. The input connector has a parameter, called weight, which is multiplied by an input signal value and the result is passed to the neuron. The neuron sums up all the signals and passes the result to an activation function (linear, binary, sigmoid or other). Fig. 1 shows the structure of a feed-forward neural network [5,6,7].

For example, the output signal for the neuron j in the layer b can be calculated as follows:

$$f(n(jb)) = f\left(B_{jb} + \sum_{i=1}^{k_a} W_{ia-jb} \cdot f(n(ia))\right) \tag{1}$$

where W_{x-y} is the weight assigned to a connection between neurons x and y, k_a is the number of neurons in the layer a, B is the parameter called bias (sometimes it is treated as a weight connected to the virtual fixed input signal equal to 1), and $f(x)$ is an activation function.

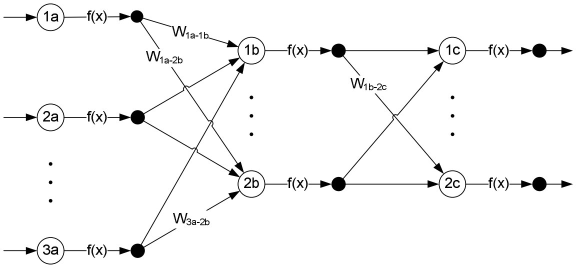


Fig. 1. A sample feed-forward neural network with 3 layers

In this paper the binary and sigmoid activation functions will be used.

$$f_{binary}(x) \begin{cases} 0, x < 0 \\ 1, x \geq 0 \end{cases} \tag{2}$$

$$f_{sigmoid}(x) = \frac{1}{1 + e^{-ax}} \tag{3}$$

Let us note that formula (1) represents a hyper-plane that splits the hyperspace of the input signals into two parts. Especially when the binary activation function is used, the output value will be equal to 0 for the data points of the first part, and it will be equal to 1 for the data points of the other part. So, the aim of learning is to find such hyper-planes that can split the data points, for which the output signal is different. Fig. 2 shows a sample set of the data points for which the output signal is 0 (the black points) and 1 (the white points). The lines n1 - n7 show the partitions made by the corresponding neurons.

There are several learning algorithms that are based on calculating derivative of some error function in respect to all parameters. Usually, the most frequently used error function is the mean square error (MSE), calculated for the output of

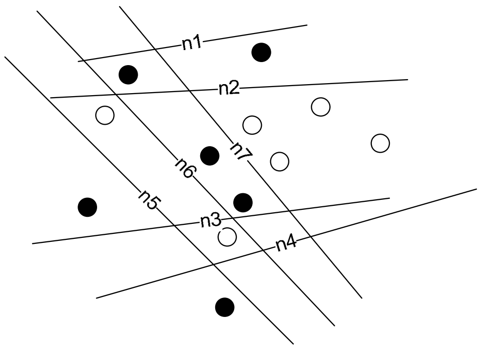


Fig. 2. A sample set of the data points

the neural network. Such algorithms require that the activation function is differentiable, therefore the usage of the binary functions for this case is impossible, and instead the sigmoid function is used [5,6,7]. In the presented approach we create few leading layers that contain neurons which have their weights already set, thus we do not need to apply the learning algorithm to them. Thus, in these layers we use the binary activation function. However end layers constitute a normal neural network, which has to be learned in a typical way, so we use the sigmoid activation function there.

2.2 Parallel Decomposition and Argument Reduction

In order to reduce a complexity of neural networks we create a number of leading layers with an extra number of neurons. Then we have to reduce them to the necessary minimum. To do this, a parallel decomposition and an argument reduction are used. These methods are commonly used in the logic synthesis of Boolean functions and we briefly present them below [8,9,10].

A problem occurs, when we want to implement a large Boolean function using components with a limited number of outputs. Note that such a parallel decomposition can also alleviate the problem of an excessive number of inputs of the function. This is because for typical functions most outputs do not depend on all input variables.

As an example let us consider a multiple-output function F (Table 1). Assume that F has to be decomposed into two components, G and H, with disjoint sets Y_G and Y_H of the output variables. Let us note that the set X_G of the input variables, on which the output variables from Y_G depend, may be smaller than X. Similarly, for the set X_H of input variables on which the outputs from Y_H depend, may be smaller than X. As a result, the components G and H have not only less outputs, but also less inputs than F. The exact formulation of the parallel decomposition problem depends on the constraints imposed by the implementation style. One possibility is to find the sets Y_G and Y_H, such that

Table 1. Function F

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	y_1	y_2	y_3	y_4	y_5	y_6
1	0	0	0	1	1	1	0	0	0	0	0	0	0	–	0
2	1	0	1	0	0	0	0	0	0	0	0	–	1	0	1
3	1	0	1	1	1	0	0	0	0	0	1	1	0	1	1
4	1	1	1	1	0	1	0	0	0	0	1	1	1	1	0
5	1	0	1	0	1	0	0	0	0	0	0	0	–	0	1
6	0	0	1	1	1	0	0	0	0	1	1	0	1	0	0
7	1	1	1	0	0	0	0	0	0	1	0	–	0	1	0
8	1	0	1	1	0	1	0	0	0	1	1	0	0	–	1
9	1	0	1	1	0	1	1	0	0	–	1	0	1	–	1
10	1	1	1	0	0	0	0	1	0	1	0	1	0	1	–
11	0	0	0	1	1	1	0	0	1	0	0	1	0	–	1
12	0	0	0	1	1	0	0	0	1	–	–	1	0	0	0

$card(X_G + card(X_H$ is minimal. Partitioning the set of outputs into only two disjoint subsets is not important here, because the procedure can be applied iteratively for the resulting components G and H.

The minimal sets of input variables, on which each output of F depends, are:

$y_1 : \{x_1, x_2, x_6\}$
$y_2 : \{x_3, x_4\}$
$y_3 : \{x_1, x_2, x_4, x_5, x_9\}, \{x_1, x_2, x_4, x_6, x_9\}$
$y_4 : \{x_1, x_2, x_3, x_4, x_7\}$
$y_5 : \{x_1, x_2, x_4\}$
$y_6 : \{x_1, x_2, x_6, x_9\}$

An optimal two-block decomposition, minimizing $card(X_G)+card(X_H)$, is $Y_G = \{y_2, y_4, y_5\}$ and $Y_H = \{y_1, y_3, y_6\}$, with $X_G = \{x_1, x_2, x_3, x_4, x_7\}$ and $X_H = \{x_1, x_2, x_4, x_6, x_9\}$. The truth tables for the components G and $H$are shown in the tables 2 and 3.

Table 2. Function G of parallel decomposition

	X_1	x_2	x_3	x_4	x_7	y_2	y_4	y_5
1	0	0	0	1	0	0	1	0
2	1	0	1	0	0	0	1	0
3	1	0	1	1	0	1	0	1
4	1	1	1	1	0	1	1	1
5	0	0	1	1	0	1	1	0
6	1	1	1	0	0	0	0	1
7	1	0	1	1	1	1	1	–

Table 3. Function H of parallel decomposition

	x_1	x_2	x_4	x_6	x_9	y_1	y_3	y_6
1	0	0	1	1	0	0	0	0
2	1	0	0	0	0	0	0	1
3	1	0	1	0	0	0	1	1
4	1	1	1	1	0	0	1	0
5	0	0	1	0	0	1	0	0
6	1	1	0	0	0	1	1	0
7	1	0	1	1	0	1	0	1
8	0	0	1	1	1	0	1	1
9	0	0	1	0	1	-	1	0

The algorithm itself is general in the sense that the function to be parallel decomposed can be specified in a compact cube notation. The calculation of the minimal sets of input variables for each individual output can be a complex task. Thus in the practical implementations, heuristic algorithms are used which support calculations with the help of the so called indiscernible variables. But the simplest way to obtain the quasi-minimal set of input variables for an output of a Boolean function is just trying to eliminate an input variable and then checking the consistency of the function - if the function is still consistent, the input variable can be safely removed.

3 Application of the Decomposition to Reduce the Neural Network

The idea is based on the fact that each neuron represents a hyper-plane. We can manually create neuron by neuron, so that each represents a hyper-plane splitting the data space. However it is very difficult to decide how to split the space, as there is unlimited number of solutions, the more that we decide it only for the training set without knowing the whole space. On the other hand we should build the network with possibly least number of neurons in order to achieve the network with the sufficient abilities to generalize. To this end, our algorithm should create a surplus set of hyper-planes and then to reduce it using a parallel decomposition and an argument reduction. In that way we create a layer with the reduced set of neurons. This step is repeated till the cardinality of the unique output signals of the layer is lower than of the previous layer. Finally, for the last layer we create a typical neural network to process the output data of the last but one layer. The steps are described in the following subsections in more details.

3.1 Creating Initial Set of Neurons

As stated above, at the beginning we devise an initial set of neurons, which may be superfluous. Let us assume that there are k data points in some n dimensional hyper-space. Each point has to be mapped to one of m possible output signals. The hyper-planes have to split the training data points that are mapped to the different output signals. A naive approach would be based on calculating the centroid of the data points for which the output signal is the same. Then a hyper-plane would be created between two randomly selected centroids, and the set of data points is shattered into two parts. The algorithm is repeated for each part till there is only a single centroid in each set. We have tested this approach, but it turned out to be inefficient and in some cases inconsistent.

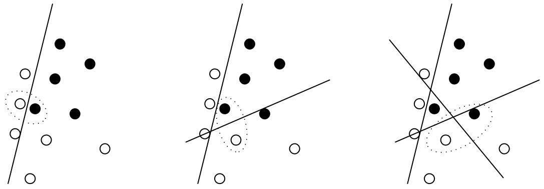


Fig. 3. Steps of the algorithm that creates hyper-planes

We base our approach on the observation that by splitting the closest distinguishable points we achieve the network with higher ability to distinct input data. Therefore we select such data points from varying groups that are closest to one other and create a hyper-plane between them. We repeat the process till the groups are homogenous. The algorithm is illustrated on Fig. 3.

3.2 Reducing the Set of Neurons

Each hyper-plane splits a hyper-space into two parts. Let us assume that for each data point, the value 0 is assigned if the data point is located in the first

Table 4. A layer of neurons presented as the truth table

Point number	Assignments for each hyper-plane					Output signal
	neuron 1	neuron 2	neuron 3	...	neuron n	
1	1	1	1	...	0	0 ...
2	0	0	1	...	0	0 ...
3	0	0	1	...	1	0 ...
...	...		...	...	...	... ...
k	0	1	0	...	1	1 ...

part of the hyper-space and the value 1 if one is located in the other part. This corresponds to assigning the value of the binary activation function (Table 4).

A logic function (similar to the one shown in Table 1) is created, so an argument reduction can be applied to it. The argument reduction removes unnecessary input arguments so the corresponding hyper-planes can be safely removed. Thus we will have a quasi-minimal set of neurons in a layer. We can also use a parallel decomposition and create the set with the minimum number of neurons for each output. Then we may group these sets to get the best results. We have performed a number of experiments, which have shown that the argument reduction used for the hyper-planes created by the algorithm sketched in p. 3.1 reduces the number of the neurons dozens times.

3.3 Normal Neural Network as the Output

The presented algorithms are not perfect. Although they significantly reduce the number of dimensions in the hyper-space for the layers, there is rather a slim chance that the cardinality of different output signals is equal to the cardinality of unique output signals of the system. To this end, we can (1) map the output signals of the last layer to the desired output signals or (2) create a simple neural network which can process them. We have selected the second solution for our experiments.

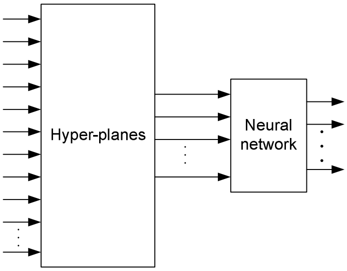


Fig. 4. A model of reduced neural network

4 Experimental Results

The experiments have been performed with the RPROP [6] learning algorithm for two cases: (1) a normal neural network and (2) a reduced neural network. The learning process was stopped when the error measure had fallen behind 0.01. If the algorithm had stuck in a local minimum, the learning process was started again, and a time was reset. In the experiments the neural network simulator and the decomposition tool created by authors were used. The measured variables were:

- a total learning time, i.e. the time of creating the hyper-planes and reducing the neural network (in case of testing the reduced network) + a time of learning),
- a learning result (a part of learning vectors that were correctly classified),
- a test result (a part of testing vectors that were correctly classified) and
- a number of local minima (an average number of events that the learning algorithm got stuck in a local minimum).

Each test was repeated for 5 times and an average value was calculated for each variable.

4.1 Comparison of Learning Time, Generalization Ability

In the first experiment 10 inputs and 4 outputs neural network was used. It analyzed the input vector, translating it to a code. The number of learning vectors was 492. We have divided the initial network into 4 parts for both normal and reduced neural network tests, so that each output was evaluated independently (one part per single output).

The second experiment was performed for the pattern classifier. Input data were matrices of 36 characters, each character in 3 variants – so the total number of learning vectors was equal to 108. There were 360 inputs and 36 outputs – each output is activated for the single character.

Table 5. The results of learning and testing the ones counter

Neural network	Total learning time	Learning result	Test result	Number of local minimums
Normal	38.6	0.9455	0.5779	3.6
Reduced	23.7	0.9565	0.6335	1.0

Table 6. The results of learning and testing the pattern classifier

Neural network	Total learning time	Learning result	Test result	Number of local minimums
Normal	33,9	0.9889	0.7630	0.0
Reduced	0,9	1.0000	0.7778	0.0

The results mentioned above show that the learning time was significantly shortened and the generalization abilities were slightly increased for the reduced network. Moreover, Table 5 shows that for the normal neural network the learning process stuck in the local minimum some 3 times more than for the reduced one.

4.2 Comparison of Abilities to Generalize for Different Noise Level in the Test Data

In the third experiment an input data were matrices of 93 characters, each character in 3 variants – so the total number of learning vectors was 279. The size of the input matrix was 1200. 7 outputs were binary representation of the character code. The tests were performed for 4 sets of the test vectors with a different noise level. The initial neural network was divided into 7 networks - each network for the single output.

Table 7 shows that the learning time was over 30 times shorter in the case of the reduced neural network. Furthermore the generalization abilities were better

Table 7. The learning time comparison

Neural network	Total learning time
Normal	345.5
Reduced	10.4

Table 8. The generalization ability for the different noise levels

Neural network	Test result - Generalization ability			
	Noise 25	Noise 50	Noise 100	Noise 200
Normal	0.515	0.389	0.278	0.121
Reduced	0.722	0.647	0.449	0.371

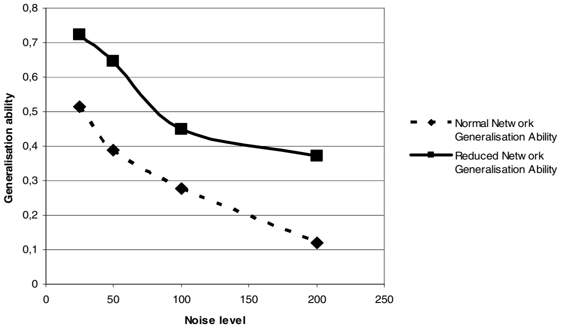


Fig. 5. The abilities to generalize for the different noise levels

for each examined noise level (Table 8). Fig. 5 depicts that they decrease slower for the reduced neural network than for the normal one.

5 Conclusions

The presented method allows simplifying a learning process of a feed-forward neural network. It is useful especially to solve various pattern recognition problems. The experimental results presented in the paper prove the effectiveness and efficiency of the proposed method. The time of learning, as well as, the probability of sticking in a local minimum were significantly reduced. The method allows also increasing the generalization abilities. The technique we put forward can be improved in the future works. The goal is to develop an algorithm for creating hyper-planes such that it would eliminate the need of using a normal neural network in the last layer.

References

1. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishing, Dordrecht (1991)
2. Niewiadomski, H.: Serial decomposition of feedforward neural networks. Warsaw University of Technology. M. Sc. Thesis (2003)
3. Lewandowski, J., Rawski, M., Migacz, M., Rybiński, H.: Analysis of decomposition of feed-forward neural network and it's impact on ability to generalization. Multimedia and network information systems. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, pp. 269–278 (2006)
4. Lewandowski, J.: Dekompozycja w sieciach neuronowych. Warsaw University of Technology, M. Sc Thesis (2006)
5. Haykin, S.: Neural Networks. A Comprehensive Foundation, 2nd edn. Pearson Education, Inc. Delhi, India (2005)
6. Osowski, S.: Sieci neuronowe do przetwarzania informacji. Oficyna Wydawnicza Politechniki Warszawskiej. Warszawa (2000)
7. Kasabov, N.K.: Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering. Massachusetts Institute of Technology (1996)
8. Łuba, T., Selvaraj, H., Nowicka, M., Kraśniewski, A.: Balanced multilevel decomposition and its applications in FPGA-based synthesis. In: Saucier, G., Mingnotte, A. (eds.) Logic and Architecture Synthesis, Chapman & Hall, Sydney (1995)
9. Łuba, T., Lasocki, R., Rybnik, J.: An Implementation of Decomposition Algorithm and its Application in Information Systems Analysis and Logic Synthesis. In: Ziarko, W. (ed.) Rough Sets, Fuzzy Sets and Knowledge Discovery. Workshops in Computing Series, pp. 458–465. Springer, Heidelberg (1994)
10. Brzozowski, J.A., Łuba, T.: Decomposition of Boolean Functions Specified by Cubes. Journal of Multi-Valued Logic & Soft. Computing 9, 377–417 (2003)

Combining Answers of Sub-classifiers in the Bagging-Feature Ensembles

Jerzy Stefanowski

Institute of Computing Sciences, Poznań University of Technology,
ul. Piotrowo 2, 60–965 Poznań, Poland
Jerzy.Stefanowski@cs.put.poznan.pl

Abstract. Improving classification performance of learning systems can be achieved by constructing multiple classifiers which include sets of sub-classifiers, whose individual predictions are combined to classify new objects. The diversification of sub-classifiers is one of necessary conditions for improving the classification accuracy. To obtain more diverse sub-classifiers we extend the bagging approach by integrating sampling different distributions of learning examples with selecting multiple subsets of features. We summarize results of our experiments on studying the usefulness of different feature selection techniques in this extension. The main aim of the paper is to examine the use of three methods for aggregating predictions of sub-classifiers in the extended bagging classifier. Our experimental results show that the extended classifier, with a dynamic choice of answers instead of a simple voting aggregation rule, is more accurate than standard bagging.

Keywords: machine learning, multiple classifiers, bagging, feature selection, aggregation rules.

1 Introduction

In the last decade it has been observed a growing research interest in increasing classification accuracy of supervised learning systems by *integrating* several classifiers into combined systems [4,9,19]. Such systems are known under the names *multiple classifiers, ensembles methods* or *committees.* It has been showed that these classifiers were often more accurate than the component classifiers that make them up and could be used for difficult tasks, where standard single classifiers failed. More motivations for creating such systems are presented, e.g. in [4,9]. Some methodological studies say that combining identical classifiers is useless [9]. The *diversification* of these sub-classifiers is treated as a necessary condition for their efficient combination.

Several methods have been proposed to get *diverse* base classifiers inside multiple classifiers. In general two kinds of approaches are distinguished [4,9,19]. In the first approaches, the same learning algorithm is used over *different samples* of the input data set, while in the other approaches different learning algorithms are applied to the same data set. In this paper we focus our attention on the

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 574–583, 2007.

bagging approach [2], which is based on using *bootstrap sampling* to get several different distributions of input examples and learning diversified classifiers from these samples. We have chosen it because it is quite efficient, we have good previous experience with combing it with rule induction [15], and its architecture could be easier extended than boosting or more advanced classifiers. Moreover, we are interested in *feature ensembles*, where different learning sets are obtained by selecting multiple, different subsets of features in the input data [12].

It should be noticed, however, that these two diversification approaches are considered independently in the current literature. So, one can ask a question about integrating together bootstrap sampling and selection of multiple feature subsets. The main motivation behind this idea is to make the ensemble more diverse than using any of these approaches alone. Such research have already been undertaken. In [10] Latinne et al proposed to combine the standard bagging with random selection of feature subsets over each bootstrap sample. They showed that this combination outperformed each of the single approach. The author, together with Kaczmarek, have also studied in [16] this kind of an integration, i.e. the bagging was extended in the other way by integrating bootstrap sampling with more advanced methods of feature selection, which are based on evaluation of the relationship between single feature, or feature subsets, and the target class. This paper was focusing on choosing the most appropriate feature selection methods. The following paper will summarize the main of these results and compare them to evaluations of new extensions of the bagging.

However, the other aspect while constructing multiple classifiers is to choose a proper technique for aggregating predictions of sub-classifiers. The standard bagging uses a majority vote rule. In other multiple classifiers more sophisticated techniques, as e.g. meta-learning from predictions of sub-classifiers, are also applied. Moreover, let us stress the observation made by other researchers, see e.g. [17], saying that if some sub-classifiers are more accurate in some subspaces of the input data but may be inaccurate on the rest of data, then it could be beneficial to promote their decisions for these objects which they are better specialized for. We think that this issue could be considered in the proposed extended bagging as the component sub-classifiers could be more diversified and may refer to different aspects of learning data.

Therefore, the main aim of this paper is to experimentally verify the usefulness of new methods for integrating the answers of sub-classifiers in our proposed enhancement of bagging. In particular, we focus attention on dynamic selection of classifiers and meta-learning methods and compare them against previous approaches on a diverse collection of benchmark data sets [1]. To be consistent with experiments from [10,16] all configurations of bagging sub-classifiers are *decision trees* induced by the Ross J. Quinlan's $C4.8$ algorithm.

2 Feature Selection and Feature Ensembles

The feature subset selection is a well studied problem in machine learning, data mining or statistics [11,8]. Typically, this problem is referred to a single learning algorithm and aims at finding a subset of features leading to not worse classification accuracy than the set of all features. The selection of features is done according a chosen *evaluation measure*. Some measures evaluate a degree of relationship between values of a single feature and a decision class and allow to rank features. Other measures and search strategies are more appropriate for evaluating subsets of features.

However, within the context of multiple classifiers the motivation is different. Feature subset selection is used as a mechanism for introducing the *diversity of base classifiers*. According to it, the learning sets for creating sub-classifiers are obtained by using different subsets of features for each of them (with maintaining the presence of the same learning examples in each learning set) [12]. The subsets of features are either overlapping or disjoint.

In some problems the features are naturally grouped, for example in signal processing (speaker identification or image recognition). The author also met such a grouping in e-mail machine classification, where separate features concerned the text content of the email body, the next features were coming from the email subject, others were extracted from the header or the attachments.

In general, natural grouping features may not occur and their multiple subsets should be automatically found. Reviews of such approaches are given in [19,9]. Quite well known is *Random Subspace Method* introduced by Ho [6], which consists of training several classifiers from input data sets constructed with a given proportion k of features picked randomly from the original set of features - the author of this method suggested in his experiment to select around 50% of the original set of features. There are also other "non-random" methods, where the correlation between each feature and the output class is computed and the base classifier is trained only on the most correlated subset of features. Other "favorite class" feature selection iterative methods have been considered by Puuronen at el. [13] - they also derived special contextual merit measures instead of using the simple correlation. Kuncheva also described in [9] the use of genetic algorithms to guide the random search in the space of possible feature subsets with intensifying a chosen diversity measure in a population of feature subsets. However, several experimental studies of different methods led to conclusions that there is no one best method for all situations [19].

3 Integrating Bagging with Feature Selection

Let us start from reminding principles of the bagging approach, introduced by Breiman [2]. It is based on an idea of running the same learning algorithm several times to get diversified classifiers, each time using a different distribution of the learning examples. Diversified learning sets are obtained from the input data set by *bootstraping sampling* with replacement. Each sample has the same size as the original set, however, some examples do not appear in it, while others may appear more than once. For a learning set with m examples, the probability of an example being selected at least once is $1-(1-1/m)^m$. For a large m, this is about 1 - $1/e$. According to Breiman [2] each bootstrap sample contains $\approx 63.2\%$ unique examples from the original set. Let $\{T_1, \ldots, T_s\}$ be a set of bootstrap samples. From each sample T_i a classifier C_i is induced by the same learning algorithm and the final classifier C^* is formed by aggregating s classifiers. A final classification of object x is built by an *equal voting scheme* on $C_1, C_2, \ldots, C_s$, i.e. the object is assigned to the class predicted most often by these sub-classifiers.

Let us shortly discuss the previous, related approaches to integrate bootstrap sampling in the bagging with the selection of subsets of features. According to our best knowledge, this idea was introduced by Latinne et al [10]. In this proposal, first, S bootstrap samples T_i of the learning set are generated with the same sampling schema as in [2]. Then, for each sample independently select R different subsets of features are randomly selected. The proportion of features K selected from the original sets is a parameter of the approach (details of tuning this value - the same for all subsets are given in [10]). Proceeding in this way, $S \cdot R$ learning sets are obtained to which the same learning algorithm is applied. In [10] decision tree were induced as base classifiers. The experimental results with this approach, called further $BagFs$, showed that it performed better than both diversification approaches used alone.

The author and Kaczmarek introduced in [16] another version of this enhanced approach, where subsets of features were selected according to more complex methods than plain random choice only. For each bootstrap sample, R random feature selection iterations were replaced by R different selection methods. The choice of these methods was common for all S bootstraps and each of them was conducted according to another evaluation measure. Sub-classifiers were trained on more classification relevant subsets of features and we hypothesized that as a result of choosing different methods these feature subsets could also be sufficiently diversified.

In [16] we experimentally studied the problem of choosing such methods and performed experiments on the same data sets as used in the following paper. In all experiments the sub-classifiers were decision trees induced by the C4.8 algorithm available in WEKA [20] - to be consistent with earlier works [10]. We used standard options of this algorithm. We tested 8 feature selection methods, which were either available in "filter" options of WEKA or had to be implemented. Due to the size of this paper we skip all detailed results (see e.g. [16]) and summarize that finally we propose to use the following evaluation measures:

- *Contextual-merit measure*: Proposed in [7] evaluates single features not their subset. It assigns the highest merit to features, where examples from different classes have different values.
- *Info-Gain* : The well known measure based on the information entropy often used in symbolic induction.
- *Chi-Squared statistic*: It is based on widely used statistics to evaluate pairs of features. Any numeric feature have to be discretized [20].
- *Correlation-based measure*: The idea behind it is that a good subset should contain features highly correlated with the class but uncorrelated with each other; For its definition see [5].

Table 1. A comparison of classification accuracy for standard classifier and bagging variants (an average value with a standard deviation represented in %)

Data	C4.8	Bag_{49}	Bag_7Fs_7	$Bag_{10}DFS_5$
glass	67.76±1.44	74.77±1.62	77.01±1.80	76.87±2.2
bupa	65.42±1.21	73.62±0.85	71.91±1.81	70.32±1.64
vote	94.23±0.65	94.80±0.28	94.83±0.39	94.97±0.11
breast	94.48±0.62	96.25±0.39	94.56±0.75	95.99±0.38
election	90.56±0.66	91.22±0.76	92.23±0.66	91.73±0.48
wine	93.82±1.18	96.07±0.88	95.00±1.44	96.69±1.04
ecoli	83.10±1.04	84.38±0.80	84.61±0.74	83.99±1.2
german	69.22±1.30	74.14±0.88	73.65±1.12	74.43 ±0.98

As the last method we considered the Random Subspace Method [6], because it is based on an absolutely different principle than above methods. Nearly all these methods evaluate the single features and the choice of features is done according to their ranking - which requires using the threshold parameter k for the best features (details of tuning it are given in [16]).

Below we summarize our experience with this extension. First, we had to decide about the number of sub-classifiers inside bagging. Previous experimental studies, e.g. [10,14,16], showed that choosing a high number (e.g. up to 343) did not led to much better accuracy while significantly increased computational costs. Following these observations we chose a configuration with around 49 component sub-classifiers - also to be consistent with previous results from [10]. Thus, our enhanced bagging was started with 10 bootstrap samples and, then, for each of these samples 5 iterations of chosen feature subsets selection methods were applied - this version is denoted as $Bag_{10}DFS_5$. We also tested more versions of this integration changing the number of bootstraps and feature selection loops but it seemed that next increasing the number of different feature selection less influenced the final classification than increasing the number of bootstraps. As we wanted to study another way of extending bagging with feature selection, we also created a classifier Bag_7Fs_7, where for each of 7 bootstraps 7 multiple feature subsets were randomly chosen by plain drawing (in the similar way as proposed in [10]). Moreover, to extend a comparison of bagging variants we added a standard bagging classifier built with 49 bootstraps (denoted as Bag_{49}). In all bagging configurations the equal weight voting was used as an aggregation method. Finally, we also evaluate a classification performance of the standard single decision tree induced by the same implementation of the C4.8 algorithm. Classification accuracy of all compared classifiers are presented in Table 1. The classification accuracy was estimated by the 10-fold stratified cross validation. All results were presented in tables as average values with standard deviations. We used 8 following data sets: *glass, bupa, vote, breast cancer Wisconsin, bush-election, wine, ecoli, german.* All the data sets, except *election*, were coming from UCI repository [1].

We also decided to examine other possible configurations of $Bag - DFS$ approach including a smaller number iterations of feature selection. First, we studied 5 new configurations of enhanced bagging, where for each of 12 bootstraps 4 different feature selection iterations were considered (i.e. one of the previous methods was temporary skipped). We also checked next configurations with 3 feature selection iterations for each of 16 bootstraps. Due to the page limits we skip detailed results (see [16]) and remark that the variant with feature selection methods based on Context-merit, Correlation based measures and Random selection was the best – its performance is given in Table 2 in column 2 as $Bag_{16}DFS_3 + EV$. In the additional experiments presented in this study we will also use the *wrapper approach* [8], where the search algorithm conducts a *forward stepwise search* for a subset of features using an induction of classifier inside an internal cross-validation to evaluate their classification accuracy [14].

4 Aggregating Answers of Sub-classifiers

In general, there are two kinds of methods for aggregating predictions of sub-classifiers: *group combination* or *specialized selection*. In the first method all classifiers are consulted to classify a new object while the other method chooses only these classifiers whose are "expertised" for this object.

Voting is the most common method used to combine predictions of single sub-classifiers. The classification prediction of each base classifier is considered as an equally weighted vote for the particular class. The class that receives the highest number of votes is selected as a final decision. The vote of each classifier may be *weighted*, e.g., by estimating its accuracy. There are also more advanced aggregation rules, e.g. using Bayesian rule or fuzzy operators - see [19,9].

Yet another idea includes *explicitly training a combination rule* - usually a *second level meta-learning classifier* is put on the outputs of base classifiers and has to merge these predictions into the final decision of the system. It is based on the idea of *meta-learning*, which is loosely defined as "learning from learned knowledge" [3]. Shortly speaking predictions made by the base classifiers on a set of extra validation examples, together with correct decision classes, form a basis for a meta-level training set and the learning algorithm is used to create the meta-classifier.

Few specialized *selection* methods have also been proposed, for review see e.g. [17]. In case of bagging or feature ensembles the *dynamic integration* methods, called *Dynamic Selection, Dynamic Voting* were considered as useful ones[17]. They are based on estimating local accuracy for sub-classifiers. Having learning data, the classification accuracy of respective sub-classifiers on each learning example is calculated, e.g. by cross-validation, and stored. When a new example is provided for classification, first its k nearest neighbors (examples) are found in the learning set using a distance metric based on its feature values. Then, the classification accuracies of all the sub-classifiers on this neighbors' set are found out in the previously stored estimates. Using these estimates one can establish which sub-classifiers are better performing on previous learning examples the

Table 2. Comparing aggregation techniques: equal weight voting, stacked combination vs. dynamic voting comparison

Data	$Bag_{16}DFS_3$	$Bag_{16}DFS_3$	$Bag_{16}DFS_3$	$BagFS$
set	$+EV$	$+DV$	$+SC$	$+DV$
glass	76.54±1.9	76.87±1.87	68.71±2.33	76.26±1.18
bupa	70.9±1.13	71.39±1.0	66.81±1.41	71.74±2.04
vote	95.0±0.1	95.0±0.1	94.40±0.16	94.77±0.64
breast	96.07±0.36	96.18±0.22	95.26±0.44	96.44±0.34
election	91.98±0.75	92.50±0.53	90.95±0.7	92.39±0.52
wine	97.08±0.96	97.08±1.02	93.31±1.28	96.74±0.37
ecoli	83.80±0.89	84.16±0.91	80.77±1.46	83.51±0.43
german	74.58±0.59	74.79±0.61	71.97±1.2	73.29±1.08

most similar to new coming object. In *Dynamic Voting* all of the sub-classifiers are used in a weighted voting, each with a weight proportional to its estimated accuracy. *Dynamic Selection* chooses the subset of classifiers with the highest classification accuracy to produce the final decision. According to [17] the above methods led to a better accuracy than the simple Equal Weight Voting for both bagging and boosting classifiers.

5 Experiments

The aim of these experiments is to evaluate the impact of applying different methods of integrating sub-classifiers answers on the classification accuracy of the proposed enhanced approach integrating different feature subset selection methods with bootstrap sampling within bagging framework. The following aggregation methods are verified:

- Simple Equal Weight Voting,
- Stacked Meta-Combiner - which was implemented as a decision tree induced by C4.5 algorithm.
- Dynamic Voting.

In dynamic voting we compute nearest learning examples of the classified object with an Euclidean distance measure for numeric features and Cost-Salzberg Value Difference Metric for symbolic ones. We tested several variants of the parameter "a number" of nearest neighbors and stayed with 21 - what was similar to the parameter used in [17]. In carried out experiments we applied the best found variant of our approach, i.e. $Bag_{16}DFS_3$, and extended it by using either Dynamic Voting method or Stacked Combiner for aggregation of sub-classifiers answers. To have more extensive comparison we also used it for the bagging with only random feature selection iterations denoted as $BagFs + DV$. The results are given in Table 2 are presented in the same way as before.

Moreover we studied the possibility of introducing the wrapper approach as an extra feature selection methods. We skipped it in earlier experiments with looking for the best extended variant [16] because it was the most computationally demanding [14]. We chose the $Bag_{16}DFS_3 + DV$ variant, which led to the best improvement of classification accuracy and considered its two new possible configurations containing the use of wrapper, i.e.: $Bag_{12}DFS_3 + wrapper$ (where for each of 12 bootstrap the 4 feature subsets were selected by methods using Correlation-based, Contextual Merit measures, Plain Random Drawing and the wrapper approach and $Bag_{16}DFS_2 + wrapper$ (where the wrapper was added as an additional feature selection to Correlation-based measure and Plain Random Drawing for each of 16 samples) Classification results for all these bagging variants are presented in Table 3.

Table 3. Classification accuracy of different bagging variants

Data set	Bag_7Fs_7	$Bag_{16}DFS_3$	$Bag_{12}DFS_3$ + wrapper	$Bag_{16}DFS_2$ + wrapper
glass	77.01±1.80	76.87±1.87	77.11±1.15	77.53±1.82
bupa	71.91±1.81	71.39±1.0	72.23±1.72	72.46±1.9
vote	94.83±0.39	95.00±0.11	95.35±0.2	94.97±0.1
breast	94.56±0.75	96.18±0.22	96.28±0.29	96.69±0.26
election	92.23±0.66	92.50±0.53	92.66±0.37	92.73±0.48
wine	95.00±1.44	97.08±1.02	97.64±0.66	97.36±0.71
ecoli	84.61±0.74	84.16±0.91	84.94±0.82	85.39±1.02
german	73.65±1.12	74.79±0.61	74.36±0.73	74.86±0.96

6 Discussion of Experimental Results and Final Remarks

Improving classification performance of classifiers induced by data mining methods is still one of the key problems in knowledge discovery and machine learning. One of the solutions is to integrate base classifiers into combined systems. In this paper we discussed different approaches for extending the bagging classifier, which besides boosting is one of the most popular multiple classifier. Following the postulate of the diversification of answers from the ensemble the main idea was to integrate bootstrap sampling with various feature selection methods which should produce a larger number of more diverse component classifiers. The main contribution of this paper has been to study usefulness of different methods for integrating answers of these sub-classifiers inside bagging instead of simple voting combination rule, which is typically applied in the bagging.

Firstly, let us summarize the results of the experimental evaluation of these enhancements starting from verification of various feature selection methods. The first observation is that all versions of the extended bagging classifier are significantly better than a single, standard classifier (in the sense of t - Student paired statistical test). Moreover, for nearly all data they are competitive comparing to the standard version of the bagging classifier (Bag_{49}). The exception is *bupa* data set, where Bag_{49} is still the best.

The best version of the extended bagging classifier proposed in this paper, called $Bag_{16}DFS_3 + DV$, is significantly better than the variant with using only random selections of multiple feature subsets (Bag_7Fs_7) on 3 out of 8 data sets (precisely *breast, wine* and *german*) while for other data sets observed improvements were not significant in the sense of paired t test. This classifier consisted of 16 bootstrap samples duplicated 3 times, each time using a different feature selection method - Correlation-based measure, Contextual Merit measure and Plain Random drawing. One should also notice that the new $Bag_{16}DFS_3 + DV$ was generally better for data containing a higher number of examples, while insignificant differences occurred for data with a smaller number of examples.

Introducing the wrapper method inside enhanced bagging slightly increased the classification accuracy – see Table 3.

Let us now discuss results of evaluating different methods of integrating subclassifiers answers on the classification accuracy. Implementing the dynamic voting to combine answers of base classifiers led to better results for the $Bag_{16}DFS_3$ classifier, while having less influence on the plain random $BagFs$ classifier. However, no progress was noticed for the other aggregation technique based on incorporating stacked generalization with an additional decision tree. Perhaps other meta-learning algorithm, as e.g. naive Bayes, could be chosen. It also seems that the dynamic voting worked slightly better when the wrapper method has been applied as an additional feature selection method - see Table 3. Here, we can also remark that in the most related research [17] the dynamic voting, applied directly to feature selection, also improved the classification performance of bagging. However, we should conclude from our experiments that observed increases of the classification accuracy around 1-2 % are not too impressive. It seems that the considered aggregation methods has less influenced the final classification accuracy than an earlier integration of bagging with selections of multiple feature subsets.

Finally, we should pay attention to computational requirements - in particular if we think about mining larger data. Although the proposed extended classifier is more accurate, we also observed the growth of computational costs comparing it to the traditional approach. For instance, in our experiments single C4.8 classifier was built on the *glass* data set in 1.5 second, Bag_{49} in 26 seconds, Bag_7Fs_7 in 27 seconds, $Bag_{16}DFS_3$ in 234 seconds and with wrapper even more. Thus, if the time restrictions are important, the simple random feature selection could be an acceptable alternative to more advanced approaches.

References

1. Blake, C., Koegh, E., Mertz, C.J.: Repository of Machine Learning, University of California at Irvine (1999)
2. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
3. Chan, P.K., Stolfo, S.J.: A comparative evaluation of voting and meta-learning on partitioned data. In: Proceedings of the 12th International Conference on Machine Learning, San Francisco, pp. 90–98 (1995)

4. Dietrich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
5. Hall, M.: Correlation-based feature selection for discrete and numeric class machine learning. In: Proc. 17th Conf. on Machine Learning (2000)
6. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
7. Hong, S.J.: Use of contextual information for feature ranking and discretization. IEEE Transactions on Knowledge and Data. Engineering 9, 718–730 (1997)
8. Kohavi, R., Sommerfield, D.: Feature subset selection using the wrapper method: overfitting and dynamic search space topology. In: Proceedings of the 1st Int. Conference on Knowledge Discovery and Data Mining, Montreal, pp. 192–197. AAAI Press, Stanford, California (1995)
9. Kuncheva, L.: Combining Pattern Classifiers: Methods and Algorithms. Wiley, Chichester (2004)
10. Latinne, P., Debeir, O., Decaestecker, Ch.: Mixing bagging and multiple feature subsets to improve classification accuracy of decision tree combination. In: Proc. of the 10th Belgian-Dutch Conf. on Machine Learning, Tilburg University (2000)
11. Liu, H., Motoda, H.: Feature Selection for Data Mining and Knowledge Discovery. Kluwer Academic Publishers, Dordrecht (1998)
12. Optiz, D.: Feature selection for ensembles. In: Proc. of the 16th National Conference on Artificial Intelligence, pp. 379–384. AAAI/MIT Press, Stanford, California (1999)
13. Puuronen, S., Skrypnyk, I., Tsymbal, A.: Ensemble feature selection based on contextual merit and correlation heuristics. In: Caplinskas, A., Eder, J. (eds.) ADBIS 2001. LNCS, vol. 2151, pp. 155–168. Springer, Heidelberg (2001)
14. Stefanowski, J.: An experimental evaluation of improving rule based classifiers with two approaches that change representations of learning examples. Engineering Applications of Artificial Intelligence Journal 17, 439–445 (2004)
15. Stefanowski, J.: The bagging and n2-classifiers based on rules induced by MODLEM. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 488–497. Springer, Heidelberg (2004)
16. Stefanowski, J., Kaczmarek, M.: Integrating attribute selection to improve accuracy of bagging classifiers. In: Proc. of the AI-METH, Conference - Recent Developments in Artificial Intelligence Methods, Gliwice, 2004, pp. 263–268 (2004)
17. Tsymbal, A., Puuronen, S.: Bagging and Boosting with dynamic integration of classifiers. In: Zighed, A.D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 116–125. Springer, Heidelberg (2000)
18. Tsymbal, A., Puuronen, S., Sktypnyk, I.: Ensemble feature selection with dynamic integration of classifiers. In: Proc. of Int. ICSC Congress on Computational Intelligence Methods and Applications, CIMA', Bangor, pp. 558–564 (2001)
19. Valentini, G., Masuli, F.: Ensambles of learning machines. In: Marinaro, M., Tagliaferri, R. (eds.) Neural Nets. LNCS, vol. 2486, pp. 3–19. Springer, Heidelberg (2002)
20. Weka, machine learning software in Java, University of Waikato, http://www.cs.waikato.ac.nz/ml/weka/index.html

Monotonic Behavior of Entropies and Co–entropies for Coverings with Respect to Different Quasi–orderings*

Daniela Bianucci and Gianpiero Cattaneo

Dipartimento di Informatica, Sistemistica e Comunicazione
Università di Milano – Bicocca
Via Bicocca degli Arcimboldi 8, I–20126 Milano, Italia
{bianucci, cattang}@disco.unimib.it

Abstract. Some quasi–ordering for coverings are here defined. Different definitions of entropies and co–entropies for coverings are examined, also focusing on the distinction between the here called *global* and *pointwise* approaches to co–entropies. The entropies and co–entropies are defined both for coverings generated from an information system via a similarity relation and for generic coverings of a universe X (i.e., coverings not necessarily generated from an information system via a similarity relation). The behavior of the entropies and co–entropies for coverings with respect to the defined quasi–orderings is thus explored.

1 Introduction

In this work we explore the behavior of entropies and co–entropies for coverings of a universe X with respect to some quasi-ordering relations, as extension of the similar concepts widely used in the case of partitions of the standard approach to information theory [Sha48], with possible applications to incomplete information systems. In order to facilitate the comprehension of some extensions, let us briefly introduce in this introductory section the main concepts and results in the stronger context of partitions of X, whose collection will be denoted by $\Pi(X)$ in the sequel.

First of all, let us recall that the original Pawlak approach to rough set theory is essentially based (see [Cat98]) on an *approximation space*, formalized as a pair $\langle X, \pi \rangle$, where X is a (finite) set of *objects* (the *universe* of the discourse) and $\pi = \{A_1, A_2, \ldots, A_N\}$ is a partition of X, generally induced by an indiscernibility equivalence relation in the context of an information system [Paw81], whose elements are the *elementary sets* (in the Pawlak terminology) or *elementary events* (in the probabilistic approach). In general, each elementary set A_i of the partition π is assumed to represents a *granule* of some knowledge supported by the approximation space. The σ–algebra $\mathcal{E}(X, \pi)$ generated by the partition

* The author's work has been supported by MIUR\PRIN project "Automata and Formal languages: mathematical and application driven studies".

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 584–593, 2007.

π consists of the empty set and the collection of all set theoretic unions of elementary events. A (measurable) set $E_j \in \mathcal{E}(X, \pi)$ is called a *definable set* in the context of rough set theory and *event* in the context of measurable spaces of probability theory.

For each granule A_i we can consider the counting measure $m(A_i) = |A_i|$, which is a *granularity measure* of the elementary event A_i. Besides this granularity measure, it is further on possible to consider the *probability measure* $p(A_i) = |A_i|/|X|$ of the elementary event A_i, which is related to the *uncertainty* of the occurrence of event A_i. For any partition $\pi \in \Pi(X)$ we thus can associate two "vectors": the *granularity distribution* $\boldsymbol{m}(\pi) = (m(A_1), m(A_2), \ldots, m(A_N))$ and the *probability distribution* $\boldsymbol{p}(\pi) = (p(A_1), p(A_2), \ldots, p(A_N))$.

From the probabilistic point of view, for any granule A_i of the partition π it is possible to introduce the non–negative real quantity $G(A_i) := \log m(A_i) \in \mathbb{R}_+$, interpreted as the *granularity measure* of A_i, obtaining in this way a *discrete random variable* $\boldsymbol{G}[\pi] = (\log m(A_1), \log m(A_2), \ldots, \log m(A_N))$, called the "granularity random variable" generated by the partition π. Similarly, for each event A_i we can consider the quantity in the unit real interval $I(A_i) := -\log p(A_i)$, which in information theory is known as the *information function* measuring the uncertainty related to the probability $p(A_i)$ of occurrence of the (elementary) event A_i. In this case, we have a discrete "uncertainty random variable" $\boldsymbol{I}[\pi] = (-\log p(A_1), -\log p(A_2), \ldots, -\log p(A_N))$. As usual in statistical theory, the *expectation value* of the granularity random variable $G[\pi]$ with respect to the probability distribution $\boldsymbol{p}(\pi)$ (also *granularity average* of π) is given by the quantity $Exp\,\langle \boldsymbol{G}[\pi], \boldsymbol{p}(\pi)\rangle = \sum_{i=1}^{N} G(A_i)\, p(A_i)$. This quantity in the sequel will be denoted simply by $E(\pi)$, and trivially $E(\pi) = \frac{1}{|X|}\sum_{i=1}^{N} |A_i| \log |A_i|$. Similarly the average of the information uncertainty random variable $\boldsymbol{I}[\pi]$ with respect to the probability distribution $\boldsymbol{p}(\pi)$ is given by $Exp\,\langle \boldsymbol{I}[\pi], \boldsymbol{p}(\pi)\rangle = \sum_{i=1}^{N} I(A_i)\, p(A_i)$ (the *uncertainty average* of π), usually called the *entropy* of the partition π, and simply denoted by $H(\pi)$. Also in this case we have the complete formula $H(\pi) = -\frac{1}{|X|}\sum_{i=1}^{N} |A_i| \log \frac{|A_i|}{|X|}$. It is easy to see that their sum, $H(\pi) + E(\pi) = \log |X|$, is invariant with respect to the choice of the partition, and this is the reason of the name "*co–entropy*" assigned to the quantity $E(\pi)$ in order to underline that it *complements* the entropy $H(\pi)$ with respect to the constant quantity $\log |X|$. Let us stress that the semantic of these two quantities associated to any partition π of the universe X must not be confused: the co–entropy $E(\pi)$ furnishes a measure of the *average granularity* of a partition π, whereas the entropy $H(\pi)$ measures the *average uncertainty* assigned to this partition.

In the collection $\Pi(X)$ of all the partitions of X, one can introduce the partial order relation $\preceq$, for which we say that a partition π_1 (resp., π_2) is *finer* (resp., *coarser*) than a partition π_2 (resp., π_1), written $\pi_1 \preceq \pi_2$, iff $\forall\, A_i \in \pi_1$, $\exists B_j \in \pi_2$ such that $A_i \subseteq B_j$. The strict partial ordering is as usual defined as $\pi_1 \prec \pi_2$ iff $\pi_1 \preceq \pi_2$ and $\pi_1 \neq \pi_2$ (this means that there must exist at least a situation for which $A_i \subset B_j$). The set $\Pi(X)$ with respect to this partial ordering $\preceq$ is a lattice, such that the meet $\pi_1 \wedge \pi_2$ of two partitions $\pi_1 = \{A_i; i = 1, 2, \ldots, R\}$ and $\pi_2 = \{B_1, B_2, \ldots, B_S\}$ is given by the collection of all the nonempty intersections

$C_{i,j} = A_i \cap B_j \neq \emptyset$. An important result about co–entropy is its strict monotonic behavior: let $\pi_1 \prec \pi_2$, then $E(\pi_1) < E(\pi_2)$ (see for instance [BCC07]). From this result it follows the strict anti–monotonicity of the standard information entropy: let $\pi_1 \prec \pi_2$, then $H(\pi_2) < H(\pi_1)$.

In order to bridge the gap with respect to some generalizations of co–entropy which can be found in literature (see for instance [LX00]), we consider now a partition π from a *pointwise* point of view, i.e., for each point $x \in X$ we will take into account the granule $gr(x)$ which contains this object (or, more properly, the granule generated by x through the chosen equivalence relation). If the finite universe is $X = \{x_1, x_2, \ldots, x_M\}$, the partition π will be thus described by the *pointwise* collection $\pi_p := \{gr(x_1),\, gr(x_2),\, \ldots,\, gr(x_M)\}$. For this *pointwise* collection, we can define a *pointwise* co–entropy as: $E_{LX}(\pi) = \frac{1}{|X|}\sum_{x\in X} |gr(x)| \log |gr(x)|$ [BCC07]. This pointwise co–entropy behaves monotonically with respect to the ordering $\preceq$ of partitions (see [BCC07] for more details). Similarly, we can define the *pointwise* entropy $H_{LX}(\pi) = -\sum_{x\in X} \frac{|gr(x)|}{|X|} \log \frac{|gr(x)|}{|X|}$, which unfortunately behaves neither monotonically, nor anti–monotonically with respect to $\preceq$.

Let us stress that two other binary relations can be introduced on $\Pi(X)$ according to the following definitions: $\pi_1 \ll \pi_2$ iff $\forall B_j \in \pi_2$, $\exists \{A_{j_1}, A_{j_2}, \ldots, A_{j_p}\} \subseteq \pi_1$: $B_j = A_{j_1} \cup A_{j_2} \cup \ldots \cup A_{j_p}$ and $\pi_1 \trianglelefteq \pi_2$ iff $\forall x \in X$, $gr_{\pi_1}(x) \subseteq gr_{\pi_2}(x)$. In [BCC07] it has been recalled that these three partial order relations $\preceq$, $\ll$ and $\trianglelefteq$, are equivalent in the partition context, i.e., they define the same partial order relation on $\Pi(X)$, but it has been stressed that in the covering context they become three different quasi–orderings. Hence, in the transition from the partitioning context to the covering one, the main objectives are: **(Gen1)** to generalize the single ordering for partitions (described in three equivalent ways) in three distinct quasi–orderings for coverings; **(Gen2)** to generalize the definition of co–entropy for partition to the covering case (firstly via the global co–entropies and then by the pointwise approach, the former one computationally more economic than the latter); **(Gen3)** explore whether or not the monotonicity of co–entropies defined in the covering context is preserved. The final aim of the present work is to provide a complete answer to the question "which are the co–entropies for coverings that behave monotonically with respect to the quasi–orderings $\preceq$, $\ll$, and $\trianglelefteq$?" The results discussed here (and completing some partial results proved in [BCC07]) are that unfortunately any considered global co–entropy for coverings do not behave monotonically, differently from the pointwise approaches, which all behave monotonically with respect to the three above introduced quasi–orderings. The cases of monotonicity will be formally proven; when neither a monotonic nor an anti–monotonic behavior can be proven, counterexamples will be illustrated.

1.1 Co–entropies and Entropies for Coverings

The so–called *global* co–entropies defined for coverings as the more natural generalizations of the co–entropy for partitions, do not succeed in preserving the above discussed monotonicity. On the contrary, it is the *pointwise* approach to co–entropies for coverings (introduced for the first time by [LX00], and further on extended in more general way in [BCC07]) which preserves the desired monotonicity. The aim of this work is to extensively cover the treatment of quasi orderings and co–entropies in the covering case, also completing some partial results appeared in [BCC07]. Let us recall that the definition of a generic covering γ of a universe X, can be described as a collection of *nonempty* subsets of X such that $\bigcup_{i=1}^{N} B_i = X$. The collection of all coverings of the universe X will be denoted by $\Gamma(X)$.

From a generic covering we can construct or extract other coverings, following determinate criteria. In [BCC07] we have introduced the definition of *genuine* covering formalized as a covering $\gamma = \{B_1, B_2, \ldots, B_N\}$ for which the following condition is satisfied: $\forall B_i \in \gamma,\ \forall B_j \in \gamma,\ B_i \subseteq B_j \quad \text{implies} \quad B_i = B_j$. In the sequel, we will denote by $\Gamma_g(X)$ the class of all genuine coverings of X.

2 Quasi–ordering for Coverings

In [BCC07] some quasi–orderings (i.e., reflexive and transitive, but in general non anti–symmetric relations [Bir67, p. 20]) for coverings has been introduced. The first quasi–ordering is given by the following binary relation:

$$\text{Let } \gamma, \delta \in \Gamma(X), \text{ then } \gamma \preceq \delta \quad \text{iff} \quad \forall\, C_i \in \gamma,\ \exists D_j \in \delta : C_i \subseteq D_j \tag{1}$$

In this case, we will say that γ is *finer* than δ or that δ is *coarser* than γ. The corresponding *strict* quasi–order relation is $\gamma \prec \delta$ iff $\gamma \preceq \delta$ and $\gamma \neq \delta$. As remarked in [BCC07], in the class of genuine coverings $\Gamma_g(X)$ the quasi–ordering relation $\preceq$ results to be an ordering. Another quasi–ordering on $\Gamma(X)$ different from (1) and with no general relationship with it (in the covering context), is defined by the following binary relation:

$$\gamma \ll \delta \quad \text{iff} \quad \forall D \in \delta,\ \exists \{C_1, C_2, \ldots, C_p\} \subseteq \gamma : D = C_1 \cup C_2 \cup \ldots \cup C_p \tag{2}$$

After the introduction of these two quasi–orderings for coverings, let us consider a covering γ and its partition $\pi(\gamma)$ obtained according to the procedure described in the previous section, then both $\pi(\gamma) \ll \gamma$ and $\pi(\gamma) \preceq \gamma$ hold.

In [BCC07] we have also introduced two possible kinds of similarity classes induced for an object x of the universe X by a covering γ of X: the *upper granule* $\gamma_u(x) = \cup\{C \in \gamma : x \in C\}$ and the *lower granule* $\gamma_l(x) := \cap\{C \in \gamma : x \in C\}$ generated by x. Thus, for any $x \in X$ we can define the *granular rough approximation of x induced by γ* as the pair $r_\gamma(x) := \langle \gamma_l(x), \gamma_u(x) \rangle$, where $\gamma_l(x) \subseteq \gamma_u(x)$. The collections $\gamma_u := \{\gamma_u(x) : x \in X\}$ and $\gamma_l := \{\gamma_l(x) : x \in X\}$ of all such granules are both coverings of X, called the *upper covering* and the *lower covering* generated by γ. It is easy to prove that the following hold:

$$\gamma_l \preceq \gamma \preceq \gamma_u \qquad \text{and} \qquad \gamma_l \ll \gamma \ll \gamma_u \tag{3}$$

We can now introduce three more quasi–ordering relations on $\Gamma(X)$ defined by the following three binary relations:

$$\gamma \trianglelefteq_u \delta \quad \text{iff} \quad \forall x \in X,\ \gamma_u(x) \subseteq \delta_u(x); \qquad \gamma \trianglelefteq_l \delta \quad \text{iff} \quad \forall x \in X,\ \gamma_l(x) \subseteq \delta_l(x);$$

$$\gamma \trianglelefteq \delta \quad \text{iff} \quad \gamma \trianglelefteq_l \delta \quad \text{and} \quad \gamma \trianglelefteq_u \delta. \tag{4}$$

We have that $\gamma \preceq \delta$ implies $\gamma \trianglelefteq_u \delta$, but in general $\gamma \trianglelefteq_l \delta$ does not hold. It is also trivial to show that for every covering γ, one has $\gamma_l \trianglelefteq \gamma \trianglelefteq \gamma_u$. Let us denote by $\Gamma_l(X) := \{\gamma_l : \gamma \in \Gamma(X)\}$ the family of all lower coverings and by $\Gamma_u(X) := \{\gamma_u : \gamma \in \Gamma(X)\}$ the family of all upper coverings of X induced by γ. Then we have that the pair $r(\gamma) := \langle \gamma_l, \gamma_u \rangle$ is the *rough approximation of the covering* γ with respect to all quasi–orderings $\preceq$, $\ll$, and $\trianglelefteq$ in the *rough approximation space* (see [Cat98, CC04]) $\langle \Gamma(X), \Gamma_l(X), \Gamma_u(X) \rangle$ consisting of the collection $\Gamma(X)$ of all *approximable coverings*, and the collections $\Gamma_l(X)$ (resp., $\Gamma_u(X)$) of *lower* (resp., *upper*) *definable* coverings.

We can thus introduce another quasi–ordering on the family $\Gamma(X)$ of all coverings of X as

$$\gamma \Subset \delta \quad \text{iff} \quad \delta \trianglelefteq_l \gamma \ \text{ and } \ \gamma \trianglelefteq_u \delta. \tag{5}$$

In particular we have that, for this defined quasi–ordering, the following property holds:

$$\gamma \Subset \delta \quad \text{implies} \quad \forall x \in X, \ \ \delta_l(x) \subseteq \gamma_l(x) \subseteq (???) \subseteq \gamma_u(x) \subseteq \delta_u(x) \tag{6}$$

where the question marks represent an intermediate covering granule $\gamma(x)$, which is something of “hidden” in the involved structure. This “local” behavior can be equivalently described as

$$\forall x \in X, \ \ r_\gamma(x) := \langle \gamma_l(x), \gamma_u(x) \rangle \sqsubseteq \langle \delta_l(x), \delta_u(x) \rangle =: r_\delta(x)$$

where $\sqsubseteq$ means that, according to (6), for any point $x \in X$ the local approximation $r_\gamma(x)$ given by the covering γ is better than the local approximation $r_\delta(x)$ given by the covering δ. We will conventionally denote this fact simply by $r(\gamma) \sqsubseteq r(\delta)$, so that (6) can be summarized by

$$\gamma \Subset \delta \quad \text{implies} \quad r(\gamma) \sqsubseteq r(\delta)$$

2.1 Entropies and Co–entropies for Coverings of a Universe

In [BCC07] we proposed a first approach to define a global co–entropy and entropy for coverings, which, in short, consists of the following steps:

(1) let us consider the characteristic functional $\chi_{B_i} : X \mapsto \{0,1\}$ of the set B_i defined for any point $x \in X$ as $\chi_{B_i}(x) = 1$ if $x \in B_i$ and $= 0$ otherwise;
(2) $\forall x \in X$, let us define $n(x) := \sum_{i=1}^{N} \chi_{B_i}(x)$;
(3) $\forall x \in X$, let us define $\omega_{B_i}(x) := \frac{1}{n(x)} \chi_{B_i}(x)$;
(4) the measure of the generic set B_i of the covering γ is defined as $m(B_i) := \sum_{x \in X} \omega_{B_i}(x) = \sum_{x \in X} \frac{1}{n(x)} \chi_{B_i}(x)$;
(5) the probability of occurrence of B_i is $p(B_i) := \frac{1}{|X|} m(B_i)$;
(6) lastly, we can define the global entropy for coverings as
$H(\gamma) = -\sum_{i=1}^{N} p(B_i) \log p(B_i)$ and the global co–entropy as
$E(\gamma) = \frac{1}{|X|} \sum_{i=1}^{N} m(B_i) \log m(B_i)$.

Let us note that one of the main characteristics of this approach is that $\forall x \in X, \sum_{i=1}^{N} \omega_{B_i}(x) = 1$, and consequently, that $\sum_{i=1}^{N} m(B_i) = |X|$. This leading to the fact that $\boldsymbol{p}(\gamma) = (p(B_1), p(B_2), \ldots, p(B_N))$ defines a probability distribution, that is a reason for which we first considered this approach, together with the fact that $\forall \gamma \in \Gamma(X), \quad H(\gamma) + E(\gamma) = \log |X|$. An important drawback of this co–entropy is that it could assume negative values, as shown in example 4.1 of [BCC07]. Also if this is not considered as a drawback, even in the restricted case of genuine coverings this co–entropy does not behave monotonically with respect to all the quasi–orderings $\preceq$, $\ll$, and $\trianglelefteq_j$ $(j = l,\, u)$ as shown in the next example.

Example 1. In the universe $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$, let us consider the two *genuine* coverings $\gamma = \{C_1 = \{1, 4, 5\}, C_2 = \{2, 4, 5\}, C_3 = \{3, 4, 5\}, C_4 = \{14, 15\}, C_5 = \{4, 5, \ldots, 13\}\}$ and $\delta_1 = \{D_1 = \{1, 4, 5\} = C_1, D_2 = \{2, 4, 5\} = C_2, D_3 = \{3, 4, \ldots, 13, 14\} = C_3 \cup C_5, D_4 = \{4, 5, \ldots, 14, 15\} = C_4 \cup C_5, \}$. Trivially, $\gamma \preceq \delta_1$ and $\gamma \ll \delta_1$. In this case we obtain $E(\gamma) = 2.05838 < 2.18897 = E(\delta_1)$, as expected.

In the same universe, let us now take the *genuine* covering $\delta_2 = \{F_1 = \{1, 4, 5, \ldots, 12, 13\} = C_1 \cup C_5, F_2 = \{2, 4, 5, \ldots, 12, 13\} = C_2 \cup C_5, F_3 = \{3, 4, \ldots, 12, 13\} = C_3 \cup C_5, F_4 = \{4, 5, \ldots, 14, 15\} = C_4 \cup C_5\}$. Trivially, $\gamma \preceq \delta_2$ and $\gamma \ll \delta_2$, but, contrary to the previous case, we here obtain an anti–monotonic result, with $E(\gamma) = 2.05838 > 1.91613 = E(\delta_2)$.

On the other hand, in the universe $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, let us consider the two *genuine* coverings $\gamma_1 = \{\{1, 3\}, \{2, 3, 4, 5\}, \{10, 11, 12\}, \{3, 4, \ldots, 11\}\}$ and $\delta_1 = \{\{1, 3, 4, \ldots, 10, 11\}, \{2, 3, \ldots, 10, 11\}, \{3, 4, \ldots, 11, 12\}\}$. Trivially, $\gamma_1 \preceq \delta_1$ and $\gamma_1 \ll \delta_1$. According to the procedure previously illustrated, we can construct the *lower* and *upper coverings* for γ_1 and δ_1, obtaining the lower coverings: $\gamma_{1_l} = \{\gamma_{1_l}(1) = \{1, 3\}, \gamma_{1_l}(2) = \{2, 3, 4, 5\}, \gamma_{1_l}(3) = \{3\}, \gamma_{1_l}(4) = \gamma_{1_l}(5) = \{3, 4, 5\}, \gamma_{1_l}(6) = \gamma_{1_l}(7) = \gamma_{1_l}(8) = \gamma_{1_l}(9) = \{3, 4, \ldots, 10, 11\}, \gamma_{1_l}(10) = \gamma_{1_l}(11) = \{10, 11\}, \gamma_{1_l}(12) = \{10, 11, 12\}\}$; $\delta_{1_l} = \{\delta_{1_l}(1) = \{1, 3, 4, \ldots, 10, 11\}, \delta_{1_l}(2) = \{2, 3, \ldots, 10, 11\}, \delta_{1_l}(3) = \delta_{1_l}(4) = \ldots = \delta_{1_l}(10) = \delta_{1_l}(11) = \{3, 4, \ldots, 10, 11\}, \delta_{1_l}(12) = \{3, 4, \ldots, 11, 12\}\}$; and the upper coverings: $\gamma_{1_u} = \{\gamma_{1_u}(1) = \{1, 3\}, \gamma_{1_u}(2) = \{2, 3, 4, 5\}, \gamma_{1_u}(3) = \{1, 2, \ldots, 10, 11\}, \gamma_{1_u}(4) = \gamma_{1_u}(5) = \{2, 3, \ldots, 10, 11\}, \gamma_{1_u}(6) = \gamma_{1_u}(7) = \gamma_{1_l}(8) = \gamma_{1_u}(9) = \{3, 4, \ldots, 10, 11\}, \gamma_{1_u}(10) = \gamma_{1_u}(11) = \{3, 4, \ldots, 11, 12\}, \gamma_{1_u}(12) = \{10, 11, 12\}\}$; $\delta_{1_u} = \{\delta_{1_u}(1) = \{1, 3, 4, \ldots, 10, 11\}, \delta_{1_u}(2) = \{2, 3, \ldots, 10, 11\}, \delta_{1_u}(3) = \delta_{1_u}(4) = \ldots = \delta_{1_u}(10) = \delta_{1_u}(11) = \{X\}, \delta_{1_u}(12) = \{2, 3, \ldots, 11, 12\}\}$. We can observe that $\gamma_1 \trianglelefteq_l \delta_1$ and $\gamma_1 \trianglelefteq_u \delta_1$, hence $\gamma_1 \trianglelefteq \delta_1$. The co–entropies are $E(\gamma_1) = 1.85592 < 2.00000 = E(\delta_1)$, $E(\gamma_{1_l}) = 1.32072 < 1.60097 = E(\delta_{1_l})$, and $E(\gamma_{1_u}) = 0.94928 < 1.59646 = E(\delta_{1_u})$. These results could suggest a monotonic behavior of this co–entropy with respect to $\preceq$, $\ll$ and $\trianglelefteq_j$.

We can see that unfortunately this does not correspond to a general behavior. In the same universe, let us consider the *genuine* covering $\gamma_2 = \{\{1, 3\}, \{2, 3, 4, 5\}, \{12\}, \{3, 4, \ldots, 11\}\}$. We have $\gamma_2 \preceq \delta_1$ and $\gamma_2 \ll \delta_1$. Let us construct the *lower* and *upper coverings* of γ_2: $\gamma_{2_l} = \{\gamma_{2_l}(1) = \{1, 3\}, \gamma_{2_l}(2) = \{2, 3\}, \gamma_{2_l}(3) = \{3\}, \gamma_{1_l}(4) = \gamma_{2_l}(5) = \ldots = \gamma_{2_l}(10) = \gamma_{2_l}(11) = \{3, 4, \ldots, 10, 11\}, \gamma_{2_l}(12) =$

$\{12\}\}$; $\gamma_{2_u} = \{\gamma_{2_u}(1) = \{1,3\}, \gamma_{2_u}(2) = \{2,3\}, \gamma_{2_u}(3) = \{1,2,\ldots,10,11\}$, $\gamma_{2_u}(4) = \gamma_{2_u}(5) = \ldots = \gamma_{2_u}(10) = \gamma_{2_u}(11) = \{3,4\ldots,10,11\}$, $\gamma_{2_u}(12) = \{12\}\}$. Again we have $\gamma_2 \trianglelefteq_l \delta_1$ and $\gamma_2 \trianglelefteq_u \delta_1$ and thus $\gamma_2 \trianglelefteq \delta_1$, but in this case the co–entropies are $E(\gamma_2) = 2.21646 > 2.00000 = E(\delta_1)$, $E(\gamma_{2_l}) = 2.11842 > 1.60097 = E(\delta_{1_l})$, and $E(\gamma_{2_u}) = 1.73407 > 1.59646 = E(\delta_{1_u})$.

For these reasons, in [BCC07], we tried a second approach defining an entropy and corresponding co–entropy, somehow inspired by the Liang and Xu approach introduced in [LX00]. In order to illustrate the definition of these global entropy and co–entropy, let us first define the *total outer measure* of γ as $m^*(\gamma) := \sum_{i=1}^{N} |B_i| \geq |X|$, and let us consider the probability $p_{LX}(B_i) = \frac{|B_i|}{|X|}$. The entropy $H_{LX}^{(g)}$ of the covering γ is defined as:

$$H_{LX}^{(g)}(\gamma) := -\sum_{i=1}^{N} p_{LX}(B_i) \log p_{LX}(B_i) = m^*(\gamma)\frac{\log|X|}{|X|} - \frac{1}{|X|}\sum_{i=1}^{N} |B_i| \log |B_i|$$

The corresponding co–entropy $E_{LX}^{(g)}(\gamma)$ is:

$$E_{LX}^{(g)}(\gamma) := \frac{1}{|X|}\sum_{i=1}^{N} |B_i| \cdot \log |B_i|. \tag{7}$$

Unfortunately, we found out that this co–entropy (7) shows neither a monotonic, nor an anti–monotonic general behavior with respect to the quasi–orderings $\preceq$, $\ll$, and $\trianglelefteq_j$, as it is illustrated in the following example.

Example 2. Making reference to example 1, let us consider the *genuine* coverings γ_2 and δ_1, recalling that $\gamma_2 \preceq \delta_1$, $\gamma_2 \ll \delta_1$ and that $\gamma_2 \trianglelefteq \delta_1$. According to (7) we obtain $E_{LX}^{(g)}(\gamma_2) = 2.71078 < 8.30482 = E_{LX}^{(g)}(\delta_1)$, $E_{LX}^{(g)}(\gamma_{2_l}) = 2.71078 < 10.68226 = E_{LX}^{(g)}(\delta_{1_l})$, and $E_{LX}^{(g)}(\gamma_{2_u}) = 5.88192 < 12.29265 = E_{LX}^{(g)}(\delta_{1_u})$, as desired.

In the same universe, let us now take the *genuine* coverings $\gamma_3 = \{\{1,3,4,5,6\}, \{2,3,4,5,6\}, \{5,6,7,\ldots,11,12\}, \{3,4,5,6,7,8\}\}$ and $\delta_2 = \{\{1,3,4,5,6\}, \{2,3,4,5,6\}, \{3,4,\ldots,11,12\}\}$. Trivially we can see that $\gamma_3 \preceq \delta_2$ and $\gamma_3 \ll \delta_2$. With little more effort, we can find out that in this case we also have $\gamma_3 \trianglelefteq \delta_2$. Unluckily in the present example we obtain $E_{LX}^{(g)}(\gamma_3) = 5.22742 > 4.70321 = E_{LX}^{(g)}(\delta_2)$, $E_{LX}^{(g)}(\gamma_{3_l}) = 5.43494 > 5.36988 = E_{LX}^{(g)}(\delta_{2_l})$, and $E_{LX}^{(g)}(\gamma_{3_u}) = 12.28818 > 8.28818 = E_{LX}^{(g)}(\delta_{2_u})$ which represents a behavior opposite to the one previously observed with γ_2 and δ_1.

The just considered two entropies and co–entropies are of *global type* in the sense that they involve global subsets of the investigated covering. Let us see now two other entropies and co–entropies inspired by the Liang and Xu approach [LX00], described also in [BCC07], whose definition is deeply *pointwise*. Given a covering γ of X, let us consider the similarity classes $\gamma_l(x)$ and $\gamma_u(x)$ generated by every element $x \in X$, and the resultant coverings γ_l and γ_u, previously described. We can define the following *pointwise* entropies:

$$H_{LX}(\gamma_j) := -\sum_{x \in X} \frac{|\gamma_j(x)|}{|X|} \log \frac{|\gamma_j(x)|}{|X|} \quad \text{for} \quad j = l, u \tag{8}$$

The corresponding *pointwise* co–entropies are defined as:

$$E_{LX}(\gamma_j) := \frac{1}{|X|} \sum_{x \in X} |\gamma_j(x)| \log |\gamma_j(x)| \quad \text{for} \quad j = l, u. \tag{9}$$

We have the following relationship between $H_{LX}(\gamma_j)$ and $E_{LX}(\gamma_j)$: $H_{LX}(\gamma_j) + E_{LX}(\gamma_j) = \frac{\log |X|}{|X|} \cdot \sum_{x \in X} |\gamma_j(x)|$. In [BCC07] we showed that the following property holds.

Proposition 1. *Let γ_1 and γ_2 be two coverings of X such that $\gamma_1 \trianglelefteq_j \gamma_2$, then we have that $E_{LX}(\gamma_{1j}) \leq E_{LX}(\gamma_{2j})$. In particular, with respect to the quasi–ordering (5) we have that*

$$\gamma_1 \Subset \gamma_2 \quad \textit{implies} \quad E_{LX}(\gamma_{2l}) \leq E_{LX}(\gamma_{1l}) \leq (???) \leq E_{LX}(\gamma_{1u}) \leq E_{LX}(\gamma_{2u}) \tag{10}$$

Moreover, in in the same work we also introduced the *rough co–entropy approximation* of the covering γ defining it as as the pair $r_E(\gamma) = \langle E_{LX}(\gamma_l), E_{LX}(\gamma_u)\rangle$, where we have the following: $0 \leq E_{LX}(\gamma_l) \leq E_{LX}(\gamma_u) \leq |X| \cdot \log |X|$. So, summarizing, we have that $\gamma_1 \Subset \gamma_2$ implies $r_E(\gamma_1) \sqsubseteq r_E(\gamma_2)$.

2.2 Co–entropy for Coverings Induced by Similarity Relation in Incomplete Information Systems

From the rough set theory point of view, we know that when we deal with an incomplete information system $\mathcal{IS} = \langle X, Att, F\rangle$ (with F only partially defined on $X \times Att$), for any family $\mathcal{A}$ of attributes we can define the similarity relation $\mathcal{S}_\mathcal{A}$ on X as

$$x\mathcal{S}_\mathcal{A}y \quad \text{iff} \quad \forall\, a \in \mathcal{A}, \text{either} \quad f_a(x) = f_a(y) \quad \text{or} \quad f_a(x) = * \quad \text{or} \quad f_a(y) = *. \tag{11}$$

The granules generated by this relation are denoted as *similarity classes* $s_\mathcal{A}(x) = \{y \in X : (x, y) \in \mathcal{S}_\mathcal{A}\}$. All the similarity classes induced by a similarity relation $\mathcal{S}_\mathcal{A}$ constitute a covering, here denoted by $\gamma(\mathcal{A}) := \{s_\mathcal{A}(x) : x \in X\}$; in fact it results that $x \in s_\mathcal{A}(x) \neq \emptyset$, and so a fortiori $X = \cup\{s_\mathcal{A}(x) : x \in X\}$. In the sequel we will indicate with $\Gamma(\mathcal{IS}) := \{\gamma(\mathcal{A}) \in \Gamma(X) : \mathcal{A} \subseteq Att\}$ the collection of the coverings generated by all the possible families of attributes $\mathcal{A}$ from Att. With respect to a covering $\gamma(\mathcal{A})$ induced by a similarity relation from an incomplete information system, Liang and Xu in [LX00] introduced an interesting approach to the here called co–entropy defining it as follows

$$E_{LX}(\gamma(\mathcal{A})) = \frac{1}{|X|} \sum_{x \in X} |s_\mathcal{A}(x)| \cdot \log |s_\mathcal{A}(x)| \tag{12}$$

This co–entropy behaves monotonically with respect to the quasi–ordering $\preceq$.

Proposition 2. *In an incomplete information system $\langle X, Att, F\rangle$ let us consider two families of attributes $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{B} \subseteq \mathcal{A}$. Let the induced coverings of X be $\gamma(\mathcal{A})$ and $\gamma(\mathcal{B})$. The following holds:*

$$\gamma(\mathcal{A}) \preceq \gamma(\mathcal{B}) \quad \textit{implies} \quad E_{LX}(\gamma(\mathcal{A})) \leq E_{LX}(\gamma(\mathcal{B})). \tag{13}$$

The proof of this property is similar to the proof of Theorem 1 in [LX00] (Proposition 6.2 in [BCC07]). Furthermore, the following holds for the co–entropy (12) generated in the context of incomplete information systems.

Proposition 3. *Given an incomplete information system $\langle X, Att, F\rangle$ and two similarity relations defined on the objects of X for the two families of attributes $\mathcal{A}$ and $\mathcal{B}$, let the induced coverings of X be respectively $\gamma(\mathcal{A})$ and $\gamma(\mathcal{B})$. We have that for any $\mathcal{A}, \mathcal{B} \subseteq Att$, with $\mathcal{B} \subseteq \mathcal{A}$:*

$$\gamma(\mathcal{A}) \ll \gamma(\mathcal{B}) \quad \textit{implies} \quad E_{LX}(\gamma(\mathcal{A})) \leq E_{LX}(\gamma(\mathcal{B})). \tag{14}$$

Moreover, if $\exists\ s_{\mathcal{B}}(x_k) = s_{\mathcal{A}}(x_{k_1}) \cup s_{\mathcal{A}}(x_{k_2}) \cup \ldots \cup s_{\mathcal{A}}(x_{k_p})$ such that $p \geq 2$, the strict monotonicity $E_{LX}(\gamma(\mathcal{A})) < E_{LX}(\gamma(\mathcal{B}))$ holds.

Proof. Let the two coverings be respectively $\gamma(\mathcal{A}) = \{s_{\mathcal{A}}(x_1), s_{\mathcal{A}}(x_2), \ldots, s_{\mathcal{A}}(x_N)\}$ and $\gamma(\mathcal{B}) = \{s_{\mathcal{B}}(x_1), s_{\mathcal{B}}(x_2), \ldots, s_{\mathcal{B}}(x_N)\}$. We have that $\gamma(\mathcal{A}) \ll \gamma(\mathcal{B})$, hence $\forall\, s_{\mathcal{B}}(x_i) \in \gamma(\mathcal{B})$, $\exists \{s_{\mathcal{A}}(x_{i_1}), s_{\mathcal{A}}(x_{i_2}), \ldots, s_{\mathcal{A}}(x_{i_p})\} \subseteq \gamma(\mathcal{A}) : s_{\mathcal{B}}(x_i) = s_{\mathcal{A}}(x_{i_1}) \cup s_{\mathcal{A}}(x_{i_2}) \cup \ldots \cup s_{\mathcal{A}}(x_{i_p})$. Let us consider the simple case in which $s_{\mathcal{B}}(x_k) = s_{\mathcal{A}}(x_{k_1}) \cup s_{\mathcal{A}}(x_{k_2}) \cup \ldots \cup s_{\mathcal{A}}(x_{k_p})$ with $p \geq 2$, and $s_{\mathcal{B}}(x_j) = s_{\mathcal{A}}(x_j)$ for any $j \neq k$. Then we have that:

$E_{LX}(\gamma(\mathcal{B})) = \frac{1}{|X|}\sum_{x_i \in X} |s_{\mathcal{B}}(x_i)| \cdot \log|s_{\mathcal{B}}(x_i)| = \frac{1}{|X|}\big(\sum_{x_j \in X,\, j \neq k} |s_{\mathcal{B}}(x_j)| \cdot \log|s_{\mathcal{B}}(x_j)| + |s_{\mathcal{B}}(x_k)| \cdot \log|s_{\mathcal{B}}(x_k)||\big) = \frac{1}{|X|}\big(\sum_{x_j \in X,\, j \neq k} |s_{\mathcal{A}}(x_j)| \cdot \log|s_{\mathcal{A}}(x_j)| + |s_{\mathcal{A}}(x_{k_1}) \cup \ldots \cup s_{\mathcal{A}}(x_{k_p})| \cdot \log|s_{\mathcal{A}}(x_{k_1}) \cup \ldots \cup s_{\mathcal{A}}(x_{k_p})||\big)$. Since $|s_{\mathcal{A}}(x_{k_1}) \cup \ldots \cup s_{\mathcal{A}}(x_{k_p})| \cdot \log|s_{\mathcal{A}}(x_{k_1}) \cup \ldots \cup s_{\mathcal{A}}(x_{k_p})| > |s_{\mathcal{A}}(x_k)| \log|s_{\mathcal{A}}(x_k)|$, we obtain that $E_{LX}(\gamma(\mathcal{A})) < E_{LX}(\gamma(\mathcal{B}))$. Hence in general we have that, given two coverings $\gamma(\mathcal{A})$ and $\gamma(\mathcal{B})$ such that $\gamma(\mathcal{A}) \ll \gamma(\mathcal{B})$, $E_{LX}(\gamma(\mathcal{A})) \leq E_{LX}(\gamma(\mathcal{B}))$. In particular, when $p \geq 2$ the strict monotonicity $E_{LX}(\gamma(\mathcal{A})) < E_{LX}(\gamma(\mathcal{B}))$ holds.

3 Conclusions

In this work we have explored different quasi–orderings on coverings and different definitions of entropies and co–entropies for coverings. We have analyzed the behavior of these co–entropies with respect to the defined quasi–orderings. Table 1 illustrates the compact answer "Yes/No" for the monotonicity of the co–entropies for coverings here analyzed.

We have shown that the here called *pointwise* co–entropies behave monotonically with respect to all the quasi–orderings $\preceq$, $\ll$, and $\trianglelefteq_j$. We have also shown that with respect to the quasi–ordering $\Subset$, the co–entropies $E_{LX}(\gamma_j)$ behave according to (10). On the contrary, for what concerns the *global* co–entropies $E(\gamma)$ and $E_{LX}^{(g)}(\gamma)$ we have observed that unfortunately they behave neither

Table 1. Behavior of co–entropies for coverings with respect to some quasi–orderings: “Yes” stands for monotonicity, “No” for non–monotonicity (nor anti–monotonicity)

	Global		Pointwise	
	$E(\gamma)$	$E^{(g)}_{LX}(\gamma)$	$E_{LX}(\gamma)$	$[E_{LX}(\gamma_j)]$
$\preceq$	No	No	Yes	
$\ll$	No	No	Yes	
$\unlhd_l$	No	No	Yes	
$\unlhd_u$	No	No	Yes	

monotonically nor anti–monotonically with respect to all the considered quasi–orderings $\preceq$, $\ll$, and $\unlhd_j$, even in the restricted case of genuineness of coverings. Hence, we may (for now) conclude that a co–entropy for coverings, in order to behave monotonically with respect to some (quasi) orderings, should be *pointwise* defined .

References

[BCC07] Bianucci, D., Cattaneo, G., Ciucci, D.: Entropies and co–entropies of coverings with application to incomplete information systems. Fundamenta Informaticae 75, 77–105 (2007)

[Bir67] Birkhoff, G.: Lattice theory, 3rd edn. American Mathematical Society, vol. XXV. American Mathematical Society Colloquium Publication, Providence, Rhode Island (1967)

[Cat98] Cattaneo, G.: Abstract approximation spaces for rough theories. In: Polkowski and Skowron, pp. 59–98 [PS98]

[CC04] Cattaneo, G., Ciucci, D.: Investigation about Time Monotonicity of Similarity and Preclusive Rough Approximations in Incomplete Information Systems. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 38–48. Springer, Heidelberg (2004)

[HHZ04] Huang, B., He, X., Zhong, X.: Rough entropy based on generalized rough sets covering reduction. Journal of Software 15, 215–220 (2004)

[LX00] Liang, J., Xu, Z.: Uncertainty measure of randomness of knowledge and rough sets in incomplete information systems. In: Proc. of the 3rd World Congress on Intelligent Control and Automata. Intelligent Control and Automata 4, 2526–2529 (2000)

[Paw81] Pawlak, Z.: Information systems - theoretical foundations. Information Systems 6, 205–218 (1981)

[PS98] Polkowski, L., Skowron, A. (eds.): Rough sets in knowledge discovery 1. Physica–Verlag, Heidelberg, New York (1998)

[Sha48] Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal 27, 379–423, 623–656 (1948)

[ZW03] Zhu, W., Wang, F.-Y.: Reduction and axiomization of covering generalized rough sets. Information Sciences 152(1), 217–230 (2003)

Design and Implementation of Rough Rules Generation from Logical Rules on FPGA Board

Akinori Kanasugi and Mitsuhiro Matsumoto

Department of Electronics, Tokyo Denki University
2-2, Nishikicho, Chiyoda-ku, Tokyo 101-8457, Japan
kanasugi@d.dendai.ac.jp,05gmd17@ed.cck.dendai.ac.jp
http://www.d.dendai.ac.jp/lab_site/vs/index_e.html

Abstract. In this paper, the design, simulation, implementation and experiment of rough set processor are described. The experiment result shows that the proposed processor is ten times faster than PC, though the clock frequency is about 70 times lower.

Keywords: Rough sets, Processor, FPGA.

1 Introduction

The rough set theory can discover profitable knowledge from the incomplete database [1][2]. The calculation of rough set is simple, but it is difficult to obtain quick responses by software tools on general computers. Then, the authors have been proposed the architecture of a special processor for rough set theory [3][4]. In this paper, the design, the implementation and the experiment of rough set processor on FPGA evaluation board are described.

2 Architecture

Fig. 1 shows the flow of knowledge discovery by rough set theory. In the present, the processor treats only the part where knowledge is discovered from a large-scale logical function. The reason is the processing of text data and the setting of the threshold values are difficult for hardware.

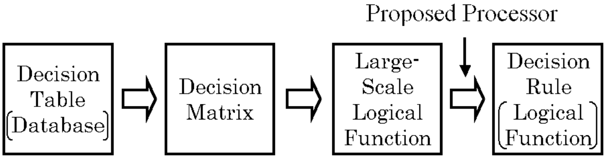

Proposed Processor
Decision Table (Database)
Decision Matrix
Large-Scale Logical Function
Decision Rule (Logical Function)

Fig. 1. Flow of knowledge discovery

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 594–602, 2007.

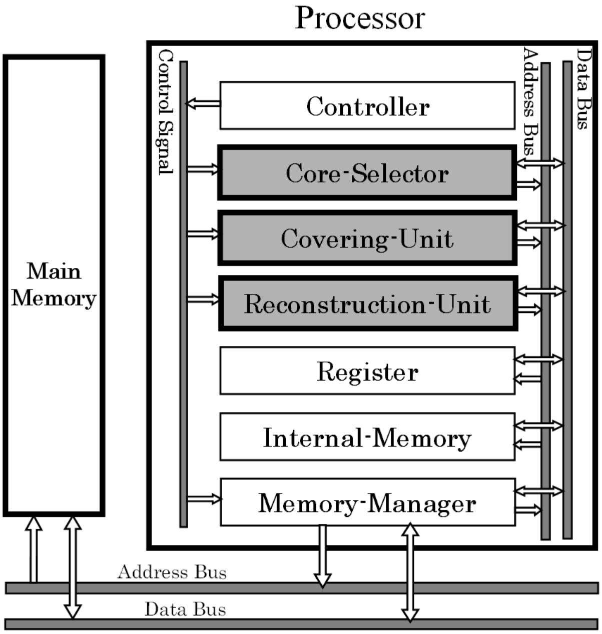


Fig. 2. Block diagram of proposed processor

$$
\begin{aligned}
F = \; & f\,(X_0, X_1, X_2, \ldots\ldots\ldots, X_{2030}, X_{2031})\\
= \; & (X_0 \vee 0 \vee X_2 \vee 0 \vee X_4 \vee \ldots\ldots\ldots \vee 0 \vee X_{2031})\\
\wedge \; & (X_0 \vee X_1 \vee 0 \vee X_3 \vee 0 \vee \ldots\ldots\ldots \vee X_{2030} \vee X_{2031})\\
\wedge \; & (0 \vee 0 \vee X_2 \vee 0 \vee X_4 \vee \ldots\ldots\ldots \vee X_{2030} \vee 0)\\
\wedge \; & (X_0 \vee X_1 \vee 0 \vee X_3 \vee X_4 \vee \ldots\ldots\ldots \vee 0 \vee X_{2031})\\
\wedge \; & (X_0 \vee 0 \vee X_2 \vee X_3 \vee 0 \vee \ldots\ldots\ldots \vee 0 \vee 0)\\
\wedge \; & (X_0 \vee 0 \vee X_2 \vee 0 \vee X_4 \vee \ldots\ldots\ldots \vee X_{2030} \vee X_{2031})\\
\wedge \; & (0 \vee X_1 \vee 0 \vee X_3 \vee 0 \vee \ldots\ldots\ldots \vee 0 \vee X_{2031})\\
\wedge \; & (\;\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \quad \vdots\;)\\
\wedge \; & (\;\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \quad \vdots\;)\\
\wedge \; & (\;\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \quad \vdots\;)\\
\wedge \; & (\;\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \quad \vdots\;)\\
\wedge \; & (\;\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \quad \vdots\;)\\
\wedge \; & (0 \vee X_1 \vee 0 \vee X_3 \vee 0 \vee \ldots\ldots\ldots \vee X_{2030} \vee 0)\\
\wedge \; & (X_0 \vee 0 \vee X_2 \vee 0 \vee X_4 \vee \ldots\ldots\ldots \vee X_{2030} \vee X_{2031})\\
\wedge \; & (X_0 \vee X_1 \vee 0 \vee X_3 \vee 0 \vee \ldots\ldots\ldots \vee 0 \vee X_{2031})
\end{aligned}
$$

1,000,000 rows

2,032 columns

Fig. 3. Input logical function

Fig. 2 and Fig. 3 show the block diagram and input logical function of the proposed processor, respectively. In Fig. 2, “Core-Selector” and “Covering-Unit” reduce the data in the pre-processing, and ”Reconstruction-Unit” extracts the rules in the post-processing.

2.1 Core Selector

"Core Selector" selects some core data and transfers to "Core Register". The row which contains a lot of '0' is chosen as a core. Fig. 4 shows the block diagram of Core Selector.

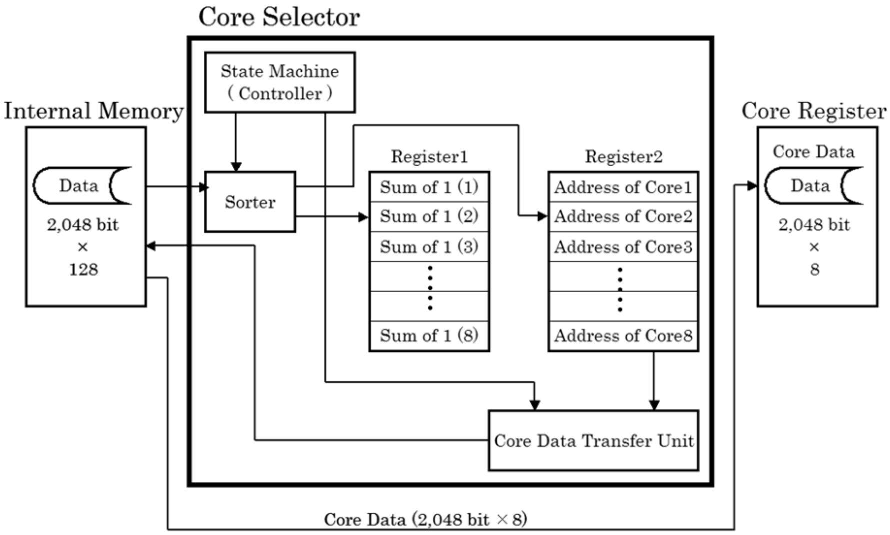


Fig. 4. Block diagram of Core Selector

2.2 Covering Unit

"Covering Unit" deletes the data that can be deleted by using the selected core data. By this pre-processing of "Core Selector" and "Covering Unit", the post-processing time of "Reconstruction Unit" is reduced.

The reduction processing by covering is concisely described as follows. In expression (1), the shortest term $(x_0 + x_2 + x_6)$ is chosen as a core.

$$F = (x_0 + x_2 + x_6) \wedge (x_0 + x_1 + x_2 + x_5 + x_6) \wedge (x_1 + x_2 + x_6 + x_7). \qquad (1)$$

The second term can be deleted, because it satisfies $(x_0 + x_2 + x_6) \subseteq (x_0 + x_1 + x_2 + x_5 + x_6)$. On the other hand, the third term can't be deleted, because $x_0 + x_2 + x_6) \not\subseteq (x_1 + x_2 + x_6 + x_7)$. That is, the condition "*core* $\subseteq$ *other term*" is checked in the covering unit.

Fig. 5 shows the block diagram of covering unit. Each input data (2,048 bit) is read from the internal memory, and the judgment processing is done to each data field as shown in this figure. This processing is done by 16 bit, and executed 127 times. The judgment processing is continued by using the following core data until there is an error in the result of the judgment processing. The error means the failure of covering.

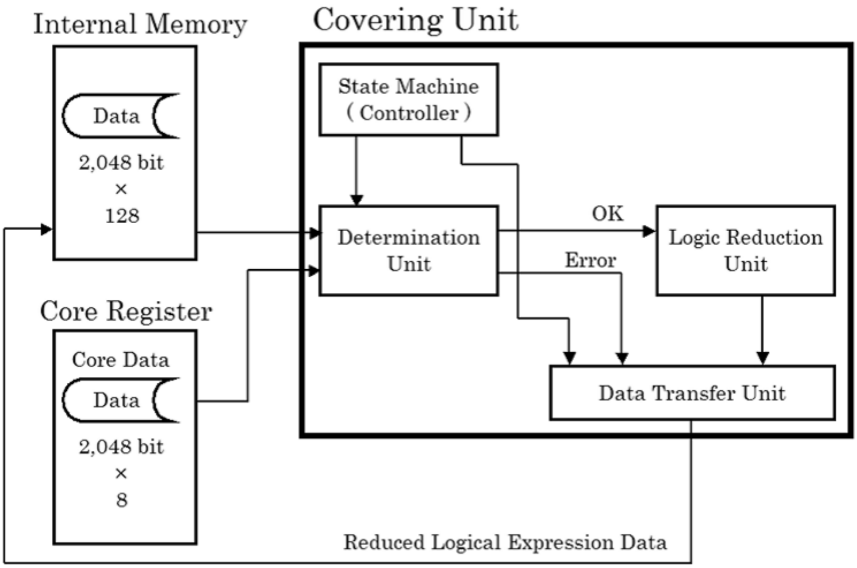


Fig. 5. Block diagram of Covering Unit

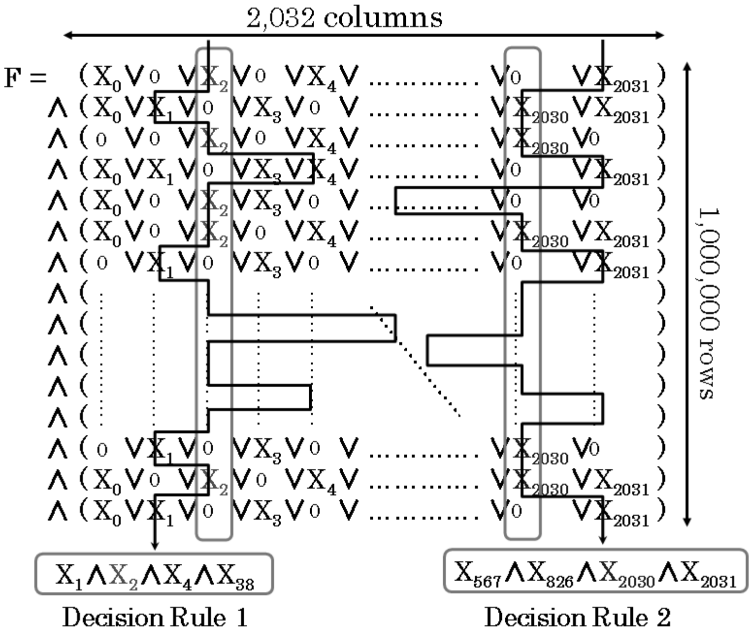

2,032 columns
F =
1,000,000 rows
$X_1 \wedge X_2 \wedge X_4 \wedge X_{38}$
Decision Rule 1
$X_{567} \wedge X_{826} \wedge X_{2030} \wedge X_{2031}$
Decision Rule 2

Fig. 6. Discovery of important rules

2.3 Reconstruction Unit

"Reconstruction Unit" discovers the most important rules from the reduced logical function. In this post-processing, the most important rule is the logical term that has a few variables. Fig. 6 shows the discovery of the most important rules.

The reconstruction unit consists of the parallel counter unit, the sorter, and the control circuit. Fig. 7 shows the block diagram of parallel counter unit. Because the processing time for the parallel counter unit is long, this unit processes

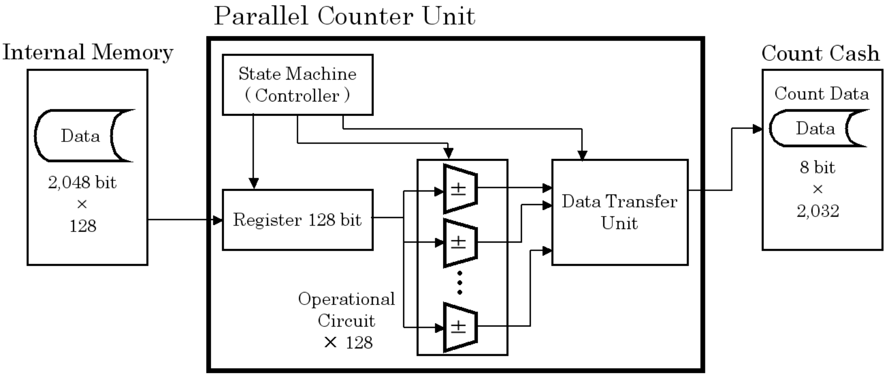


Fig. 7. Block diagram of Parallel Counter Unit

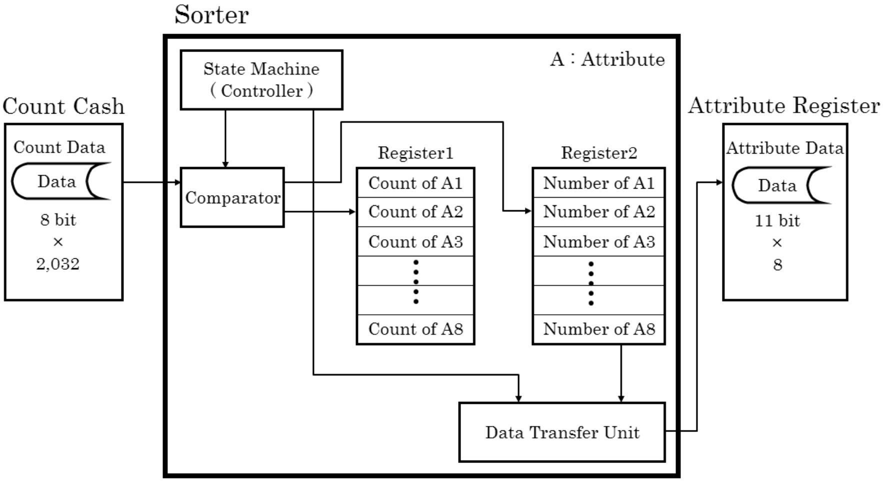


Fig. 8. Block diagram of Sorter

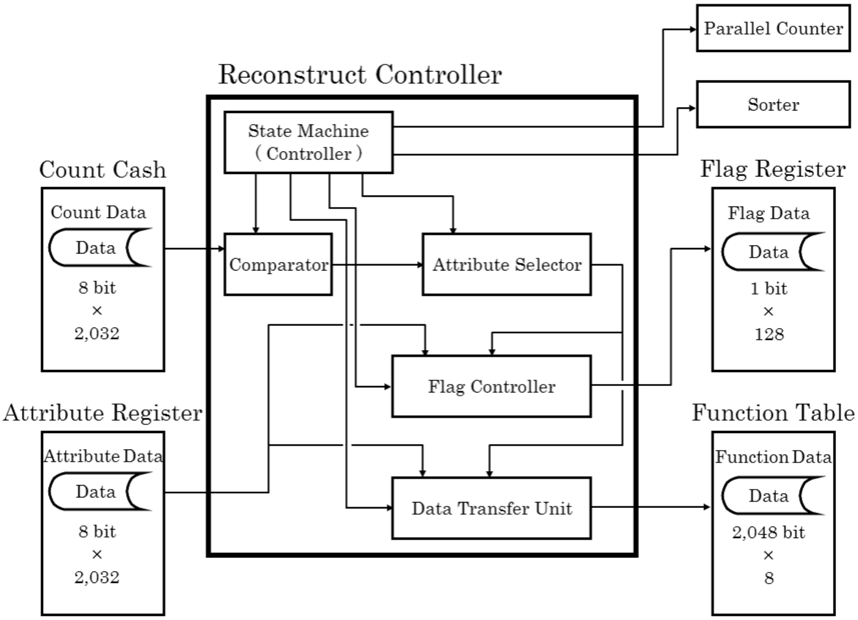


Fig. 9. Block diagram of Reconstruction Controller

the data of 128 bits in parallel. Fig. 8 and Fig. 9 show the block diagram of sorter and reconstruction controller [3][4].

3 Design and Simulation

The authors performed the design and the simulation using the logic synthesis tool (Xilinx ISE WebPACK 8.2i) and HDL simulator (Mentor Graphics ModelSim XE 6.1e) [5]. The target FPGA is Spartan3E (500 thousands gates, Xilinx Inc.) [5]. In this paper, the input data size is chosen as 2,032 columns and 128 rows. The number of core data is eight and the number of rule is eight. Table 1 summarizes the synthesized gate number and simulated processing time.

Table 1. Gate number and processing time

Unit name	Gate number (gates)	Processing time (μs)
Core-Selector (8 core data)	4,828	47
Covering-Unit (8 core data)	1,906	84
reconstruction-Unit (8 rules)	31,311	7,176
Total	38,045	7,307

4 Implementation and Experiments

The proposed processor and the test circuits have implemented on the FPGA evaluation board (Xilinx Spartan-3E Starter Kit). The FPGA board includes the 500 thousand gates FPGA LSI, switches, LEDs, VGA port, 2-line LCD display, etc.

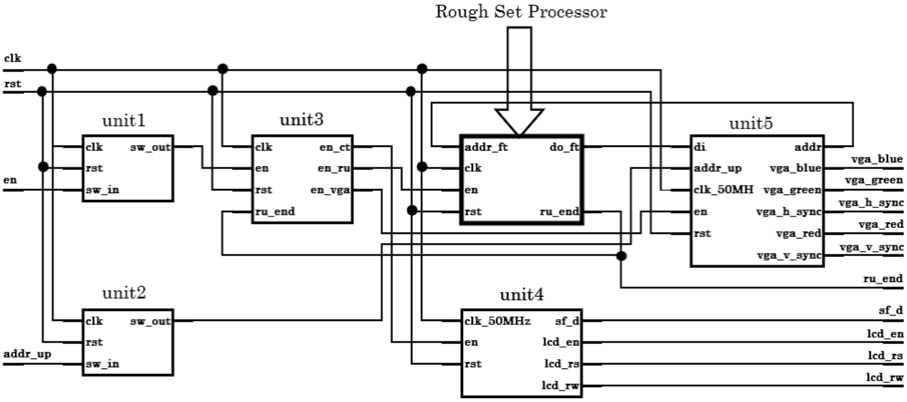


Fig. 10. Test Circuit

Fig. 10 shows the test circuit. Table 2 summarizes the units. Fig. 11 shows the photograph of experiment where a VGA monitor is connected to the FPGA board. Fig. 12 shows the example of obtained rules by the proposed processor. The input logical function is 2,032 bit and 128 rows random data.

Table 2. Units of the test circuit

Unit	Function	Gate number (gates)
1, 2	Chattering prevention	447 + 447
3	Sequencer	47
4	Measurement of processing time	3,932
5	VGA monitor controller	2,036

A program was made by C language for the speed comparison with a general-purpose microprocessor. The process is only reconstruction. The development tool is “Visual Studio.net”, and OS is “Windows XP”. To measure pure processing time, the input data is built into the program code, and the screen drawing is avoided. Table 3 summarizes the comparison of processing time with PC. It is clarified that the proposed processor is ten times faster than PC, though the clock frequency is about 70 times lower.

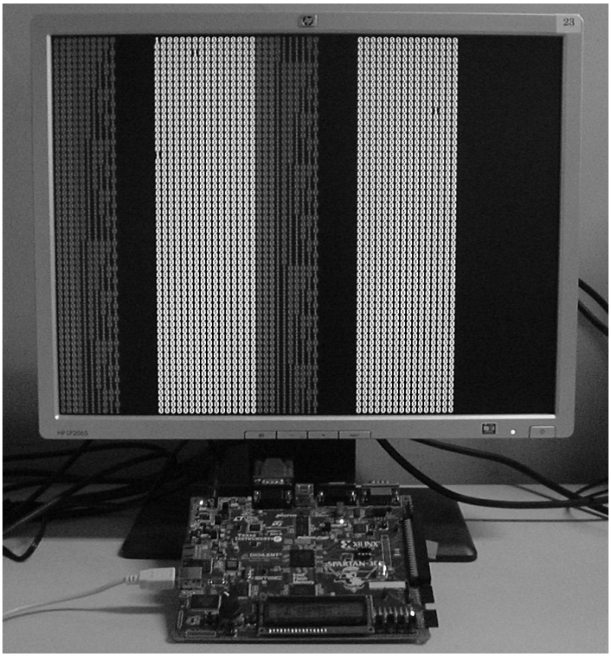
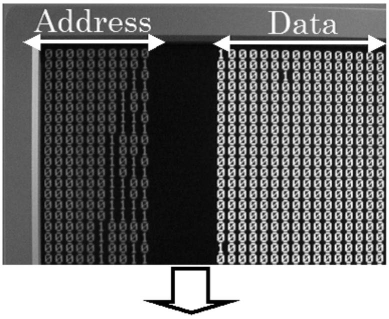


Fig. 11. Experiment with FPGA board and VGA monitor



- Decision Rule1 = $(X_0 \wedge X_{38} \wedge X_{288} \wedge X_{1148})$
- Decision Rule2 = $(X_{83} \wedge X_{788} \wedge X_{788} \wedge X_{1808})$
- Decision Rule3 = $(X_0 \wedge X_{50} \wedge X_{270} \wedge X_{877})$
- Decision Rule4 = $(X_0 \wedge X_{17} \wedge X_{1204} \wedge X_{1532})$
- Decision Rule5 = $(X_{26} \wedge X_{385} \wedge X_{430})$
- Decision Rule6 = $(X_{70} \wedge X_{385} \wedge X_{1355})$
- Decision Rule7 = $(X_{471} \wedge X_{1148} \wedge X_{1274})$
- Decision Rule8 = $(X_1 \wedge X_{17} \wedge X_{285} \wedge X_{1204})$

Fig. 12. Photograph of monitor (partial) and obtained rule

Table 3. Comparison of processing time with PC

	Clock	Time (μs) (reconstruction unit)
Proposed Processor	50 (MHz)	7,176
PC (Xeon)	3.4 (GHz)	72,539

Fig. 13 shows the LCD display (on board) for processing time. The experiment result (7,308 μs) is corresponding to the simulation result (7,307 μs) with high accuracy.

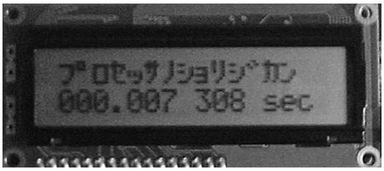


Fig. 13. LCD display (on board) for processing time

5 Conclusion

In this paper, the design, simulation, implementation and experiment of rough set processor are described. The processor was implemented on a 500 thousands gates FPGA evaluation board with VGA monitor. The experiment result shows that the proposed processor is ten times faster than PC, though the clock frequency is about 70 times lower. The problem in the future is further speed-up. The pipeline processing, the optimization of clock frequency and parallel operation of the processor are effective for the achievement.

Acknowledgments. The author express gratitude to professor Skowron of Warsaw university. This research is partially supported by Research Institute for Science and Technology of Tokyo Denki University under Grant Q03-04.

References

1. Pawlak, Z.: Rough Sets – Theoritical Aspects of Reasoning about Data-. Kluwer Academic Publishers, Dordrecht Boston London (1991)
2. Pal, S.K., Skowron, A.: Rough Fuzzy Hybridization – A New Trend in Decision Making-. Springer, Heidelberg (1999)
3. Kanasugi, A.: Design of Architecture for Rough Set Processor. In: Proc. Int. Workshop on Rough Set Theory and Granular Computing, pp. 201–204 (2001)
4. Kanasugi, A.: In: Inuguchi M., et al. (eds) Rough Set Theory and Granular Computing. pp. 273–280. Springer, Heidelberg (2002)
5. http://www.xilinx.com/

A Computationally Efficient Nonlinear Predictive Control Algorithm with RBF Neural Models and Its Application

Maciej Ławryńczuk and Piotr Tatjewski

Institute of Control and Computation Engineering, Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland, tel. +48 22 234-73-97
M.Lawrynczuk@ia.pw.edu.pl, P.Tatjewski@ia.pw.edu.pl

Abstract. This paper details a computationally efficient (suboptimal) nonlinear Model Predictive Control (MPC) algorithm with Radial Basis Function (RBF) type neural network models and discusses its application to a polymerisation reactor. Neural model of the process is used on-line to determine the local linearisation and the nonlinear free trajectory. Unlike the nonlinear MPC technique, which hinges on non-convex optimisation, the presented algorithm is more reliable and less computationally demanding because it results in a quadratic programming problem, whereas its closed-loop control performance is similar.

1 Introduction

Model Predictive Control (MPC) is recognised as the only advanced control technique which has been very successful in practical applications [2], [8], [14], [15], [16]. MPC algorithms can take into account constraints imposed on both process inputs (manipulated variables) and outputs (controlled variables), which usually decide on quality, economic efficiency and safety. Furthermore, MPC techniques are very efficient in multivariable process control.

Structure of the model used in nonlinear MPC decides on the controller's accuracy and computational burden. Fundamental (first-principles) models, although potentially very precise, are usually not suitable for on-line control because they are very complicated and may lead to numerical problems. Since neural network models [1] are able to approximate precisely nonlinear behaviour of technological processes [3], [10], have relatively small number of parameters and simple structure, they can be effectively used in MPC algorithms as process models. Feedforward perceptron multilayer networks [5], [6], [7], [10], [15], [16] and RBF (Radial Basis Function) networks [13] are usually considered.

The paper describes the computationally efficient MPC with Nonlinear Prediction and Linearisation (MPC-NPL) algorithm with RBF type neural network models and its application to a polymerisation process. Analogously to its variant with perceptron type neural models [5], [6], [7], [15], [16], the algorithm gives good closed-loop performance and, unlike the nonlinear MPC technique, which hinges on nonlinear optimisation, it uses on-line only the numerically reliable quadratic programming approach.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 603–612, 2007.

2 Model Predictive Control Algorithms

In the MPC algorithms [8], [16] at each consecutive sampling instant k a set of future control increments is calculated

$$\Delta\boldsymbol{u}(k) = [\Delta u(k|k)\ \Delta u(k+1|k)\ldots\Delta u(k+N_u-1|k)]^T \tag{1}$$

It is assumed that $\Delta u(k+p|k) = 0$ for $p \geq N_u$, where N_u is the control horizon. The objective is to minimise the differences between the predicted values of the output $\hat{y}(k+p|k)$ and the reference trajectory $y^{ref}(k+p|k)$ over the prediction horizon N. The following quadratic cost function is usually used

$$J(k) = \sum_{p=1}^{N}(y^{ref}(k+p|k) - \hat{y}(k+p|k))^2 + \sum_{p=0}^{N_u-1}\lambda_p(\Delta u(k+p|k))^2 \tag{2}$$

where $\lambda_p > 0$ are weighting factors. Typically, $N_u < N$, which decreases the dimensionality of the optimisation problem and leads to smaller computational load. Only the first element of the determined sequence (1) is applied to the process, the control law is then

$$u(k) = \Delta u(k|k) + u(k-1) \tag{3}$$

At next sampling instant, $k+1$, the prediction is shifted one step forward and the whole procedure is repeated.

Since the constraints have to be usually taken into account, future control increments are determined as the solution to the following optimisation problem (assuming hard output constraints [8], [16] for simplicity of presentation)

$$\begin{gathered}\min_{\Delta u(k|k)\ldots\Delta u(k+N_u-1|k)}\{J(k)\}\\ u^{\min} \leq u(k+p|k) \leq u^{\max} \quad p = 0,\ldots,N_u-1\\ -\Delta u^{\max} \leq \Delta u(k+p|k) \leq \Delta u^{\max} \quad p = 0,\ldots,N_u-1\\ y^{\min} \leq \hat{y}(k+p|k) \leq y^{\max} \quad p = 1,\ldots,N\end{gathered} \tag{4}$$

Predicted output values of the output over the prediction horizon are calculated using a dynamic model of the process. The choice of the model (linear or nonlinear, if nonlinear – fundamental or black-box) is crucial. This decision affects not only the possible control accuracy but also the computational load and reliability of the whole control policy. MPC algorithms based on linear models have been usually applied in practice [2], [14], [16], since the predictions $\hat{y}(k+p|k)$ can be expressed as a linear combination of decision variables, which means that the optimisation problem (4) is a quadratic programming one [8], [15], [16]. Unfortunately, when the process exhibits severe nonlinearity, such an approach is likely to result in poor closed-loop control performance, even instability. In general, a nonlinear model used for prediction purposes leads to a non-quadratic, non-convex and even multi-modal optimisation problem. For such problems there are no sufficiently fast and reliable optimisation algorithms, i.e. those which would be able to determine the global optimal solution at each sampling instant and within predefined time limit as it is required in on-line control. Gradient based optimisation techniques may terminate in local minima while global ones substantially increase the computational burden, yet they still give no guarantee that the global solution is found.

To overcome the computational problems inevitable in MPC with nonlinear optimisation, a few alternatives have been found. For example, affine nonlinear models of neural structure result in a quadratic programming problem [4]. The whole MPC algorithm can be approximated by a trained off-line neural network [11]. Yet another option is to use a combination of a neural steady-state model and a simplified nonlinear second order quadratic dynamic model [12]. Although the resulting optimisation task is not convex, the model is relatively simple, the approach is reported to be successful in many industrial applications.

Bearing in mind all the aforementioned computational difficulties typical of nonlinear MPC, linearisation-based MPC techniques, in which only a quadratic programming problem is solved on-line, are reasonable alternatives. Compared to MPC algorithms with full nonlinear optimisation, they are suboptimal, but in most practical applications the accuracy is sufficient [2], [5], [6], [7], [15], [16].

3 MPC-NPL Algorithm with Neural Models

3.1 Neural Model of the Process

Let the Single-Input Single-Output (SISO) process under consideration be described by the following nonlinear discrete-time equation

$$y(k) = g(\boldsymbol{x}(k)) = g(u(k-\tau), \ldots, u(k-n_B), y(k-1), \ldots, y(k-n_A)) \tag{5}$$

where $g : \Re^{n_A+n_B-\tau+1} \longrightarrow \Re \in C^1$, $\tau \leq n_B$. The RBF type neural network containing one hidden layer with Gaussian functions and linear output is used as the function g in (5). Output of the model can be expressed as

$$\begin{aligned} y(k) = g(\boldsymbol{x}(k)) &= w_0 + \sum_{i=1}^{K} w_i \exp(-\left\|\boldsymbol{x}(k) - \boldsymbol{c}_i\right\|_{\boldsymbol{Q}_i}) \\ &= w_0 + \sum_{i=1}^{K} w_i \exp(-z_i(k)) \end{aligned} \tag{6}$$

where K is the number of hidden nodes. The vectors $\boldsymbol{c}_i$ and the diagonal weighting matrices $\boldsymbol{Q}_i = diag(q_{i,1}, \ldots, q_{i,n_A+n_B-\tau+1})$ describe centres and widths of the nodes, respectively, $i = 1, \ldots, K$. The model (6) is sometimes named Hyper Radial Basis Function (HRBF) neural network in contrast to the ordinary RBF neural networks in which widths of the nodes are constant. Let $z_i(k)$ be the sum of inputs of the i-th hidden node. Recalling the arguments of the model (5)

$$z_i(k) = \sum_{j=1}^{I_u} q_{i,j}(u(k-\tau+1-j) - c_{i,j})^2 + \sum_{j=1}^{n_A} q_{i,I_u+j}(y(k-j) - c_{i,I_u+j})^2 \tag{7}$$

where the number of the network's input nodes depending on input signal u is $I_u = n_B - \tau + 1$.

Considering the prediction over the horizon N, the quantities $z_i(k+p|k)$ and consequently $\hat{y}(k+p|k)$ depend on future values of control signal (i.e. decision variables of the control algorithm), values of control signal applied to the plant at previous sampling instants, future output predictions and measured values of the plant output signal. From (7) one has

$$\begin{aligned} z_i(k+p|k) = &\sum_{j=1}^{I_{uf}(p)} q_{i,j}(u(k-\tau+1-j+p|k) - c_{i,j})^2 \\ &+ \sum_{j=I_{uf}(p)+1}^{I_u} q_{i,j}(u(k-\tau+1-j+p) - c_{i,j})^2 \\ &+ \sum_{j=1}^{I_{yp}(p)} q_{i,I_u+j}(\hat{y}(k-j+p|k) - c_{i,I_u+j})^2 \\ &+ \sum_{j=I_{yp}(p)+1}^{n_A} q_{i,I_u+j}(y(k-j+p) - c_{i,I_u+j})^2 \end{aligned} \tag{8}$$

where $I_{uf}(p) = \max(\min(p-\tau+1, I_u), 0)$ is the number of the network's input nodes depending on future control signals and $I_{yp}(p) = \min(p-1, n_A)$ is the number of the network's input nodes depending on output predictions.

3.2 MPC-NPL Optimisation Problem

In the MPC-NPL algorithm at each sampling instant k the neural model is used on-line twice: to determine the local linearisation and the nonlinear free trajectory. It is assumed that the output prediction can be expressed as the sum of the forced trajectory, which depends only on the future (on future input moves $\Delta \boldsymbol{u}(k)$) and the free trajectory $\boldsymbol{y}^0(k)$, which depends only on the past [16]

$$\hat{\boldsymbol{y}}(k) = \boldsymbol{y}^0(k) + \boldsymbol{G}(k)\Delta \boldsymbol{u}(k) \tag{9}$$

where

$$\hat{\boldsymbol{y}}(k) = [\hat{y}(k+1|k) \ldots \hat{y}(k+N|k)]^T \tag{10}$$

$$\boldsymbol{y}^0(k) = \left[y^0(k+1|k) \ldots y^0(k+N|k)\right]^T \tag{11}$$

The dynamic matrix $\boldsymbol{G}(k)$ of dimension $N \times N_u$ is calculated on-line from the nonlinear model taking into account the current state of the plant

$$\boldsymbol{G}(k) = \begin{bmatrix} s_1(k) & 0 & \ldots & 0 \\ s_2(k) & s_1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ s_N(k) & s_{N-1}(k) & \ldots & s_{N-N_u+1}(k) \end{bmatrix} \tag{12}$$

The step-response coefficients of the linearised model are determined from

$$s_j(k) = \sum_{i=1}^{\min(j,n_B)} b_i(k) - \sum_{i=1}^{\min(j-1,n_A)} a_i(k) s_{j-i}(k) \tag{13}$$

where $a_i(k)$ and $b_i(k)$ are coefficients of the linearised model. Calculation of these quantities and the nonlinear free trajectory is detailed in the following subsections.

On the one hand, the suboptimal prediction calculated from (9) is different from the optimal one determined from the nonlinear neural model as it is done in MPC algorithms with nonlinear optimisation [5], [7], [15], [16]. On the other hand, thanks to using the superposition principle (9), the optimisation problem (4) becomes the following quadratic programming task

$$\begin{gathered}\min_{\boldsymbol{\Delta u}(k)} \left\{ J(k) = \left\| \boldsymbol{y}^{ref}(k) - \boldsymbol{y}^0(k) - \boldsymbol{G}(k)\boldsymbol{\Delta u}(k) \right\|^2 + \left\| \boldsymbol{\Delta u}(k) \right\|^2_{\boldsymbol{\Lambda}} \right\} \\ \boldsymbol{u}^{\min} \leq \boldsymbol{J}\boldsymbol{\Delta u}(k) + \boldsymbol{u}^{k-1} \leq \boldsymbol{u}^{\max} \\ -\boldsymbol{\Delta u}^{\max} \leq \boldsymbol{\Delta u}(k) \leq \boldsymbol{\Delta u}^{\max} \\ \boldsymbol{y}^{\min} \leq \hat{\boldsymbol{y}}(k) \leq \boldsymbol{y}^{\max}\end{gathered} \tag{14}$$

where the vectors of length N are

$$\boldsymbol{y}^{ref}(k) = \left[y^{ref}(k+1|k) \ldots y^{ref}(k+N|k) \right]^T \tag{15}$$

$$\boldsymbol{y}^{\min}(k) = \left[y^{\min} \ldots y^{\min} \right]^T \tag{16}$$

$$\boldsymbol{y}^{\max}(k) = [y^{\max} \ldots y^{\max}]^T \tag{17}$$

the vectors of length N_u are

$$\boldsymbol{u}^{\min}(k) = \left[u^{\min} \ldots u^{\min} \right]^T \tag{18}$$

$$\boldsymbol{u}^{\max}(k) = [u^{\max} \ldots u^{\max}]^T \tag{19}$$

$$\boldsymbol{\Delta u}^{\max}(k) = [\Delta u^{\max} \ldots \Delta u^{\max}]^T \tag{20}$$

$$\boldsymbol{u}^{k-1}(k) = [u(k-1) \ldots u(k-1)]^T \tag{21}$$

the matrix of dimension $N_u \times N_u$ is

$$\boldsymbol{J} = \begin{bmatrix} 1\;0\;0\ldots0 \\ 1\;1\;0\ldots0 \\ \vdots\;\vdots\;\vdots\;\ddots\;\vdots \\ 1\;1\;1\ldots1 \end{bmatrix} \tag{22}$$

and $\boldsymbol{\Lambda} = diag(\lambda_0, \ldots, \lambda_{N_u-1})$. Structure of the MPC-NPL algorithm is depicted in Fig. 1. At each sampling instant k the following steps are repeated:

1. Linearisation of the neural model: obtain the matrix $\boldsymbol{G}(k)$.
2. Calculate the nonlinear free trajectory $\boldsymbol{y}^0(k)$ using the neural model.
3. Solve the quadratic programming problem (14) to determine $\boldsymbol{\Delta u}(k)$.
4. Apply $u(k) = \Delta u(k|k) + u(k-1)$.
5. Set $k := k+1$, go to step 1.

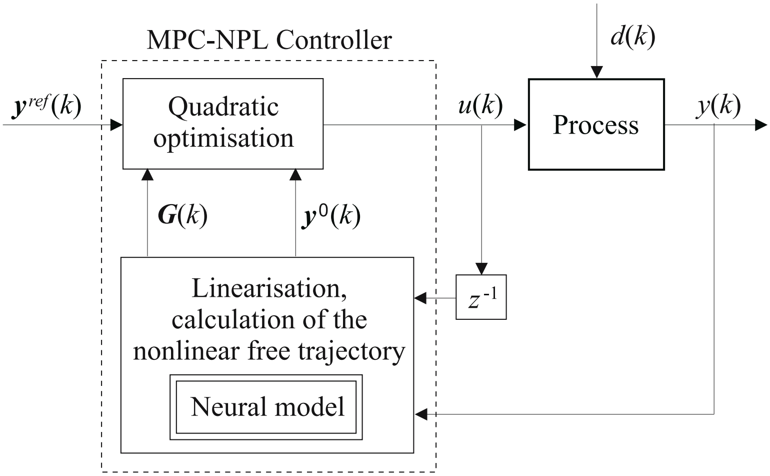


Fig. 1. Structure of the MPC algorithm with Nonlinear Prediction and Linearisation (MPC-NPL)

3.3 On-Line Linearisation of the Neural Model

Defining a linearisation point as the vector composed of past input and output signal values corresponding to the arguments of the nonlinear model (5)

$$\bar{\boldsymbol{x}}(k) = [\bar{u}(k-\tau)\ \ldots\ \bar{u}(k-n_B)\ \bar{y}(k-1)\ \ldots\ \bar{y}(k-n_A)]^T \tag{23}$$

and using Taylor series expansion at this point, the linear approximation of the model, obtained at sampling instant k, can be expressed as

$$\begin{aligned} y(k) = g(\bar{\boldsymbol{x}}(k)) &+ \sum_{l=1}^{n_B} b_l(\bar{\boldsymbol{x}}(k))(u(k-l) - \bar{u}(k-l)) \\ &- \sum_{l=1}^{n_A} a_l(\bar{\boldsymbol{x}}(k))(y(k-l) - \bar{y}(k-l)) \end{aligned} \tag{24}$$

The coefficients of the linearised model are calculated from

$$a_l(\bar{\boldsymbol{x}}(k)) = -\frac{\partial g(\bar{\boldsymbol{x}}(k))}{\partial y(k-l)} \quad l = 1, \ldots, n_A \tag{25}$$

$$b_l(\bar{\boldsymbol{x}}(k)) = \begin{cases} 0 & l = 1, \ldots, \tau - 1 \\ \dfrac{\partial g(\bar{\boldsymbol{x}}(k))}{\partial u(k-l)} = & l = \tau, \ldots, n_B \end{cases} \tag{26}$$

Taking into account the structure of the neural model and corresponding equations (6) and (7), one has

$$a_l(\bar{\boldsymbol{x}}(k)) = 2\sum_{i=1}^{K} w_i \exp(-z_i(\bar{\boldsymbol{x}}(k))) q_{i,I_u+l}(y(k-l) - c_{i,I_u+l}) \quad l = 1, \ldots, n_A \tag{27}$$

$$b_l(\bar{\boldsymbol{x}}(k)) = \begin{cases} 0 & l = 1,\ldots,\tau-1 \\ -2\sum_{i=1}^{K} w_i \exp(-z_i(\bar{\boldsymbol{x}}(k))) q_{i,l-\tau+1}(u(k-l) - c_{i,l-\tau+1}) & l = \tau,\ldots,n_B \end{cases} \tag{28}$$

3.4 Calculation of the Nonlinear Free Trajectory

The nonlinear free trajectory $y^0(k+p|k)$, $p = 1,\ldots,N$, is calculated on-line recursively from the general prediction equation

$$\hat{y}(k+p|k) = y(k+p|k) + d(k) \tag{29}$$

where the quantities $y(k+p|k)$ are calculated from the nonlinear neural model (6). The "DMC type" disturbance model is used. The unmeasured disturbance $d(k)$ is assumed to be constant over the prediction horizon. It is estimated from

$$d(k) = y(k) - y(k|k-1) = y(k) - \left(w_0 + \sum_{i=1}^{K} w_i \exp(-z_i(k)) \right) \tag{30}$$

where $y(k)$ is a measured value while the quantity $y(k|k-1)$ is calculated from the model (6). From (29) the nonlinear free trajectory is given by

$$y^0(k+p|k) = w_0 + \sum_{i=1}^{K} w_i \exp(-z_i^0(k+p|k)) + d(k) \tag{31}$$

The quantities $z_i^0(k+p|k)$ are determined from (8) assuming no changes in control signals from sampling instant k onwards and replacing predicted output signals from $k+1$ by corresponding values of the free trajectory

$$\begin{aligned} u(k+p|k) &:= u(k-1) \quad p \geq 0 \\ \hat{y}(k+p|k) &:= y^0(k+p|k) \quad p \geq 1 \end{aligned} \tag{32}$$

hence

$$\begin{aligned} z_i^0(k+p|k) = & \sum_{j=1}^{I_{uf}(p)} q_{i,j}(u(k-1) - c_{i,j})^2 \\ & + \sum_{j=I_{uf}(p)+1}^{I_u} q_{i,j}(u(k-\tau+1-j+p) - c_{i,j})^2 \\ & + \sum_{j=1}^{I_{yp}(p)} q_{i,I_u+j}(y^0(k-j+p|k) - c_{i,I_u j})^2 \\ & + \sum_{j=I_{yp}(p)+1}^{n_A} q_{i,I_u+j}(y(k-j+p) - c_{i,I_u j})^2 \end{aligned} \tag{33}$$

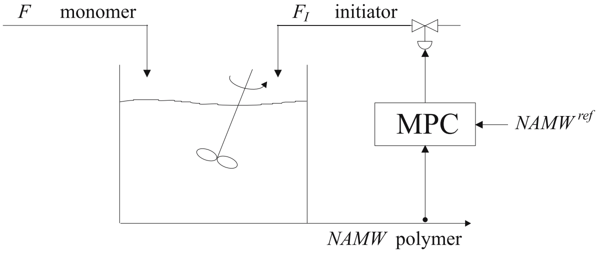


Fig. 2. Polymerisation reactor control system structure

4 Simulation Results

The process under consideration is a polymerisation reaction taking place in a jacketed continuous stirred tank reactor [9] depicted in Fig. 2. The reaction is the free-radical polymerisation of methyl methacrylate with azo-bis-isobutyronitrile as initiator and toluene as solvent. The output $NAMW$ (Number Average Molecular Weight) is controlled by manipulating the inlet initiator flow rate F_I. Flow rate F of the monomer is assumed to be constant.

Three models of the process are used. The fundamental model [9] is used as the real process during simulations. An identification procedure is carried out, a linear model and a neural one are obtained. Both empirical models have the same input arguments determined by $\tau = 2$, $n_A = n_B = 2$. The horizons are $N = 10$, $Nu = 3$, the weighting coefficients $\lambda_p = 0.2$. The manipulated variable is constrained, $F_I^{\min} = 0.003$, $F_I^{\max} = 0.06$, the sampling time is 1.8 min.

As the reference trajectories, three set-point changes occurring at $k = 1$ are considered, namely from $NAMW = 20000$ to $NAMW = 25000$, $NAMW = 30000$ and $NAMW = 40000$, respectively. Simulation results of the MPC algo-

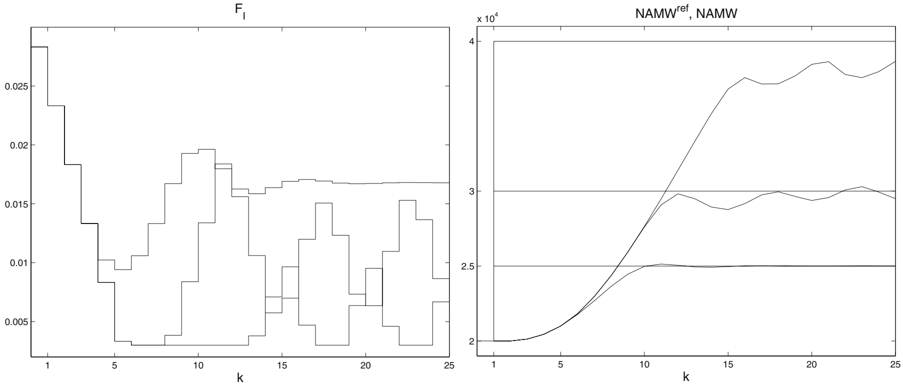


Fig. 3. Simulation results of the MPC algorithm with the linear model

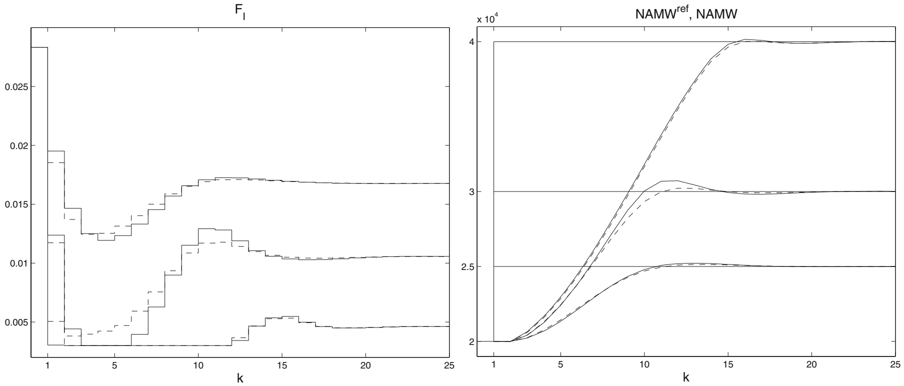


Fig. 4. Simulation results of the MPC-NPL (dotted) and MPC-NO (solid) algorithms with the same neural network model

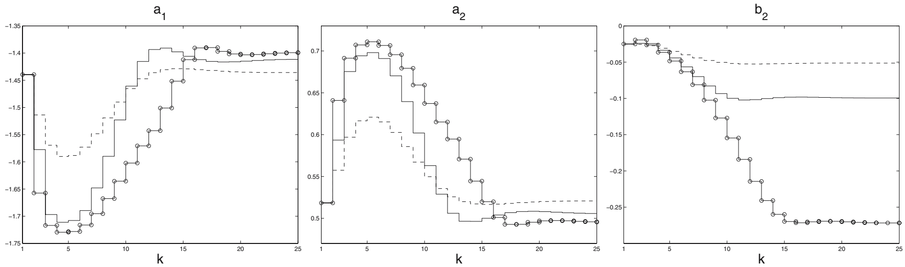


Fig. 5. Coefficients $a_1(k)$, $a_2(k)$, $b_2(k)$ of the linearised model in the MPC-NPL algorithm with the neural network model: $NAMW^{ref} = 25000$ (dotted), $NAMW^{ref} = 30000$ (solid), $NAMW^{ref} = 40000$ (solid-circles)

rithm with the linear model are depicted in Fig. 3. It works well for the smallest set-point change, but for medium and big ones the system becomes unstable. Simulation results of the MPC-NPL and MPC with Nonlinear Optimisation (MPC-NO) algorithms with the same neural network model are depicted in Fig. 4. Both nonlinear algorithms are stable. Moreover, for three considered set point changes the closed-loop performance obtained in the suboptimal MPC-NPL algorithm with quadratic programming is very close to that obtained in computationally prohibitive MPC-NO approach, in which a nonlinear optimisation problem has to be solved on-line at each sampling instant. The bigger the change in the set-point variable, the bigger the changes of the coefficients $a_1(k)$, $a_2(k)$, $b_2(k)$ of the linearised model as it is shown in Fig. 5.

5 Conclusion

Reliability, computational efficiency and closed-loop accuracy are the advantages of the presented MPC-NPL algorithm with RBF type neural network models.

The MPC-NPL algorithm uses on-line only the numerically reliable quadratic programming procedure, the necessity of full nonlinear optimisation is avoided. Although suboptimal, in practice the algorithm gives performance comparable to that obtained in MPC schemes with nonlinear optimisation.

Neural networks (primarily perceptron type and RBF type) are able to approximate precisely nonlinear nature of processes and, unlike fundamental models, have relatively small number of parameters and simple structure. Hence, they are particularly suitable to be used in nonlinear MPC algorithms.

References

1. Haykin, S.: Neural networks – a comprehensive foundation. Prentice-Hall, Englewood Cliffs (1999)
2. Henson, M.A.: Nonlinear model predictive control: current status and future directions. Computers and Chemical Engineering. 23, 187–202 (1998)
3. Hussain, M.A.: Review of the applications of neural networks in chemical process control – simulation and online implmementation. Artificial Intelligence in Engineering 13, 55–68 (1999)
4. Liu, G.P., Kadirkamanathan, V., Billings, S.A.: Predictive control for non-linear systems using neural networks. International Journal of Control 71, 1119–1132 (1998)
5. Ławryńczuk, M.: A family of model predictive control algorithms with artificial neural networks. Submitted to International Journal of Applied Mathematics and Computer Science
6. Ławryńczuk, M., Tatjewski, P.: An efficient nonlinear predictive control algorithm with neural models and its application to a high-purity distillation process. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 76–85. Springer, Heidelberg (2006)
7. Ławryńczuk, M.: Nonlinear model predictive control algorithms with neural models of processes (in Polish). PhD thesis. Warsaw University of Technology (2003)
8. Maciejowski, J.M.: Predictive control with constraints, Harlow. Prentice-Hall, Englewood Cliffs (2002)
9. Maner, B.R., Doyle, F.J., Ogunnaike, B.A., Pearson, R.K.: Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order Volterra models. Automatica 32, 1285–1301 (1996)
10. Nørgaard, M., Ravn, O., Poulsen, N.K., Hansen, L.K.: Neural networks for modelling and control of dynamic systems. Springer, Heidelberg (2000)
11. Parisini, T., Sanguineti, M., Zoppoli, R.: Nonlinear stabilization by receding-horizon neural regulators. International Journal of Control 70, 341–362 (1998)
12. Piche, S., Sayyar-Rodsari, B., Johnson, D., Gerules, M.: Nonlinear model predictive control using neural networks. IEEE Control Systems Magazine 20, 56–62 (2000)
13. Pottmann, M., Seborg, D.E.: A nonlinear predictive control strategy based on radial basis function models. Computers and Chemical Engineering 21, 965–980 (1997)
14. Qin, S.J., Badgwell, T.A.: A survey of industrial model predictive control technology. Control Engineering Practice 11, 733–764 (2003)
15. Tatjewski, P.: Soft computing in model-based predictive control. International Journal of Applied Mathematics and Computer Science 16, 101–120 (2006)
16. Tatjewski, P.: Advanced control of industrial processes, Structures and algorithms. Springer, Heidelberg (2007)

Operations on Interval Matrices

Barbara Pękala

Institute of Mathematics, University of Rzeszów,
Rejtana 16A, 35-310 Rzeszów, Poland
bpekala@univ.rzeszow.pl

Abstract. We consider algebraic properties of interval matrices. Operations on interval matrices are strictly connected with interval-valued fuzzy sets. We examine lattice and semigroup properties of interval matrices. Next, we discuss asymptotic properties in semigroup of powers. In particular, a convergence of power sequence is examined. There is a dependence between this convergence and convergence of powers of boundary matrices.

Keywords: lattice, interval, interval matrices.

Math. Sub. Class.: 15A30, 06D72, 65G30.

1 Introduction

In 1965 L.A. Zadeh presented fuzzy sets. Since then many new extensions of this concept have been given as IFS (cf. [1]) or IVFS (cf. [6], [17]). Arithmetic operations on sets were first introduced in [18]. Standard properties of interval arithmetic are completed in [13], where set of intervals form a lattice (non distributive lattice). Then $\sup - \inf$ product of matrices over this lattice is not associative. The convergence of powers of interval matrices was studied in [9], [10] and [11].

In this paper we consider lattice operations on intervals induced by partial order between intervals. Such lattice is distributive and we can consider $\sup - \inf$ product (associative) of matrices over this lattice. We want to examine some properties of this product of interval matrices.

Interval matrices, product of interval matrices and powers of a matrix can be used in describing of a system dynamics, interval errors (cf.[13], [8]), time-invariant fuzzy systems (cf.[7]) and systems analysis (cf.[14]).

First part of this paper deals with algebraic operations on sets. In the second part we consider interval lattice and interval matrices. Next we consider some facts connected with properties of elements in a semigroup. We concentrate on convergence of the sequence of powers for fuzzy matrices connected with interval matrices.

2 Algebraic Operations on Sets

At the beginning we recall on algebraic operations on the sets.

Definition 1 (cf.[12], p.8). *Let $(X, *)$ be a nonempty set with a binary operation. Extension of the operation $*$ on sets $A, B \subset X$ we call the operation*

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 613–621, 2007.

$$A * B := \{a * b : \ a \in A, \ b \in B\} \subset X.$$

For this operation the following theorems hold.

Theorem 1 (cf.[12], p.9). *Operation $*$ in 2^X preserves commutativity, associativity and the neutral element of operation $*$ in X.*

Proof. If the operation $*$ is commutative in X, then
$A * B = \{a * b : \ a \in A, b \in B\} = \{b * a : \ b \in B, a \in A\} = B * A$.
If the operation $*$ is associative in X, then
$(A * B) * C = \{(a * b) * c : \ a \in A, b \in B, c \in C\} = \{a * (b * c) :$
$a \in A, b \in B, c \in C\} = A * (B * C)$.
If the operation $*$ has the neutral element e in X, then
$A * \{e\} = \{a * e : \ a \in A\} = \{e * a : \ a \in A\} = \{a : \ a \in A\}$.

Theorem 2. *Let $(X, *, \diamond)$ be a nonempty set with two binary operations. Distributivity and absorption of the operation $*$ over the operation $\diamond$ in X implies distributivity and absorption of this operations in 2^X.*

Proof. Let the operation $*$ be left-distributive over $\diamond$ in X. We obtain
$A * (B \diamond C) = \{a * (b \diamond c) : \ a \in A, b \in B, c \in C\} =$
$\{(a * b) \diamond (a * c) : \ a \in A, b \in B, c \in C\} = (A * B) \diamond (A * C)$.
By analogy we can prove the right-distributivity.
We consider the right-absorption:
$(A \diamond B) * A = \{(a \diamond b) * a : \ a \in A, b \in B\} = \{a : \ a \in A\} = A$.
By analogy we can prove the left-absorption.

3 Interval Calculus

Interval calculus is an extension of arithmetic operations on closed intervals. We consider closed intervals: $A = [\underline{a}, \overline{a}]$, $B = [\underline{b}, \overline{b}]$, $A, B \subset [0,1]$, $A, \ B \neq \emptyset$ $(\underline{a} \leq \overline{a}, \ \underline{b} \leq \overline{b})$ with the relation

$$\underset{A,B \subset [0,1]}{\forall} A \leq B \Leftrightarrow \underline{a} \leq \underline{b}, \ \overline{a} \leq \overline{b}. \tag{1}$$

The relation (1) is the partial order relation (cf. [15]). In this paper we put $S = \{[\underline{a}, \overline{a}] : \ \underline{a}, \overline{a} \in [0,1], \ \underline{a} \leq \overline{a}\}$, where $[0,0] = \min S$, $[1,1] = \max S$ with respect to (1).

Theorem 3. *If $A, B \in S$, then operations generated by binary* min " $\wedge$ " *and* max " $\vee$ " *have the form*

$$A \wedge B = [\underline{a} \wedge \underline{b}, \ \overline{a} \wedge \overline{b}], \ A \vee B = [\underline{a} \vee \underline{b}, \ \overline{a} \vee \overline{b}]. \tag{2}$$

Proof. Using binary operations min and max in Definition 1 we have
$x \in A \wedge B \Rightarrow \underset{a \in A, \, b \in B}{\exists} x = a \wedge b \Rightarrow x \geq \underline{a} \wedge \underline{b}, \ x \leq \overline{a} \wedge \overline{b} \Rightarrow x \in [\underline{a} \wedge \underline{b}, \ \overline{a} \wedge \overline{b}]$.

Then $A \wedge B \subset [\underline{a} \wedge \underline{b},\ \overline{a} \wedge \overline{b}]$. Now
$x \in [\underline{a} \wedge \underline{b},\ \overline{a} \wedge \overline{b}] \Rightarrow \underline{a} \wedge \underline{b} \leq x \leq \overline{a} \wedge \overline{b} \Rightarrow \underset{a \in A,\, b \in B}{\exists}\ \underline{a} \leq a \leq \overline{a},\ \underline{b} \leq b \leq \overline{b}.$
Putting $a = (\underline{a} \vee x) \wedge \overline{a},\ b = (\underline{b} \vee x) \wedge \overline{b}$ we get
$a \wedge b = (\underline{a} \vee x) \wedge (\underline{b} \vee x) \wedge \overline{a} \wedge \overline{b} = ((\underline{a} \wedge \underline{b}) \vee x) \wedge (\overline{a} \wedge \overline{b}) = x \wedge (\overline{a} \wedge \overline{b}) = x.$
So $[\underline{a} \wedge \underline{b},\ \overline{a} \wedge \overline{b}] \subset A \wedge B$.
Analogously we can prove the formula (2) for $\vee$.

Definition 2 (cf.[3], p.6,12). *The lattice (distributive lattice)* $(L, \wedge, \vee)$ *is a non empty set L with two monotone, associative operations with laws of absorption (distributivity) and with relation*

$$a \leq b \Leftrightarrow a \wedge b = a \ for \ a, b \in L. \tag{3}$$

The condition (3) holds in S for relation (1) and operations (2). So from above definition and Theorems 1, 2 we have

Theorem 4. *The structure* $(S, \wedge, \vee)$ *with operations* (2) *forms a distributive lattice.*

In the following example we obtain lattice operations (2) for given intervals.

Example 1. *For* $A, B \in S$ *we can consider two cases:*
If A, B *are comparable, e.g.* $A = [0.1, 0.5]$, $B = [0.2, 0.8]$*, then* $A \wedge B = A$, $A \vee B = B$,
If A, B *are non-comparable, e.g.* $A = [0.1, 0.8]$, $B = [0.3, 0.5]$*, then we obtain two new intervals* $A \wedge B = [0.1, 0.5]$ *and* $A \vee B = [0.3, 0.8]$.

Definition 3 (cf.[3], p.6). *The lattice L is complete if for every set* $A \subset L$ *there exist extremal bounds* $\inf A \in L$, $\sup A \in L$ *denoted also by* $\bigwedge A$, $\bigvee A$.

For indexed family of intervals we have more general operations.

Definition 4. *Let* $T \neq \emptyset$, $A_t \in S$, $A_t = [\underline{a}_t, \overline{a}_t]$, $t \in T$*. We have*

$$\bigvee_{t \in T} A_t = [\sup_{t \in T} \underline{a}_t, \sup_{t \in T} \overline{a}_t], \qquad \bigwedge_{t \in T} A_t = [\inf_{t \in T} \underline{a}_t, \inf_{t \in T} \overline{a}_t]. \tag{4}$$

Theorem 5. *The lattice* $(S, \wedge, \vee)$ *is a complete lattice.*

Proof. Operations (4) give extend bounds of the family $(A_t)_{t \in T}$, $A_t \subset S$.
Let $x \in \bigvee_{t \in T} A_t$, i.e. $\underset{a_t \in A_t}{\exists}\ x = \sup_{t \in T} a_t$. So $x \geq \sup_{t \in T} \underline{a}_t$, $x \leq \sup_{t \in T} \overline{a}_t$ or $x \in [\sup_{t \in T} \underline{a}_t, \sup_{t \in T} \overline{a}_t]$. Thus $\bigvee_{t \in T} A_t \subset [\sup_{t \in T} \underline{a}_t, \sup_{t \in T} \overline{a}_t]$.
Now let $x \in [\sup_{t \in T} \underline{a}_t, \sup_{t \in T} \overline{a}_t]$. Putting $a_t = (\overline{a}_t \wedge x) \vee \underline{a}_t$, we get
$\sup_{t \in T} a_t = \sup_{t \in T}[(\overline{a}_t \wedge x) \vee \underline{a}_t] = \sup_{t \in T}(\overline{a}_t \wedge x) \vee \sup_{t \in T} \underline{a}_t =$
$[(\sup_{t \in T} \overline{a}_t) \wedge x] \vee (\sup_{t \in T} \underline{a}_t) = x \vee (\sup_{t \in T} \underline{a}_t) = x,$
which proves the converse inclusion. The proof for operation $\bigwedge$ is similar.

Definition 5 (cf.[3], p.119). *The complete lattice* $(L, \vee, \wedge)$ *is infinitely distributive with respect to supremum (sup-distributive), if*

$$\underset{T\neq\emptyset}{\forall} \ \underset{t\in T}{\forall} \ \underset{a,b_t\in L}{\forall} \ a \wedge (\bigvee_{t\in T} b_t) = \bigvee_{t\in T} (a \wedge b_t), \tag{5}$$

and infinitely distributive with respect to infimum (inf-distributive), if

$$\underset{T\neq\emptyset}{\forall} \ \underset{t\in T}{\forall} \ \underset{a,b_t\in L}{\forall} \ a \vee (\bigwedge_{t\in T} b_t) = \bigwedge_{t\in T} (a \vee b_t). \tag{6}$$

The lattice sup-distributive and inf-distributive is called infinitely distributive.

Theorem 6. *The lattice* $(S, \vee, \wedge)$ *is infinitely distributive.*

Proof. Let $B = [\underline{b}, \overline{b}]$, $A_t = [\underline{a_t}, \overline{a_t}]$, $t \in T$. We consider property of sup-distributivity:
$[\underline{b}, \overline{b}] \wedge \bigvee_{t\in T} A_t = [\underline{b}, \overline{b}] \wedge [\sup_{t\in T} \underline{a_t}, \sup_{t\in T} \overline{a_t}] = [\underline{b} \wedge \sup_{t\in T} \underline{a_t}, \overline{b} \wedge \sup_{t\in T} \overline{a_t}] = [\sup_{t\in T}(\underline{b} \wedge \underline{a_t}), \sup_{t\in T}(\overline{b} \wedge \overline{a_t})] = \bigvee_{t\in T}[\underline{b} \wedge \underline{a_t}, \overline{b} \wedge \overline{a_t}] = \bigvee_{t\in T}([\underline{b}, \overline{b}][\underline{a_t}, \overline{a_t}])$,
which proves (5) in S. The proof of (6) is similar.

4 Interval Matrices

In this section we consider some properties of interval matrices.

Definition 6 (cf.[2], p. 277). *The set of matrices*

$$[\underline{A}, \overline{A}] = \{A = [a_{ij}] \in L^{m\times m} : \ \underline{a}_{ij} \leq a_{ij} \leq \overline{a}_{ij}, \ i, j = 1, 2, ..., m\}, \tag{7}$$

where $\underline{A}, \overline{A} \in [0, 1]^{m\times m}$, $\underline{A} = [\underline{a}_{ij}]$, $\overline{A} = [\overline{a}_{ij}]$ *we call the interval matrices.*

The above formula has double meaning. We have an interval of matrices and a matrix over the interval lattice $(S, \vee, \wedge)$. According to [5] we can consider the lattice of interval matrices. The set of all $m \times m$ interval matrices on S we denote by $S^{m\times m}$. In $S^{m\times m}$ we consider according to (1) the relation

$$\underset{[\underline{A},\overline{A}],[\underline{B},\overline{B}]\in S^{m\times m}}{\forall} [\underline{A}, \overline{A}] \leq [\underline{B}, \overline{B}] \Leftrightarrow \underline{A} \leq \underline{B}, \ \overline{A} \leq \overline{B}. \tag{8}$$

Moreover, for set of fuzzy matrices we have

$$\underset{[\underline{A},\overline{A}],[\underline{B},\overline{B}]\in S^{m\times m}}{\forall} [\underline{A}, \overline{A}] \leq [\underline{B}, \overline{B}] \Leftrightarrow \underset{1\leq i,j\leq m}{\forall} \underline{a}_{ij} \leq \underline{b}_{ij}, \ \overline{a}_{ij} \leq \overline{b}_{ij}. \tag{9}$$

Similarly, by (2) we get

$$([\underline{A}, \overline{A}] \vee [\underline{B}, \overline{B}])_{ij} = [\underline{a}_{ij} \vee \underline{b}_{ij}, \ \overline{a}_{ij} \vee \overline{b}_{ij}], \ ([\underline{A}, \overline{A}] \wedge [\underline{B}, \overline{B}])_{ij} = [\underline{a}_{ij} \wedge \underline{b}_{ij}, \ \overline{a}_{ij} \wedge \overline{b}_{ij}] \tag{10}$$

for $[\underline{A}, \overline{A}], [\underline{B}, \overline{B}] \in S^{m\times m}$.

Now (4) leads to

$$\bigvee_{t\in T}[\underline{A}_t,\overline{A}_t]_{ij}=[(\sup_{t\in T}\underline{a}_t)_{ij},(\sup_{t\in T}\overline{a}_t)_{ij}],\ \bigwedge_{t\in T}[\underline{A}_t,\overline{A}_t]_{ij}=[(\inf_{t\in T}\underline{a}_t)_{ij},(\inf_{t\in T}\overline{a}_t)_{ij}]. \tag{11}$$

Example 2. *We use the above formulas for matrices of intervals:*

$$[\underline{A},\overline{A}]=\begin{bmatrix}[0.2,\ 0.4] & [0.6,\ 0.9]\\ [0,\ 1] & [0.5,\ 0.8]\end{bmatrix},\ \underline{A}=\begin{bmatrix}0.2 & 0.6\\ 0 & 0.5\end{bmatrix},\ \overline{A}=\begin{bmatrix}0.4 & 0.9\\ 1 & 0.8\end{bmatrix},$$

$$[\underline{B},\overline{B}]=\begin{bmatrix}[0.4,\ 0.6] & [0.1,\ 0.7]\\ [0.9,\ 1] & [0.4,\ 0.8]\end{bmatrix},\ \underline{B}=\begin{bmatrix}0.4 & 0.1\\ 0.9 & 0.4\end{bmatrix},\ \overline{B}=\begin{bmatrix}0.6 & 0.7\\ 1 & 0.8\end{bmatrix},$$

and we have

$$[\underline{A},\overline{A}]\vee[\underline{B},\overline{B}]=\begin{bmatrix}[0.4,\ 0.6] & [0.6,\ 0.9]\\ [0.9,\ 1] & [0.5,\ 0.8]\end{bmatrix},$$

$$[\underline{A},\overline{A}]\wedge[\underline{B},\overline{B}]=\begin{bmatrix}[0.2,\ 0.4] & [0.1,\ 0.7]\\ [0,\ 1] & [0.4,\ 0.8]\end{bmatrix}.$$

Matrices over distributive lattice form a distributive lattice (cf.[5]), moreover, similarly to Theorem 6 we obtain

Theorem 7. $(S^{m\times m},\vee,\wedge)$ *is an infinitely distributive lattice.*

Now, we examine $\sup-\inf$ product of interval matrices.

Definition 7. *The* $\sup-\inf$ *product of* $[\underline{A},\overline{A}]$, $[\underline{B},\overline{B}]\in S^{m\times m}$ *is a matrix of elements:*

$$([\underline{A},\overline{A}]\circ[\underline{B},\overline{B}])_{ij}=\bigvee_{1\le k\le m}([\underline{A},\overline{A}]_{ik}\wedge[\underline{B},\overline{B}]_{kj}).$$

Theorem 8. *For* $[\underline{A},\overline{A}]$, $[\underline{B},\overline{B}]\in S^{m\times m}$ *we have*

$$([\underline{A},\overline{A}]\circ[\underline{B},\overline{B}])_{ij}=[(\underline{A}\circ\underline{B})_{ij},(\overline{A}\circ\overline{B})_{ij}], \tag{12}$$

where in the lattice $([0,1]^{m\times m},\vee,\wedge)$:

$$(A\circ B)_{ij}=\bigvee_{1\le k\le m}(a_{ik}\wedge b_{kj}). \tag{13}$$

Proof. By Definition 7, 3, 4 and (13) we obtain

$$([\underline{A},\overline{A}]\circ[\underline{B},\overline{B}])_{ij}=\bigvee_{1\le k\le m}([\underline{a}_{ik},\overline{a}_{ik}]\wedge[\underline{b}_{kj},\overline{b}_{kj}])=\bigvee_{1\le k\le m}([\underline{a}_{ik}\wedge\underline{b}_{kj},\overline{a}_{ik}\wedge\overline{b}_{kj}])=$$

$$[\bigvee_{1\le k\le m}(\underline{a}_{ik}\wedge\underline{b}_{kj}),\bigvee_{1\le k\le m}(\overline{a}_{ik}\wedge\overline{b}_{kj})]=[(\underline{A}\circ\underline{B})_{ij},(\overline{A}\circ\overline{B})_{ij}],$$

which proves (12).

Example 3. *We use the above theorem for $m = 2$,*

$$[\underline{A}, \overline{A}] = \begin{bmatrix} [0.1, 0.2] & [0.3, 0.9] \\ [0, 1] & [0.1, 0.8] \end{bmatrix}, \ [\underline{B}, \overline{B}] = \begin{bmatrix} [0, 0.7] & [0.2, 0.5] \\ [0.9, 1] & [0.5, 0.6] \end{bmatrix},$$

and we get

$$[\underline{A}, \overline{A}] \circ [\underline{B}, \overline{B}] = \begin{bmatrix} [0.3, 0.9] & [0.3, 0.6] \\ [0.1, 0.8] & [0.1, 0.6] \end{bmatrix}.$$

According to [5] we obtain

Theorem 9. *For $[\underline{A}, \overline{A}], [\underline{B}, \overline{B}], [\underline{C}, \overline{C}] \in S^{m\times m}$ we have*
$\bullet [\underline{A}, \overline{A}] \circ ([\underline{B}, \overline{B}] \circ [\underline{C}, \overline{C}]) = ([\underline{A}, \overline{A}] \circ [\underline{B}, \overline{B}]) \circ [\underline{C}, \overline{C}]$, *(associativity)*
$\bullet [\underline{A}, \overline{A}] \le [\underline{B}, \overline{B}] \Rightarrow [\underline{A}, \overline{A}] \circ [\underline{C}, \overline{C}] \le [\underline{B}, \overline{B}] \circ [\underline{C}, \overline{C}]$, *(monotonicity)*
$\bullet [\underline{A}, \overline{A}] \circ ([\underline{B}, \overline{B}] \vee [\underline{C}, \overline{C}]) = ([\underline{A}, \overline{A}] \circ [\underline{B}, \overline{B}]) \vee ([\underline{A}, \overline{A}] \circ [\underline{C}, \overline{C}])$,
(distributivity over supremum)
$\bullet [\underline{A}, \overline{A}] \circ ([\underline{B}, \overline{B}] \wedge [\underline{C}, \overline{C}]) \le ([\underline{A}, \overline{A}] \circ [\underline{B}, \overline{B}]) \wedge ([\underline{A}, \overline{A}] \circ [\underline{C}, \overline{C}])$,
(sub-distributivity over infimum)
$\bullet [\underline{A}, \overline{A}] \circ [I, I] = [I, I] \circ [\underline{A}, \overline{A}] = [\underline{A}, \overline{A}]$ $(neutral\, element)$, *where*

$$[I, I]_{ij} = \begin{cases} [1, 1], i = j \\ [0, 0], i \ne j \end{cases}, \ i, j = 1, ..., m.$$

Corollary 1. $(S^{m\times m}, \circ)$ *is an ordered semigroup with identity $[I, I]$.*

In semigroups we can consider powers of its elements.

Definition 8 (cf.[16]). *The powers of a matrix $A \in L^{m\times m}$ we call*

$$A^1 = A, \ A^{n+1} = A^n \circ A, \ n \in \mathbb{N}.$$

The sequence (A^n) is convergent, if

$$\underset{k\in\mathbb{N}}{\exists} A^{k+1} = A^k.$$

Definition 9. *The index of a matrix A is the number*

$$k = k(A) = \min\{q \in \mathbb{N} : \underset{p>q}{\exists} \ (A^p = A^q)\}.$$

Definition 10 (cf.[7]). *Let $A, B \in [0, 1]^{m\times m}$, $\mathcal{F} = \{A, B\}$. A mixed power sequence of matrices A and B is the sequence*

$$F_1 = C_1, \ F_{n+1} = F_n C_{n+1}, \ n \in \mathbb{N},$$

where $C : \mathbb{N} \to \mathcal{F}$. This family $\mathcal{F}$ is called weakly convergent, if all the sequences (F_n) are convergent. Weakly convergent family is strongly convergent, if all the limit matrices are equal.

Lemma 1 (cf.[4]). *Let $\mathcal{F} = \{\underline{A}, \overline{A}\}$. If $[\underline{A}, \overline{A}] \in S^{m\times m}$, then for every sequence (F_n), $n \in \mathbb{N}$ we have inequalities*

$$\underset{k\in\mathbb{N}}{\forall}\ \underline{A}^k \le F_k \le \overline{A}^k.$$

Theorem 10. *Let $\mathcal{F} = \{\underline{A}, \overline{A}\}$. If $(\underline{A}^n), (\overline{A}^n)$ are convergent to the same limit, then (A^n) is convergent to the same limit for each $A \in [\underline{A}, \overline{A}]$.*
In particular, $\mathcal{F}$ is strongly convergent.

Proof. For each $A \in [\underline{A}, \overline{A}]$ we have

$$\underline{A} \le A \le \overline{A}.$$

From Lemma 1 and Theorem 9 (monotonicity) we have inequalities

$$\underline{A}^k \le A^k \le \overline{A}^k,\ for\, k \in \mathbb{N},$$

so (A^n) is convergent by the Squeeze Law.

Definition 11. *The matrix $A \in [0,1]^{m\times m}$ is nilpotent, iff*

$$\underset{p\in\mathbb{N}}{\exists}\ A^P = [0].$$

From Lemma 1 and the above theorem we get

Theorem 11. *If matrix $\overline{A}^n$ is nilpotent, then sequence $([\underline{A}, \overline{A}]^n)$ is convergent.*

Now we consider the interval matrix with two different limit matrices for $(\underline{A}^n)$, $(\overline{A}^n)$. From Theorem 8 we have

$$[\underline{A}, \overline{A}]^n = [\underline{A}^n, \overline{A}^n], \tag{14}$$

so we obtain

Corollary 2. *Let $[\underline{A}, \overline{A}] \in S^{m\times m}$. If $(\underline{A}^n)$, $(\overline{A}^n)$ are convergent with indexes $k_1(\underline{A})$, $k_2(\overline{A})$ respectively, then $([\underline{A}, \overline{A}]^n)$ is convergent with $k = \max(k_1(\underline{A}),\ k_2(\overline{A}))$.*

Proof. From assumption we have:

$$\underline{A}^{k_1(\underline{A})+1} = \underline{A}^{k_1(\underline{A})},\ \ \overline{A}^{k_2(\overline{A})+1} = \overline{A}^{k_2(\overline{A})}.$$

Putting $k = \max(k_1(\underline{A}),\ k_2(\overline{A}))$ in (14) we get

$$[\underline{A}, \overline{A}]^{k+1} = [\underline{A}^{k+1}, \overline{A}^{k+1}] = [\underline{A}^k, \overline{A}^k] = [\underline{A}, \overline{A}]^k,$$

which proves that k is the index of considered interval matrix.

Example 4. *This example presents a weakly convergent family of matrices (with different limit matrices). Let* $m = 2$,

$$\underline{A} = \begin{bmatrix} 0.1 & 0.3 \\ 0 & 0.1 \end{bmatrix}, \ \overline{A} = \begin{bmatrix} 0.2 & 0.4 \\ 0.1 & 0.3 \end{bmatrix},$$

we have $\underline{A}^2 = \underline{A}^3 \neq \overline{A}^2 = \overline{A}^3$, *where*

$$(\underline{A})^2 = \begin{bmatrix} 0.1 & 0.1 \\ 0 & 0.1 \end{bmatrix}, \ (\overline{A})^2 = \begin{bmatrix} 0.2 & 0.3 \\ 0.1 & 0.3 \end{bmatrix},$$

$$\underline{A}\overline{A} = \begin{bmatrix} 0.1 & 0.3 \\ 0.1 & 0.1 \end{bmatrix}, \ \underline{A}\overline{A}\underline{A} = \begin{bmatrix} 0.1 & 0.1 \\ 0.1 & 0.1 \end{bmatrix}, \ \overline{A}\underline{A} = \begin{bmatrix} 0.1 & 0.2 \\ 0.1 & 0.1 \end{bmatrix}.$$

The family $\mathcal{F} = \{\underline{A}, \overline{A}\}$ *is weakly convergent with the limit matrices:* $\underline{A}^2$, $\overline{A}^2$, $\underline{A}\overline{A}$, $\underline{A}\overline{A}\underline{A} = \overline{A}\underline{A}^2$, $\overline{A}\underline{A}$. *From the above corollary we know, that* $([\underline{A}, \overline{A}]^n)$ *is also convergent with* $k = 2$, *because*

$$([\underline{A}, \overline{A}])^2 = \begin{bmatrix} [0.1,\, 0.2] & [0.1,\, 0.3] \\ [0,\, 0.1] & [0.1,\, 0.3] \end{bmatrix} = ([\underline{A}, \overline{A}])^3.$$

5 Conclusions

Our results show some facts connected with convergence of powers of interval matrix. Estimations of index of the interval matrix and construction of the approximation operation (the transitive closure operation) remain open problems. Interval valued fuzzy sets are, in general, extensions of fuzzy sets. Moreover, fuzzy sets theory and rough set theory are complementary. Thus, rough set methods can be used to define fuzzy concepts of approximation operations.

References

1. Atanassov, K.T.: Intuitionistic fuzzy sets, Theory and applications. Springer, Heidelberg (1999)
2. Białas, S.: Matrices Selected problems (Polish) AGH, Krakow (2006)
3. Birkhoff, G.: Lattice theory, vol. 25. AMS Coll. Publ., Providence (1967)
4. Drewniak, J., Pękala, B.: Multiple max-min products of square matrices. In: Atanassov, K.T. et al. (eds.) Soft Computing. Foundations and Theoretical Aspects, EXIT, Warszawa, pp. 171–180 (2004)
5. Give'on, Y.: Lattice matrices. Inform. Control 7, 477–484 (1964)
6. Gorzałczany, M.B.: A method of inference in approximate reasoning based on interval-valued fuzzy sets. Fuzzy Sets Syst. 21, 1–17 (1987)
7. Guu, S.M., Chen, H.H., Pang, C.T.: Convergence of products of fuzzy matrices. Fuzzy Sets Syst. 121, 203–207 (2001)
8. Kahl, P., Kreinovich, V., Lakeyev, A., Rohn, J.: Computational complexity and feasibility of data processing and interval computations. Kluwer Academic Publishers, Dordrecht (1998)

9. Mayer, G.: On the convergence of powers of interval matrices. Linear Algebra Appl. 58, 201–216 (1984)
10. Mayer, G., Arndt, H.-R.: On the semi-convergence of interval matrices. Linear Algebra Appl. 393, 15–37 (2004)
11. Mayer, G., Arndt, H.-R.: New criteria for the semiconvergence of interval matrices. SIAM J. Matrix Anal. Appl. 27(3), 689–711 (2005)
12. Moore, R.E.: Interval analysis. Prentice Hall, Englewood Cliffs (1966)
13. Moore, R.E.: Methods and applications of interval analysis. SIAM, Philadelphia (1979)
14. Negoiţă, C.V., Ralescu, D.A.: Applications of fuzzy sets to systems analysis. Birkhäuser Verlag, Basel (1975)
15. Niewiadomski, A.: Interval-valued linguistic variables. An application to linguistic summaries. In: Hryniewicz, O. et al. (eds.) Issues in Intelligent Systems. Paradigms, EXIT, Warszawa, pp. 167–183 (2005)
16. Thomason, M.G.: Convergence of powers of a fuzzy matrix. J. Math. Anal. Appl. 57, 476–480 (1977)
17. Turksen, I.B.: Interval-valued fuzzy sets based on normal forms. Fuzzy Sets Syst. 20, 191–210 (1986)
18. Young, R.C.: The algebra of many-valued quantities. Math. Ann. 104, 260–290 (1931)

The Diffie–Hellman Problem in Lie Algebras

Beata Rafalska

University of Warmia and Mazury
Olsztyn, Poland
winka@ols.vectranet.pl

Abstract. Cryptography in its present state relies increasingly on complex mathematical theories, e.g., elliptic curves, group theory, etc. We address in this article the problem of proxy signatures and we set this problem in the framework of Lie algebras. We show how to use a chosen maximal set of differentiable automorphisms in order to carry out the task of proxy signing. We also show possible attacks and the way to protect against them.

Keywords: cryptography, proxy signatures, attacks, Lie algebras, differentiable maps.

To the memory of Professor Zdzisław Pawlak

1 Introduction

1.1 Lie Algebras

Definition 1. *Let* **K** *be a commutative field. We say that a linear space* L *over* **K** *is a Lie algebra, if there is a bilinear operation*

$$L \times L \ni (a,b) \to [a,b] \in L,$$

called the Lie bracket (Lie product), satisfying the conditions:

(a) $[a,b] = -[b,a]$, *(anty-symmetry)*
(b) $\Big[a,[b,c]\Big] + \Big[b,[c,a]\Big] + \Big[c,[a,b]\Big] = 0.$

Condition (b) is called the Jacobi identity.

Obviously, condition (a) can be rewritten in an equivalent form: $[a,a] = 0$.

Fact 1. The set of all endomorphisms $End(X)$, where X is a linear space, is a Lie algebra with bracket defined as:

$$[f,g] = fg - gf,$$

for $f, g \in End(X)$.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 622–629, 2007.

Definition 2. *Let A be a linear algebra over* $\mathbf{K}$, *with an operation $A \times A \ni (a, b) \to a \star b \in A$. An endomorphism $\alpha : A \to A$ is called a differentiation, if for $a, b \in A$*

$$\alpha(a \star b) = \alpha(a) \star b + a \star \alpha(b).$$

The set of all differentiations over the algebra A is denoted as $Der(A)$.

Theorem 1. *$Der(A)$ is a Lie subalgebra of $End(A)$.*

Proof. Proof of this theorem is a simple algebraic computing.

Now, we consider some maximal set of pair-wise commuting differentiations and we denote it as $CDer(A)$, clearly, such a set is non-unique.

Fact 2. There exists a non-empty set $CDer(A)$.

Proof. Let $\alpha \in Der(A)$. Then α is commutative with α. Composite $\alpha\alpha \in Der(A)$, but α is commutative with $\alpha\alpha$ too, etc. So that $\alpha, \alpha\alpha, \ldots, \alpha\alpha\ldots\alpha$ are pair-wise commuting, and the set $\{\alpha^n : n = 1, 2, \ldots\}$ extends to a maximal set $CDer(A)$.

Fact 1.2 shows that a set $CDer(A)$ exists.

Theorem 2. *$CDer(A)$ is an algebra.*

Proof. Proof of this theorem is a simple algebraic computing.

1.2 The Diffie-Hellman Problem

Let us recall the definition of discrete logarithm from [2].

Let $\mathbb{F}_p^* = (\mathbb{Z}/p\mathbb{Z})^* = \{1, 2, \cdots, p-1\}$ be the multiplicative group of integer numbers modulo a prime number p. Let $g \in \mathbb{F}_p^*$ be a fixed element. *The discrete logarithm problem* in $\mathbb{F}_p^*$ at the base g is the problem of finding for the fixed $y \in \mathbb{F}_p^*$ of a natural number x, such that $y = g^x$ modulo p.

We remind now the Diffie-Hellman key exchange system (see [2]). Assume, that Alice and Bob want to agree on a secret key in any cryptosystem with private keys. Keys exchange occurs over an insecure communication channel, so that an adversary Charlie knows the substance of all communicates, which are sent between Alice and Bob. Alice and Bob agree at first on a large prime number p and a base g. Then Alice in secret picks a random natural number $k_A < p$ (of the same order as p) and computes the remainder from division of g^{k_A} by p and the result is sent to Bob. Bob proceeds in a similar manner and sends to Alice $g^{k_B} \in \mathbb{F}_p^*$ keeping k_B secret. The key agreed upon will be the number $g^{k_A k_B}$. The problem which Charlie is facing, is the *Diffie-Hellman problem:* having the data $g, g^{k_A}, g^{k_B} \in \mathbb{F}_p^*$, compute $g^{k_A k_B}$. It is worth to notice, that everyone who can solve the discrete logarithm problem, can solve the Diffie-Hellman problem, too.

In [1], the author generalizes the discrete logarithm problem and the Diffie-Hellman problem to cyclic groups. We define the *general discrete logarithm problem* as follows: Let $G =< a_1, a_2, \cdots, a_n >$ be a cyclic group and $f : G \to G$

be a non identity automorphism. General discrete logarithm problem is to find $f(b)$ for any $b \in G$ having given $f(a)$ for some $a \in G$. In other words the general discrete logarithm problem is to find the automorphism f knowing its action on only one element.

Suppose now, that we have two non identity automorphisms $\varphi, \psi : G \to G$ and that we know $a, \varphi(a)$ i $\psi(a)$. Then, the *general Diffie-Hellman problem* is to find $\varphi(\psi(a))$.

2 Diffie-Hellman Problem in Lie Algebras

2.1 Key Exchange System

Alice and Bob want to agree on a private key for exchange of information over an insecure channel. They agree on a Lie algebra L, a set $CDer(A)$, and an element $g \in L$. Alice picks randomly a differentiation $\alpha \in CDer(L)$, and an element $a \in L$. She sends Bob the value $\alpha([g,a])$. Bob picks at random a differentiation $\beta \in CDer(L)$, and he sends to Alice the value $\beta(\alpha([g,a]))$. Alice determines α^{-1} and computes $\beta([g,a])$:

$$\alpha^{-1}(\beta(\alpha([g,a]))) = \alpha^{-1}(\alpha(\beta([g,a]))) = \alpha^{-1}\alpha(\beta([g,a])) = \beta([g,a])$$

Now, Alice randomly chooses a next differentiation $\gamma \in CDer(L)$, and computes $\gamma(\beta([g,a]))$ and then the result is sent to Bob. Alice can compute $\gamma([g,a])$ too, and Bob, knowing the differentiation β, computes β^{-1} and finds $\gamma([g,a])$, (in analogy to Alice's computation). The value $\gamma([g,a])$ is their fixed key.

2.2 System Analysis

Notice, that Alice doesn't show a, so the adversary Charlie knows $L, g, \alpha([g,a])$, $\beta(\alpha([g,a]))$ and $\gamma(\beta([g,a]))$. To find $\gamma([g,a])$ is a problem which incorporates the general discrete logarithm problem with the general Diffie-Hellman problem described in [2]. We can restrict the problem to finding a differentiation β (exactly β^{-1}) having as information $L, g, \alpha([g,a]), \beta(\alpha([g,a]))$ and $\gamma(\beta([g,a]))$. The task of finding β, can be reduced to computation α. Finally, the problem of finding the key can be reduced to computing of the differentiation α knowing only the action of α on one element $[g,a]$. An additional impediment for Charlie is the fact, that he doesn't know the element a, which Alice doesn't show.

2.3 Sending Information

For simplifying of notation, we mark the earlier fixed key as $x = \gamma([g,a])$. Suppose, that Alice wants to send to Bob an information m already converted to an element from the Lie algebra L. Alice sends to Bob the element $y = [m,x]$. Bob knows the Lie algebra L, so he knows the Lie bracket and key x and he can compute the value m. The difficulty of this computation will depend on the specified Lie bracket and the internal multiplication in the algebra. For an example, for Lie algebra with the standard commutator and anti-commutative internal multiplication, we have:

$$[m, x] = y$$

$$mx - xm = y$$

$$2mx = y$$

$$m = 2^{-1}yx^{-1}.$$

Adversary Charlie doesn't know the key x, so he can't decode the information m. Further, if Alice computes the information hash $H(m)$, and she sends to Bob the algebra element $z = [H(m), x]$, too, then Bob will be able to verify, whether he gets the information in an unspoiled form. Analogically, to decipher the information m, Bob will decifer the hashed message $H(m)$, and he compares whether what he has got is the same as the element $H(m)$, which he got from Alice. So, for increased security, Alice sends to Bob the pair (y, z).

3 Information Signature

In our algorithm, we can use the signature scheme with proxy signers, analogical to scheme described in [4].

3.1 Notation

We mark original signer as P_0 and proxy signers as $\{P_1, P_2, \cdots, P_m\}$. We suppose, that all signers P_i have private keys a_i and the corresponding public keys $A_i = [a_i, g]$, where g is Lie algebra's element, certified by the central authority for $i = 0, \cdots, m$. Let w be a message created by the original signer P_0. Moreover, we will assume that H i H_1 are some suitably chosen collision -free hash functions.

3.2 Group Secret Key Generation

P_0 prepares the information w, and chooses randomly an algebra element r and computes $R = [r, g]$. Next, he determines the value $H = H(w, R)$ of the collision-free hash function H. Having this data, P_0 computes the group secret key,

$$d = [a_o, H] + r.$$

We notice, that $d = d(R)$ and $d = d(a_0)$, where a_0 is a private key of P_0, thus only P_0 can compute d, moreover, the private key a_0 of signer P_0 is well protected by randomly behaving hash function H. The public verification that the signature is true is not difficult, too, because we have publicly known R and A_0 :

$$d = [a_0, H] + r \longrightarrow [d, g] = [[a_0, H], g] + [r, g] = [A_0, H] + R$$

$$[d, g] = [[a_0, H] + r, g] = [a_0H - Ha_0, g] + [r, g] = [a_0H, g] - [Ha_0, g] + R$$
$$= [a_0, g]H - H[a_0, g] + R = [[a_0, g], H] + R = [A_0, H] + R.$$

3.3 Group Secret Key Share

The original signer P_0 selects a polynomial

$$f_0(x) = c_{0(t-1)}x^{(t-1)} + \cdots + c_{01}x + d,$$

where each c_i for $i = 1, \cdots, t-1$ is a random algebra element. We see, that Lie multiplication of any element by itself results in 0, so we specify power as internal multiplication in algebra. Next, P_0 computes $C_{0i} = [c_{0i}, g]$ for $i = 1, \cdots, t-1$, and he sends it to proxy signers P_i.

$$Transfer : (\{C_{0i} = [c_{0i}, g] : i = 1, \cdots, t-1\}),$$

so that,

$$\begin{array}{lllll} f_0(x) = & c_{0(t-1)}x^{(t-1}+ & \cdots + & c_{01}x+ & d \\ & \downarrow & & \downarrow & \downarrow = a_0H + r \\ & C_{0(t-1)} & & C_{01} & HA_0 + R \end{array}$$

Let x_i be the public identity of P_i. Now, P_0 distributes the secret key $d_0 = f_0(0)$ distributing the values $y_{i0} = f_0(x_i)$ for each $P_i \in P$, and he sends them by secret channels.

$$P_0 \longmapsto f_0(x) = \sum_{i=1}^{t-1} c_{0i}x^i + d \qquad d = a_0H + r.$$

Each proxy signer can verify y_{i0} by the equation

$$[y_{i0}, g] = \underbrace{A_0H + R}_{[d,g]} + \sum_{j=1}^{t-1} x_i^j C_{0j}.$$

3.4 The Proxy Signature Generation

Now, each proxy signer P_i selects a secret polynomial

$$f_i(x) = c_{i(t-1)}x^{t-1} + \cdots + c_{i1}x + c_{i0} + a_i,$$

where c_{ik} for $k = 1, \cdots, t-1$ is a random Lie algebra's element and a_i is a secret key of P_i. Next, P_i computes and broadcasts $C_{ik} = [c_{ik}, g]$, for $k = 0, \cdots, t-1$.

$$Transfer : (C_{ik} : \{k = 0, \cdots, t-1\}, A_i)$$

P_i computes the value of the hash function $H_1 = H_1(w, R, M, B)$ too, where M is a message, which P_i wants to sign on behalf of the original signer P_0 and B is any subset of t (or more) proxy signers, and he computes the value $f_i(x_j)$ for $i \neq j$ and sends to P_j by the secret channel his part

$$y_{ji} := H_1 f_i(x_j), \qquad j = 1, \cdots t$$

$$P_i : y_{ji} = H_1 f_i(x_j) \longrightarrow P_j \qquad \forall j \neq i, P_j \in B.$$

Next, each signer P_j verifies the received values $f_i(x_j)$ from other $t-1$ proxy signers by the equation

$$[y_{ji}, g] = H_1(A_i + \sum_{k=0}^{t-1} x_j^k C_{ik}), \qquad \forall j \neq i, P_i \in B.$$

If all of the above equation hold, then each P_j computes his partial proxy signature from the received values as $s_j = \sum\limits_{i=1}^{t} y_{ji}$.

$$P_1 : s_1 = \sum_{i=1}^{t} y_{1i} = \sum_{i=1}^{t} H_1 f_i(x_1) = H_1 f_1(x_1) + H_1 f_2(x_1) + \ldots + H_1 f_t(x_1) = H_1 f(x_1)$$

$$P_2 : s_2 = \sum_{i=1}^{t} y_{2i} = \sum_{i=1}^{t} H_1 f_i(x_2) = H_1 f_1(x_2) + H_1 f_2(x_2) + \ldots + H_1 f_t(x_2) = H_1 f(x_2)$$

$$\vdots$$

$$P_t : s_t = \sum_{i=1}^{t} y_{ti} = \sum_{i=1}^{t} H_1 f_i(x_t) = H_1 f_1(x_t) + H_1 f_2(x_t) + \ldots + H_1 f_t(x_t) = H_1 f(x_t)$$

This share has a value $H_1 f(x_j)$, where $f(x)$ is the virtual polynomial

$$\begin{array}{ccccccc} f(x) = & x^{t-1}(\sum\limits_{i=1}^{t} c_{i(t-1)}) + & \cdots + & (\sum\limits_{i=1}^{t} c_{i0}) + & (\sum\limits_{i=0}^{t} a_{i0}) \\ \downarrow & \downarrow & & \downarrow & \downarrow \\ F(x) = & x^{t-1}\,\underbrace{(\sum_{i=1}^{t} C_{i(t-1)})}_{:= C'_{t-1}} + & \cdots + & \underbrace{(\sum_{i=1}^{t} C_{i0})}_{:= C'_0} + & \underbrace{(\sum_{i=1}^{t})A_i}_{:= A'} \end{array}$$

The public obligations of signers group B are

$$Transfer : (\{C'_k : k = 0, 1, \cdots, t-1\}, A')$$

Next, each proxy signer P_j computes the threshold proxy signature on M as follows:

$$\sigma_j = \sigma_j(M, B, w, R) = y_{j0} + \sum_{i=1}^{t} y_{ji} = y_{j0} + s_j.$$

P_j sends by the secret channel the calculated σ_j for each $P_i \in B$. Now, each P_j does the test of the received shares:

$$[\sigma_j, g] = [y_{j0} + s_j, g] = [y_{j0} + \sum_{i=1}^{t} y_{ji}, g] = [y_{j0}, g] + [\sum_{i=1}^{t} y_{ji}, g]$$

$$= HA_0 + R + \sum_{k=1}^{t-1} x_j^k C_{0k} + \sum_{j=1}^{t} H_1(A_j + \sum_{k=0}^{t-1} x_j^k C_{jk}).$$

Finally, if all was correct, then the threshold proxy signature is the following: $(M, C'_0, A', \sigma, w, B)$, where $\sigma = d + H_1 f(0)$.

3.5 Verification of the Proxy Signature

Addressee of the message in the first step does verify correctness of the threshold (M, C'_0, A', w, B) by the verifying equation:

$$[\sigma, g] = [d + H_1 f(0), g] = [d, g] + [H_1 f(0), g] = [f_0(0), g] + [H_1 f(0), g].$$

If this equation is true, then the addressee infers, that the proxy signature $(M, C'_0, A', \sigma, w, B)$ is the proper proxy signature obtained from the delegation key of the original signer and that the set B consists of the actual proxy signers.

Next, the addressee computes:

$$f_0(0) = d = a_0 H + r,$$

$$[f_0(0), g] = [d, g] = A_0 H + R,$$

and

$$[H_1 f(0), g] = [H_1(w, R, M, B) f(0), g] = [H_1(\sum_{i=1}^{t} f_i)(0), g]$$

$$= [H_1 \sum_{i=1}^{t} c_{i0}, g] = H_1 \sum_{i=1}^{t} [c_{i0}, g] = H_1 \sum_{i=1}^{t} C_{i0} = H_1 F(0).$$

4 The Analysis of the Insider Attack

Suppose, that one from the pair proxy signer - insider attacker (without the loss of the generality, we agree that this is P_1) wants to get a threshold proxy signature on message M. While generating the proxy signature, P_1 does not broadcast his data C_{1k}, but he waits until will receives from remaining proxy signers their data C_{i1}. Now, P_1 computes the hash function $H_1 = H_1(w, R, M, B)$ and assign $f_1(x_j)$ for $i \neq j$. P_1 can compute $y_{j1} = H_1 f_1(x_j)$, but he can't compute s_1 which is indispensable to falsify the threshold proxy signature, because he does not know all y_{ji}. So, this attack isn't practical, let us suppose that proxy signers don't continue the broadcast of the data until they receive earlier obligations from all signers. Let's see now, what happens,when the insider attacks on the later transfer of the data, i.e. just during sending y_{ji}. Then the scheme generating the proxy signature would look as follows:

Each proxy signer P_i selects the secret polynomial $f_i(x) = c_{i(t-1)} x^{t-1} + \cdots + c_{i1} x + c_{i0} + a_i$, and they compute and broadcast $C_{ik} = [c_{ik}, g]$ for $k = 0, \cdots, t-1$. Later P_i computes the value of the hash function $H_1 = H_1(w, R, M, B)$ and determines $f_i(x_j)$ for $i \neq j$. At this moment, P_1 attacks and he doesn't broadcast his value y_{ji} but waits for values from remaining signers. In this way P_1 receives all values y_{1i} for $i \neq 1$ and he computes y_{11}. In this situation, after the verification of the data, P_1 can compute his part of the threshold proxy signature

$$s_1 = \sum_{i=1}^{t} y_{1i}.$$

So now P_1 computed his part of the threshold proxy signature s_1, but he has not broadcasted his data to remaining signers. Theoretically P_1 can privately compute y_1' and for this value the likely value s_1', so the threshold proxy signature is correct for y_1', however the counterfeited value y_1' will not pass verification conducts by remaining signers, because P_1 can not alter the sent earlier C_{1k}.

Finally, we see, that the scheme of the proxy signature is resistant to the attack by any insider signer if we suppose that proxy signers will not send data, until they not receive earlier obligations.

5 Conclusion

We have presented an approach to the Diffie-Hellman problem in Lie algebras, by exploiting sets of commutative differentiations. Our results generalize in a sense the approach in [1].

References

1. Mahalanobis, A.: Diffie-Hellman key exchange protocol and non-abelian nilpotent groups (2005) http://eprint.iacr.org/2005/110.ps
2. Koblitz, N.: Algebraiczne aspekty kryptografii, WNT, Warszawa (2000)
3. Wojtyñski, W.: Grupy i algebry Liego. PWN, Warszawa (1986)
4. Pomykaa, J., Barabasz, S.: Eliptic Curve Based Threshold Proxy Signature Scheme with Known Signers. Fundamenta Informaticae 69(4), 411–425 (2006)
5. Goldwasser, S., Bellare, M.: Lecture Notes on Cryptography (1996) http://www-cse.ucsd.edu/~mihir/papers/gb.html
6. Desmedt, I.: Some Recent Research Aspect of Threshold Cryptography, Departament of Mathematics Royal Holloway University of London (1997) http://citeseer.ist.psu.edu/desmedt97some.html
7. Zhang, K.: Threshold Proxy Signature Schemes, Cambridge University Computer Laboratory (1997) http://citeseer.ist.psu.edu/zhang97threshold.html

Software Defect Classification: A Comparative Study with Rough Hybrid Approaches

Sheela Ramanna[1], Rajen Bhatt[2], and Piotr Biernot[1]

[1] Department of Applied Computer Science, University of Winnipeg,
Winnipeg, Manitoba R3B 2E9 Canada
s.ramanna@uwinnipeg.ca, pbiernot@iam.uwinnipeg.ca
[2] Samsung India Software Center,
Noida-201305, Uttar Pradesh, India
rajen.bhatt@gmail.com

Abstract. This paper is an extension of our earlier work in combining strengths of rough set theory and neuro-fuzzy decision trees in classifying software defect data. The extension includes the application of a rough-fuzzy classification trees to classifying defects. We compare classification results for five methods: rough sets, neuro-fuzzy decision trees, partial decision trees, rough-neuro-fuzzy decision trees and rough-fuzzy classification trees. The analysis of the results include a paired t-test for accuracy and number of rules. The results demonstrate that there is improvement in classification accuracy with the rough fuzzy classification trees with a minimal set of rules. The contribution of this paper is a comparative study of several hybrid approaches in classifying software defect data.

Keywords: Classification, rough-fuzzy classification trees, neuro-fuzzy decision trees, rough sets, software defects.

1 Introduction

The paper is an extension of our earlier work in combining strengths of rough set theory and neuro-fuzzy decision trees in classifying software defect data [16]. The extension includes the application of a rough-fuzzy classification trees to classifying defects. Specifically, in our previous work, the hybrid approach was limited to employing the strength of rough sets to attribute reduction as the first step in classification with neuro-fuzzy decision trees. In contrast, in this paper, we use the fuzzy-rough set proposed in [8] extended by dependency degree measure proposed in [2] to the classification problem. Other data mining methods reported in this paper are from rough set theory [13], fuzzy decision trees [20] and neuro-fuzzy decision trees [1].

In the context of software defect classification, the term data mining refers to knowledge-discovery methods used to find relationships among defect data and the extraction of rules useful in making decisions about defective modules either during development or during post-deployment of a software system. A software

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 630–638, 2007.

defect is a product anomaly (e.g, omission of a required feature or imperfection in the software product) [15]. As a result, defects have a direct bearing on the quality of the software product and the allocation of project resources to program modules. Software metrics make it possible for software engineers to measure and predict quality of both the product and the process. In this work, the defect data consists of product metrics drawn from the PROMISE[1] Software Engineering Repository data set.

There have been several studies in applying computational intelligence techniques such as rough sets [14], fuzzy clustering [7,21], neural networks [11] to software quality data. Statistical predictive models correlate quality metrics to number of changes to the software. The predicted value is a numeric value that gives the number of changes (or defects) to each module. However, in practice, it is more useful to have information about modules that are highly defective rather than knowing the exact number of defects for each module.

The contribution of this paper is a comparative study of several hybrid approaches in classifying software defect data. The results demonstrate that there is improvement in classification accuracy with the rough fuzzy classification trees with a minimal set of rules.

This paper is organized as follows. In Sect. 2, we give a brief overview of two hybrid methods. The details of the defect data and classification methods are presented in Sect. 3. This is followed by an analysis of the classification results in Sect. 4.

2 Hybrid Approaches

Fuzzy-Rough Classification Trees (FRCT) integrate rule generation technique of fuzzy decision trees and rough sets. The measure used for the induction of FRCT is fuzzy-rough dependency degree proposed in [3,4]. Neuro-fuzzy decision trees (NFDT) include a fuzzy decision tree (FDT) structure with parameter adaptation strategy based on neural networks [1].

2.1 Fuzzy-Rough Classification Trees

In this section, we briefly outline the steps of computing the fuzzy-rough dependency degree. Let $\{F_{jk}|\ k = 1, \ldots, c_j\}$ be overlapping and non-empty partitions of real-valued attributes x_j $(1 \leq j \leq p)$ on the set of training set U. F_{jk} is the k^{th} fuzzy set of j^{th} attribute x_j. The lower approximation of an arbitrary class l of F_{jk} is given by:

$$\mu_{\underline{l}}\left(F_{jk}\right) = \inf_{\forall i \in U} \max\left\{1 - \mu_{F_{jk}}\left(x_j^i\right), \mu_l\left(y^i\right)\right\}$$

where x_j^i and y^i are i^{th} value of attribute x_j. Given an attribute x_j with c_j fuzzy partitions, dependency degree γ_{x_j} of y on attribute x_j can be calculated as follows:

[1] http://promise.site.uottawa.ca/SERepository

- Calculate the lower approximation member function $\mu_{\underline{l}}\left(F_{jk}\right)$ using the above definition
- Calculate fuzzy positive region $\mu_{POS}\left(F_{jk}\right) = \sup\limits_{l=1,..,q} \left\{\mu_{\underline{l}}\left(F_{jk}\right)\right\}$
- Calculate the i^{th} pattern to the fuzzy positive region $\mu_{POS}\left(x_j^i\right) = \sup\limits_{l=1,..,q} \min\left\{\mu_{F_{jk}}\left(x_j^i\right), \mu_{POS}\left(F_{jk}\right)\right\}$
- Calculate the dependency degree $\gamma_{x_j}(y) = \frac{\sum_{i=1}^{n} \mu_{POS}\left(x_j^i\right)}{n}$

Given fuzzy partitions of feature space, leaf selection threshold β_{th}, and fuzzy-rough dependency degree γ as expanded attribute (attribute to represent each node in fuzzy decision tree) selection criterion, the general procedure for generating fuzzy decision trees using FRCT algorithm is outlined in Alg.1.

Algorithm 1. Algorithm for generating fuzzy decision trees using FRCT

```
Input  : fuzzy partitions of feature space, $\beta_{th}$, $\gamma$
Output: fuzzy decision trees
while ∃ candidate nodes do
    select node with highest $\gamma$; // dependency degree
    Generate its child-nodes;
    //root node will contain attribute with highest $\gamma$
    if ($\beta_{child-node} \geq= \beta_{th}$) then
        childnode = leafnode
    else
        search continues with child-node as new root node
    end
end
```

Before training the initial data, the α cut is usually used for the initial data [4]. Usually, α is in the interval $(0, 0.5]$. A detailed description of fuzzy-rough dependency degree is available in [3]. The cut of a fuzzy set A is defined as:

$$\mu_{A_\alpha}(a) = \begin{cases} \mu_A(a); \mu_A(a) \geq \alpha \\ 0; \mu_A(a) < \alpha \end{cases} .$$

2.2 Neuro-fuzzy Decision Trees

In the forward cycle, NFDT constructs a fuzzy decision tree using the standard FDT induction algorithm fuzzy ID3 [20]. In the feedback cycle, parameters of fuzzy decision trees (FDT) have been adapted using stochastic gradient descent algorithm by traversing back from each leaf to root nodes. During the parameter adaptation stage, NFDT retains the hierarchical structure of fuzzy decision trees. All the attributes have been fuzzified using fuzzy c-means algorithm [5] into three fuzzy clusters. From the clustered row data, Gaussian membership functions have been approximated by introducing the width control parameter λ. The center of each gaussian membership function has been initialized by fuzzy cluster centers generated by the fuzzy c-means algorithm. To initialize standard deviations, we have used a value proportional to the minimum distance between centers of fuzzy clusters. For each numerical attribute x_j and for each gaussian membership function, the Euclidean distance between the center of F_{jk} and the center of any other membership function F_{jh} is given by $dc\left(c_{jk}, c_{jh}\right)$, where $h \neq k$. For each k^{th} membership function, after calculating $dc_{\min}\left(c_{jk}, c_{jh}\right)$, the standard deviation σ_{jk} has been obtained by (1)

$$\sigma_{jk} = \lambda \times dc_{\min}\left(c_{jk}, c_{jh}\right); 0 < \lambda \leq 1, \tag{1}$$

where λ is the width control parameter. For the computational experiments reported here, we have selected various values of $\lambda \in (0, 1]$ to introduce variations in the standard deviations of initial fuzzy partitions. After attribute fuzzification, we run the fuzzy ID3 algorithm with cut $\alpha = 0$ and leaf selection threshold $\beta_{th} = 0.75$. These fuzzy decision trees have been tuned using the NFDT algorithm for 500 epochs with the target MSE value 0.001.

3 Software Defect Data

The PROMISE data set includes a set of static software metrics about the *product* as a predictor of defects in the software. There are a total of 94 attributes and one decision attribute (indicator of defect level). The defect level attribute value is TRUE if the class contains one or more defects and FALSE otherwise. The metrics at the *method level* are primarily drawn from Halstead's Software Science metrics [9] and McCabe's Complexity metrics [12]. The metrics at the *class level*, include such standard measurements as Weighted Methods per Class (WMC), Depth of Inheritance Tree (DIT), Number of Children (NOC), Response For a Class (RFC), Coupling Between Object Classes (CBO), and Lack of Cohesion of Methods (LCOM) [6]. The data includes measurements for 145 modules (objects).

Experiments reported were performed with RSES[2] using rule-based and tree-based methods. The RSES tool is based on rough set methods. Only the rule-based method which uses genetic algorithms in rule derivation [19] is reported in this paper. Experiments with non-rough set based methods were performed with WEKA[3] using a partial decision tree-based method (DT) which is a variant of the well-known C4.5 revision 8 algorithm [17]. The experiments were conducted using 10-fold cross-validation technique. The accuracy results with ROSE (another rough-set based tool) using a basic minimal covering algorithm was 79%. However, since ROSE[4] uses an internal 10-fold cross-validation technique, we have not included the experimental results in our pair-wise t-statistic test. The attributes were discretized in the case of rough set methods. The Rough-NFDT method included i) generating reducts from rough set methods ii) using the data from the reduced set of attributes to run the NFDT algorithm.

[2] http://logic.mimuw.edu.pl/∼rses

[3] http://www.cs.waikato.ac.nz/ml/weka

[4] http://idss.cs.put.poznan.pl/site/rose.html

4 Analysis of Classification Results

Tables 1 and 2 give a summary of computational experiments using five methods. Percentage classification accuracy has been calculated by $\frac{n_c}{n} \times 100\%$, where n is the total number of test patterns, and n_c is the number of test patterns classified correctly. A comparison of pairs of differences in classification accuracy and number of rules using the 10-fold cross-validated paired t-test is also discussed in this section.

We want to test the hypothesis that mean difference in accuracy or number of rules between any two classification learning algorithms is zero. Let μ_d denote

Table 1. Defect Data Classification I

10CV Accuracy% Results					
Run	NFDT	R-NFDT	Rough	FRCT	DT
1	85.71	71.42	92.9	85.71	71.43
2	85.71	92.85	78.6	85.71	85.71
3	64.28	67.58	57.1	71.42	64.29
4	71.42	71.42	57.1	92.85	71.43
5	64.28	57.14	50	71.42	71.43
6	78.57	78.57	78.6	78.57	64.29
7	85.71	71.42	71.4	85.71	85.71
8	71.42	78.57	71.4	71.42	57.14
9	92.85	100	78.6	92.85	78.57
10	89.47	89.47	84.2	89.47	84.21
Avg.Acc	78.94	77.84	71.99	82.51	73.42

Table 2. Defect Data Classification II

10CV Results - Number of Rules					
Run	NFDT	R-NFDT	Rough	FRCT	DT
1	3	2	231	7	9
2	14	18	399	5	12
3	4	3	330	4	8
4	2	5	282	9	8
5	6	1	308	5	8
6	6	4	249	7	7
7	6	7	124	3	12
8	2	6	292	3	10
9	10	7	351	9	14
10	4	4	235	7	9
Avg.#ofrules	5.7	5.7	280	5.7	9.7

the mean difference in accuracy during a 10-fold classification of software defect data. Let H0 denote the hypothesis to be tested (i.e., $H0 : \mu_d = 0$). This is our null hypothesis. The paired difference t-test is used to test this hypothesis and its alternative hypothesis ($HA : \mu_d \neq 0$). Let $\overline{d}$, S_d^2 denote the mean difference and variance in the error rates of a random sample of size n from a normal distribution N(μ_d, σ^2), where μ_d and σ^2 are both unknown. The t statistic used to test the null hypothesis is as follows:

$$t = \frac{\bar{d} - \mu_d}{S_d/\sqrt{n}} = \frac{\bar{d} - 0}{S_d/\sqrt{n}} = \frac{\bar{d}\sqrt{n}}{S_d}$$

where t has a student's t-distribution with n-1 degrees of freedom [10]. In our case, $n - 1 = 9$ relative to 10 sample error rates. The significance level α of the test of the null hypothesis H0 is the probability of rejecting H0 when H0 is true (called a Type I error). Let t_{n-1}, $\alpha/2$ denote a t-value to right of which lies $\alpha/2$ of the area under the curve of the t-distribution that has n-1 degrees of freedom. Next, formulate the following decision rule with $\alpha/2 = 0.025$:

Decision Rule: Reject $H0 : \mu_d = 0$ at significance level α if, and only if $|t - value| < 2.262$

Table 3. T-test Results

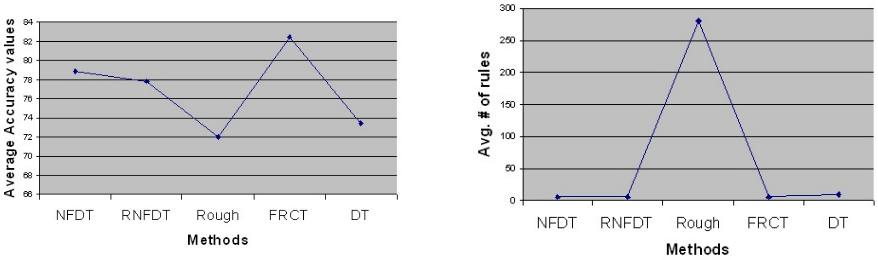
	Accuracy			Number of Rules		
Pairs	Avg. Diff.	Std. Dev.	t-stat	Avg. Diff.	Std. Dev.	t-stat
R-NFDT/NFDT	−1.10	8.24	−0.42	0.00	3.02	0.00
R-NFDT/Rough	5.85	11.67	1.59	−274.00	74.36	−11.67
R-NFDT/DT	4.42	12.58	1.11	−4.00	3.83	−3.30
NFDT/Rough	6.95	7.57	2.91	−274.40	74.42	−11.66
NFDT/DT	5.52	8.09	2.16	−4.00	2.94	−4.30
Rough/DT	−1.43	14.22	−0.32	270	75.96	11.26
FRCT/R-NFDT	4.67	10.65	1.39	0.2	5.51	0.114
FRCT/NFDT	3.57	6.94	1.63	0.2	4.32	0.147
FRCT/Rough	10.52	12.35	2.69	−274.2	75.9	−11.42
FRCT/DT	9.09	7.65	3.76	−3.8	3.23	−3.73

Pr-values for t_{n-1}, $\alpha/2$ can obtained from a standard t-distribution table. In terms of the t-test for accuracy, in general the three hybrid methods (FRCT, R-NFDT and NFDT) are comparable in that there is no significant difference in any of the methods based on the null hypothesis. In contrast, there is a difference in accuracy between three methods outlined above(FRCT/Rough, FRCT/DT and NDFT/Rough). The same is true with the three hybrid methods in terms of the number of rules. The reason being that the average number of rules used by hybrid methods are similar and few. However, the genetic algorithm-based

Table 4. Null-Hypothesis Results

Accept H0 ($u_d = 0$) if $\|t\ value\| < 2.262$				
Pairs	t-stat(Acc.)	Acc/Rej H0	t-stat(Rules)	Acc/Rej H0
R-NFDT/NFDT	−0.42	*Accept*	0.00	*Accept*
R-NFDT/Rough	1.59	*Accept*	−11.67	*Reject*
R-NFDT/DT	1.11	*Accept*	−3.30	*Reject*
NFDT/Rough	2.91	*Reject*	−11.66	*Reject*
NFDT/DT	2.16	*Accept*	−4.30	*Reject*
Rough/DT	−0.32	*Accept*	11.26	*Reject*
FRCT/R-NFDT	1.39	*Accept*	0.12	*Accept*
FRCT/NFDT	1.63	*Accept*	0.147	*Accept*
FRCT/Rough	2.69	*Reject*	−11.42	*Reject*
FRCT/DT	3.76	*Reject*	−3.73	*Reject*

classifier in RSES induces a large set of rules. As a result, there is a significant difference when classifiers are compared with the rough classifier on the basis of number of rules.



(a) Average Accuracy Values

(b) Average Number of Rules

Fig. 1. Comparison of Results

The other result that is noteworthy is the actual average accuracy values for the five methods as shown in Fig. 1(a). There is some improvement with the FRCT method reported in this paper. Fig. 1(b) demonstrates the comparable number of rules in all methods except the rough set method.

The other important observation is the role that reducts play in defect data classification. On an average, only 10 attributes (out of 95) were used by the rough set method with no significant reduction in classification accuracy. The Rough-NFDT (hybrid) method uses the reduced set of attributes resulting in a minimal number of rules with comparable accuracy. The average number of attributes (over 10 runs) is about 4. The metrics that are most significant on the class-level include: DIT, RFC, CBO and LCOM. At the method level, the metrics that are most significant include: i) Halstead's metric of *content* where the complexity of a given algorithm independent of the language used to express the algorithm ii) Halstead's metric of *level* which is level at which the program can be understood iii) Halstead's metric of *number of unique operands* which includes variables and identifiers, constants (numeric literal or string) function names when used during calls iv) total lines of code v) $branch - count$ is the number of branches for each module. Branches are defined as those edges that exit from a decision node.

5 Conclusion

This paper presents several approaches to classification of software defect data using rough set algorithms, rough-fuzzy, neuro-fuzzy decision trees and partial decision tree methods. The t-test shows that there is no significant difference between any of the hybrid methods in terms of accuracy at the 95% confidence level. However, in terms of rules, there is a marked difference. The hybrid rough-fuzzy classification tree method appears to be an improvement over earlier methods.

Acknowledgements

The research of Sheela Ramanna and Piotr Biernot is supported by NSERC Canada grant 194376.

References

1. Bhatt, R.B., Gopal, M.: Neuro-fuzzy decision trees. International Journal of Neural Systems 16(1), 63–78 (2006)
2. Bhatt, R.B.: Fuzzy-Rough Approach to Pattern Classification- Hybrid Algorithms and Optimization, Ph.D. Dissertation, Electrical Engineering Department, Indian Institute of Technology Delhi, India (2006)
3. Bhatt, R.B., Gopal, M.: On the Extension of Functional Dependency Degree from Crisp to Fuzzy Partitions. Pattern Recognition Letters 27(5), 487–491 (2006)
4. Bhatt, R.B., Gopal, M.: Induction of Weighted and Interpretable Fuzzy Classification Rules for Medical Informatics. International Journal of Systemics, Cybernetics, and Informatics 3(1), 20–26 (2006)
5. Bezdek, J.C: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum, New York (1981)
6. Chidamber, S.R., Kemerer, F.C.: A metrics suite for object-oriented design. IEEE Trans. Soft. Eng. 20(6), 476–493 (1994)
7. Dick, S., Meeks, A., Last, M., Bunke, H., Kandel, A.: Data mining in software metrics databases. Fuzzy Sets and Systems 145, 81–110 (2004)
8. Dubois, D., Prade, H.: Rough Fuzzy Sets and Fuzzy Rough Sets. Internation Journal of General Systems 17(2-3), 191–209 (1990)
9. Halstead, M.H.: Elements of Software Science. Elsevier, New York (1977)
10. Hogg, R.V., Tanis, E.A: Probability and Statistical Inference. Macmillan Publishing Co., Inc, New York (1977)

11. Khoshgoftaar, T.M., Allen, E.B.: Neural networks for software quality prediction. In: Pedrycz, W., Peters, J.F. (eds.) Computational Intelligence in Software Engineering, River Edge, NJ, pp. 33–63. World Scientific, Singapore (1998)
12. McCabe, T.: A complexity measure. IEEE Trans. on Software Engineering SE-2(4), 308–320 (1976)
13. Pawlak, Z.: Rough sets. International J. Comp. Information Science 11(3), 341–356 (1982)
14. Peters, J.F., Ramanna, S.: Towards a software change classification system: A rough set approach. Software Quality Journal 11, 121–147 (2003)
15. Peters, J.F., Pedrycz, W.: Software Engineering: An Engineering Approach. John Wiley and Sons, New York (2000)
16. Ramanna, S., Bhatt, R., Biernot, P.: A Rough-Hybrid Approach to Software Defect Classification. Joint Rough Set Symposium, Canada [to appear] (2007)
17. Quinlan, J.R: Induction of decision trees. Machine Learning 1(1), 81–106 (1986)
18. Tsang, E.C.C., Yeung, D.S., Lee, J.W.T., Huang, D.M., Wang, X.Z.: Refinement of generated fuzzy production rules by using a fuzzy neural network. IEEE Trans. on SMC-B 34(1), 409–418 (2004)
19. Wróblewski, J.: Genetic algorithms in decomposition and classification problem. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery, vol. 1, pp. 471–487. Physica-Verlag, Berlin, Germany (1998)
20. Wang, X.Z., Yeung, D.S., Tsang, E.C.C.: A comparative study on heuristic algorithms for generating fuzzy decision trees. IEEE Trans. on SMC-B 31, 215–226 (2001)
21. Yuan, X., Khoshgoftaar, T.M., Allen, E.B., Ganesan, K.: An application of fuzzy clustering to software quality prediction. In: Proc. 3rd IEEE Symp. on Application-Specific Software Engineering Technology, pp. 85–90. IEEE Computer Society Press, Los Alamitos (2000)

Dimensionality Reduction Using Rough Set Approach for Two Neural Networks-Based Applications

M. Sammany[1] and T. Medhat [2]

[1] National Water Research Center Ministry of Water Resources and Irrigation,
Kornish El-Nile, Imbaba-Giza 12666- Cairo, Egypt
sammanyddr1@gmail.com

[2] Department of Physics and Engineering Mathematics, Faculty of Engineering,Tanta
University, 31521, Tanta, Egypt
tmedhatm@yahoo.com

Abstract. In this paper, Rough Sets approach has been used to reduce the number of inputs for two neural networks-based applications that are, diagnosing plant diseases and intrusion detection. After the reduction process, and as a result of decreasing the complexity of the classifiers, the results obtained using Multi-Layer Perceptron (MLP) revealed a great deal of classification accuracy without affecting the classification decisions.

Keywords: Rough Sets, Neural networks, Multi-layered Perceptron (MLP).

1 Introduction

Recently, Artificial Neural Networks (ANN) has been applied successfully to create accurate and efficient models for classification problems [1]. The initial phase of MLP modeling requires selection of input parameter vectors and the corresponding output vectors, which adequately characterize the component to be modeled. Two rules are thus important regarding the input parameter selection. One is that the input parameter vector should be chosen in a manner so that they will weave a domain to cover the model parameter of interest. The other is to select the parameters without redundancy [2].

Dimensionality reduction methods in general try to find a reduced number of new dimensions to account for the original data. Several techniques are available, which can be seen as variants of factors analysis to find a smaller set of representative dimensions. Principal Component Analysis (PCA) [3] is the best known of these techniques: the new dimensions, linear combinations of the original features, are given by the eigenvectors (ordered by decreasing eigenvalue) of the covariance matrix of input data. The new features, called principal components, are uncorrelated and of maximum variance so that the new representation is now minimal. Successive components are of decreasing importance, and the

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 639–647, 2007.

first principal components (of higher eigenvalue) usually account for most of the variance in the input data. Unfortunately, the size of the covariance matrix is very large for high-dimension data vectors, as input vectors of dimension n give rise to a matrix of size$n \times n$, thus standard PCA methods cannot then deal with data vectors of huge number of features, because space and time costs become prohibitive [4].

Rough Sets technique introduced by Pawlak [5] is another method that can be used to extract patterns from data for classification. It is a mathematical tool to search large, complex databases for meaningful decision rules. Rough set has been applied in many applications such as machine learning, knowledge discovery, and expert systems [6-9]. It deals with the classificatory analysis of data tables. The data can be acquired from measurements or from human experts. The main goal of the rough set analysis is to synthesize approximation of concepts from the acquired data and makes reduction of data to a minimal representation. Many rough sets models have been developed in the rough set community in the last decades [10-14], including VPRS [14] and GRS [15]. Some of them have been applied in the industry data mining projects such as stock market prediction, patient symptom diagnosis, telecommunication churner prediction, and financial bank customer attrition analysis to solve challenging business problems [6,16]. Recently, rough set approach has been used as a tool for reducing the number of input data presented to a neural network [17,18]. In this study, we used the rough set approach to eliminate the superfluous attributes for two multi-class neural networks applications with different number of attributes and classes. These applications are diagnosing plant diseases and intrusion detection. The approach we used for theses applications depends on the degree of dependency between the condition attributes and decision attributes. The degree of dependency measure always lies in the range [0,1], with 0 indicating non-dependency and 1 indicating total dependency. During the analysis process, it has been shown that using a small set of attributes resulted from the rough sets approach did not affect the final classification decisions, and thus reduce the computation time for the proposed MLP models. Finally, the results obtained using neural networks revealed a great deal of classification accuracy in comparison with applying the whole information system to the model.

This paper is organized as follows: Section 2 introduces the basic ideas and concepts of Rough Sets approach. In section 3, we present the theory of rough sets approach for data reduction applied on two cases studies. Finally, the paper ends with a conclusion and the possibilities for future work.

2 Rough Sets Concepts

2.1 Information System

A data set is represented as a table, where each row represents a case, an event, or simply an object. Every column represents an attribute (a variable, an observation, a property, etc.) that can be measured for each object; the attribute may be also be supplied by a human expert or user. This table is called an information system IS [15] which can be defined as $IS = (U, A, \rho, V)$, where $U$is a non-empty finite set of objects called a *universe* and $A$is a non-empty finite set of *attributes*. Any subset $X \subseteq V$will be called a *concept* or a category in U. Each attribute $a \in A$can be viewed as a function ρ that maps elements of Uinto a set V_a, where the set V_a is called the value of a set of the attribute$a$.

$$\rho : U \times A \to V_a \tag{1}$$

2.2 Mathematical Model in Information System

To extract knowledge from information system, it is necessary to find mathematical models using the given data.

- **Indiscernibility Relation**

With any $P \subseteq A$ there is an associated equivalence relation$IND(P)$:

$$IND(P) = \{(x, y) \in U^2 : \forall a \in P, a(x) = a(y)\} \tag{2}$$

The partition of U, generated by$IND(P)$ is denoted $U/IND(P)$ (or U/P) and can be calculated as follows:

$$U/IND(P) = \otimes\{a \in P : U/IND(\{a\})\}, \tag{3}$$

Where

$$A \otimes B = \{X \cap Y : \forall X \in A, \forall Y \in B, X \cap Y \neq \phi\} \tag{4}$$

If$(x, y) \in IND(P)$, then x and $y$are indiscernible by attributes from$P$. The equivalence classes of the $P - indiscernibility$ relation are denoted$[x]_P$.

- **Lower and Upper Approximations**

Let$X \subseteq U$. X can be approximated using only the information contained within P by constructing the$P - lower$ and $P - upper$ approximations of X:

$$\underline{P}X = \{x : [x]_P \subseteq X\} \tag{5}$$

$$\overline{P}X = \{x : [x]_P \cap X \neq \phi\} \tag{6}$$

- **Positive, Negative, and Boundary Regions**

Let C and D be equivalence relations over$U$, then the positive, negative, and boundary regions can be defined as:

$$POS(C, D) = \cup_{X \in U} \underline{P}X \tag{7}$$

$$NEG(C, D) = U - \cup_{X \in U} \overline{P}X \tag{8}$$

$$BND(C, D) = U - POS(C, D) \cup NEG(C, D) \tag{9}$$

The positive region contains all objects of Uthat can be classified to classes of U/Dusing the information in attributes$C$. The boundary region $BND(C, D)$ is the set of objects that can possibly, but not certainly, be classified in this way. The negative region$NEG(C, D)$ is the set of objects that cannot be classified to classes of U/D.

– **Degree of Dependency and Reduction**

An important issue in data analysis is discovering dependencies between attributes. Intuitively, a set of attributes Ddepends totally on a set of attributes C, denoted $C \Rightarrow D$, if all attribute values from D are uniquely determined by values of attributes from C. If there exists a functional dependency between values of$D$and C, then Ddepends totally on C. In rough set theory, dependency is defined as follows: For $C, D \subset A$, it is said that D depends on$C$ in a degree $K(0 \leq K \leq 1)$, denoted $C \Rightarrow_K D$, if

$$K(C, D) = \|POS(C, D)\| \,/\, \|U\| \tag{10}$$

where$\|.\|$ is the cardinality of a set. If $K = 1$, D depends totally on C, if $0 < K < 1$, D depends partially (in a degree K on C, and if $K = 0$ then D does not depend on C. By calculating the change in dependency when an attribute is removed from the set of considered conditional attributes, a measure of the significance of the attribute can be obtained. The higher the change in dependency, the more significant the attribute is. If the significance is 0, then the attribute is dispensable.

– **Reduction of Condition Attributes Relative to Decision Attributes**

Attributes can be divided into condition attributes C and decision attributes$D$. An attribute $a \in C$ is called superfluous with respect to D if $K(C, D) = K(C - \{a\}, D)$, otherwise a is indispensable in C [15]. Eliminating a superfluous $C-$ *attribute* will not decrease or increase the degree of dependency. This means that this attribute is not necessary for the decision.

A subset $M$of the condition attributes is called a reduct of $C$with respect to D if:

$$\text{(i)} \quad K(C, D) = K(M, D),$$
$$\text{(ii)}K(M, D) \neq K(M - \{a\}, D) \;\forall\; a \in M \tag{11}$$

Thus, we can get the minimal reduction of the number of attributes.

3 Two Cases Studies

The applications, which have been chosen for this study, are different in nature wherefrom the dimensionality, degree of complexity, and the number of classes in each application. For example, the first application is a five-class problem with a few number of inputs (in comparison with the second application), but has a high degree of overlapping between classes. On the other hand, the second application is a three-class problem with low degree of complexity, but has a high degree of dimensionality. Both applications have been tackled in our previous studies [19,20] using the whole feature vector as inputs to the proposed MLP models. Here we treat the problems again by using rough sets approach for data reduction. Rosetta (version 1.4.1) is the software package used in our experiments. It is a toolkit for analyzing tabular data within the framework of rough set theory [21].

3.1 Case 1 (Diagnosing Plant Diseases)

- **Description**

In agriculture mass production, it is needed to discover the beginning of plant diseases batches early to be ready for appropriate timing control to reduce the damage and production costs, and increase the income. Leaf batches are considered the important units indicating the existence of diseases in the plant. In order to identify those leaf spots into their cause, we first need to extract their features such as color, shape, and size. Second, we need a classifier capable to learn from those features and then differentiate between them. In this regard, a neural network-based classifier has been used for diagnosing plant diseases. Neural network has been used for classifying the plant symptoms according the leaf spots categories. These categories are, yellow spotted (YS), white spotted (WS), red spotted (RS), and discolored (D) categories. In order to recognize the leaf spot category, a number of features are extracted from a segmented leaf image to be later used for classification. These features correspond to the color characteristics of the spots such as the mean of the gray level of the red, green, and blue channel of the spots. Other features correspond to morphological characteristics of the spots (see Table 1).

Table 1. The entire feature vector for the first case study (Diagnosing Plant Diseases)

#	**Description**	#	**Description**
1	AVR_R,	6	Eccentricity Measure
2	AVR_G	7	Compactness Measure
3	AVR_B	8	Extent Measure
4	The length of the principal axes	9	Euler's Number Measure
5	The diameter of a spot	10	Orientation Measure

- **Data Reduction Using Rough Set Approach**

Table 2 shows the information system of the first case study, which indicates that there are 1640 objects (records) and 11 attributes divided into 10 condition attributes "inputs" and 1 decision attribute (5 classes) as shown:

$$U = \{1, 2, 3, ..., 1640\}, \quad A = \{C, D\}, \quad C = \{i_1, i_2, i_3, ..., i_{10}\}, \quad D = \{\text{output}\}$$

By using rough set approach described in section 2, we get $RED(C)$ as shown in Table 3 (we have 9 reducts). Since the choice of reducts dose not effect on the final decision; therefore, we can use any one of them.

- **Experimental Results and Classifier Evaluation**

This problem has been treated again using two hidden layers neural network. After the reduction process, it has been noticed that; for each of the nine reducts

Table 2. Information system for first case study (Diagnosing Plant Diseases)

U/A	C				D
	i_1	i_2	…	i_{10}	output
1	210.35	156.66	…	133.74	A
2	212.83	158.21	…	23.96	A
…	…	…	…	…	…
500	174.41	85.88	…	-7.7	B
…	…	…	…	…	…
1000	189.44	99.11	…	23.57	C
…	…	…	…	…	…
1500	180.41	118.87	…	64.61	D
…	…	…	…	…	…
1640	167.69	191.61	…..	9.53	E

Table 3. Reduction of Table 2 using Rough Set Approach

#	RED(C)	number of attributes
1	$\{i_2, i_7\}$	2
2	$\{i_2, i_8\}$	2
3	$\{i_1, i_2\}$	2
4	$\{i_2, i_3\}$	2
5	$\{i_1, i_3\}$	2
6	$\{i_2, i_4, i_6\}$	3
7	$\{i_2, i_4, i_9\}$	3
8	$\{i_2, i_4, i_5\}$	3
9	$\{i_2, i_4, i_{10}\}$	3

Table 4. The correct classification rates obtained using MLP for each of the 9 reducts generated using Rough Sets approach, (WFV stands for Whole Feature Vector) (Case 1)

Feature #	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Classification Accuracy**
WFV	X	X	X	X	X	X	X	X	X	X	90%
Reduct 1		X					X				93.78%
Reduct 2		X						X			93.89%
Reduct 3	X	X									91.89%
Reduct 4		X	X								92.65%
Reduct 5	X		X								93.56%
Reduct 6		X		X		X					95.89%
Reduct 7		**X**		**X**					**X**		**96.93%**
Reduct 8		X		X	X						93.96%
Reduct 9		X		X						X	94.51%

obtained using rough set approach there is a considerable improvement in classification accuracy, comparing to our previous study. Table 4 illustrates the correct classification rates for each reduct.

Obviously, it can be seen from the previous table that, all of the scenarios resulted from the rough sets approach gave a higher classification rates than using the whole feature vector (90%) (See first row in the Table 1). This means that reducing the input vector improved the generalization capability of MLP without affecting the classification decision.

3.2 Case 2 (Intrusion Detection)

– **Description**

The ubiquity of the Internet poses serious concerns on the security of computer infrastructures and the integrity of sensitive data. Network Intrusion Detection Systems aim at protecting networks and computers from malicious network-based attacks. A MLP neural network has been used to recognize tow types of attacks in addition to the normal case (no attack), thus we have a three- classes problem.

– **Data Reduction Using Rough Set Approach**

The information system of the second case study is described as follows:

There are 13058 objects and 36 attributes divided into 35 condition attributes (inputs) and 1 decision attribute (output) as shown:

$$U = \{1, 2, ..., 13058\}, A = \{C, D\}, C = \{i_1, i_2, ..., i_{35}\}, D = \{out\}$$

By using rough set approach described in section 2, we get $RED(C)$ as shown in Table 5. From Table 5 we can see that we have 100 reducts. Since the choice of reducts dose not effect on the final decision; therefore, we can use any one of them.

Table 5. Reduction of using Rough Set Approach for Intrusion Detection

#	RED(C)	number of attributes
1	$\{i_5, i_7, i_{21}\}$	3
2	$\{i_5, i_{19}, i_{30}\}$	3
3	$\{i_5, i_{19}, i_{26}\}$	3
4	$\{i_5, i_{24}, i_{26}\}$	3
5	$\{i_5, i_{24}, i_{30}\}$	3
6	$\{i_5, i_{20}, i_{24}, i_{31}\}$	4
7	$\{i_5, i_{21}, i_{27}, i_{35}\}$	4
8	$\{i_5, i_{21}, i_{24}, i_{31}\}$	4
...	...	...
99	$\{i_5, i_{22}, i_{24}, i_{27}\}$	4
100	$\{i_5, i_{17}, i_{22}, i_{32}, i_{33}\}$	5

- **Experimental Results and Classifier Evaluation**

This problem has been treated again using two hidden layers of MLP. After the reduction process it has been noticed that, for most of the reducts obtained using rough set approach there was a considerable improvement in the classification accuracy in comparison with our previous study (94%) [18]. Applying the reduct $\{i_5, i_{17}, i_{22}, i_{32}, i_{33}\}$we got 98% of classification accuracy (on the same test set) using two hidden layers MLP.

4 Conclusion and Future Work

In this paper, Rough Sets approach has been used to reduce the number of inputs for two neural networks-based applications; that is, diagnosing plant diseases and intrusion detection. After the reduction process, and as a result of decreasing the complexity of the classification models the results obtained using Multi-Layer Perceptron (MLP) revealed a great improvement in classification accuracy than using the whole feature vector used in our previous studies. By means of rough set approach, it has been shown that reducing the number of features introduced to the network increases the model accuracy without affecting the classification decisions. More challenging classification problems can be tackled using rough set approach if we transformed the problem into higher dimension feature space that is, the *Polish Space*. Such spaces have many properties to do an optimum classification, which we will focus on in our future studies.

Acknowledgements. The authors would like to THANK.

- Mehdi Moradi, from Queen's University Kingston, Ontario, Canada, for presenting his work in the field of intrusion detection and for giving the data sets used in our experimentations, and for his guidance while doing this research.
- Dr. Mohammed Hussein El-Helly, from the Central Lab of Agriculture and Expert Systems (CLAES), Cairo, Egypt, for providing the data set used in our experiments, and his support concerning the part of diagnosing plant diseases.

References

1. Theodorios, Koutroumbas, K.: Pattern Recognition. Academic Press, Cambridge (1999)
2. Watson, P.M., Mah, M.Y, Liou, L.L A.: Input Variable Space Reductin Using Dimensional Analysis For Artificial Neural Network Modeling. IEEE. MTT-S Digest. Libr. 1, 269–272 (1999)
3. Jolliffe, I.T.: Principal Component Analysis. Springer, Heidelberg (1986)
4. Delichere, M., Memmi, D.: Neural Dimensionality Reduction for Document Processing. ESANN'2002 proceedings- European Symposium on Artificial Neural Networks, Bruges (Belgium) (April 24-26, 2002)

5. Pawlak, Z.: Rough Sets. International Journal of Information and computer Science 11, 341–356 (1982)
6. Lin, T.Y.: From Rough Sets and Neighborhood Systems to Information Granulation and Computing in Words. In: Proceedings of European Congress on Intelligent Techniques and Soft Computing, pp. 1602–1607 (1997)
7. Lin, T.Y.: Granular computing on binary relations I: data mining and neighborhood systems. II: rough set representations and belief functions. In: Lin, T.Y., Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery, pp. 107–140. Physica-Verlag, Heidelberg (1998)
8. Lin, T.Y., Yao, Y.Y., Zadeh, L.A. (eds.): Rough Sets. Granular Computing and Data Mining. Physica-Verlag, Heidelberg (2002)
9. Pawlak, Z.: Rough Sets - Theoretical Aspects of Reasoning about data. Kluwer Academic Publishers, Dordrecht (1991)
10. Hu, X., Cercone, N., Han, J., Ziarko, W.: GRS: A Generalized Rough Sets Model. In: Lin, T.Y., Yao, Y.Y., Zadeh, L. (eds.) Data Mining, Data Mining, Rough Sets and Granular Computing, pp. 447–460. Physica-Verlag, Heidelberg (2002)
11. Polkowski, L., Skowron, A.: Rough mereology. In: Proc. ISMIS, Charlotte, NC, pp. 85–94 (1994)
12. Polkowski, L., Skowron, A.: Rough mereology: A new paradigm for approximate reasoning. J. of Approximate Reasoning 15(4), 333–365 (1996)
13. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27(2-3), 245–253 (1996)
14. Ziarko, W.: Variable Precision Rough Set Model. Journal of Computer and System Sciences 46(1), 39–59 (1993)
15. Medhat, T.: Topological applications on information analysis by rough sets. Master thesis, Egypt, Tanta University, Faculty of Engineering (2004)
16. Lin, T.Y., Polkowski, L., Skowron, A. (eds.): In Rough Sets in Knowledge Discovery, pp. 107–140. Physica-Verlag, Heidelberg (1998)
17. Sapiecha, P., Selvaraj, H., Staczak, J., Sp, K., uba, T.: A Hybrid Approach to a Classifica tion Problem, Intelligent Information Processing and Web Mining. In: Kopotek, M.A., Wierzcho, S.T., Trojanowski, K. (eds.) Proceedings of the International IIS:IIPWM'04, Zakopane, Poland, May 17-20, 2004. LNCS (LNAI), pp. 99–106. Springer, Heidelberg (2004)
18. Buciak, P., uba, T., Niewiadomski, H., Pleban, M., Sapiecha, P., Selvaraj, H.: Decomposition and Argument Reduction of Neural Networks. In: IEEE Sixth International Conference on Neural Networks and Soft Computing (ICNNSC'02), Zakopane, Poland (June 11-15, 2002)
19. Sammany, M., Zagloul, K.: Support Vector Machine Versus an Optimized Neural Networks fro Diagnosing Plant Diseases. In: Proceeding of Second International Computer Engineering Conference, 26–28, 2006, IEEE (Egypt Section), RH 25-31 (2006)
20. Sammany, M., Sharawi, M., El-Beltagy, M., Saroit, I.: Artificial Neural Networks Architecture For Intrusion Detection Systems and Classification of Attacks. Accepted for publication in the fifth international conference- INFO 2007, from (March 24-26, 2007)
21. RSES version 1.4.1, copyright © 1996-2001 Knowledge Systems Group, Dept. of Com puter and Information Science, NTNU, Norway and Logic Group, Inst. of Mathematics, Warsaw University, Poland
http://www.idi.ntnu.no/~aleks/rosetta/

Decision Tables in Petri Net Models

Marcin Szpyrka and Tomasz Szmuc

Institute of Automatics,
AGH University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
mszpyrka@agh.edu.pl, tsz@agh.edu.pl

Abstract. Many monitoring and control computer systems contain a rule-based system as a part of them. Such a rule based system is used to determine which actions should be taken depending on the data collected from sensors. For both embedded and rule-based systems many different approaches have been proposed, but it is hardly possible to find a formalism that can cope with both of them. The paper deals with a problem of including decision tables into colour Petri net models. A few kinds of decision tables are considered and methods of transformation them into coloured Petri nets form called D-nets are presented. Both non-hierarchical and hierarchical D-nets are considered in the paper.

1 Introduction

The use of formal methods for embedded system development is motivated by the expectation that performing appropriate mathematical analyses can contribute to the software quality. Formal methods are usually used in the development of safety-critical systems, i.e. systems that may result in injury, loss of life or serious environmental damage upon their failure [9]. The high cost of safety-critical systems failure means that trusted methods and techniques must be used for development. For such systems, the costs of verification and validation are usually very high (more than 50% of the total system development cost). Using of formal methods can reduce the amount of testing and ensure more dependable products [3].

Multiple embedded systems are control systems that monitor quantities of interest in an environment. In response to changes in the monitored quantities they perform control operations or other externally visible actions [1]. The process of making decision which actions should be performed may be based on a rule-based system that is incorporated into such an embedded system. Thus, formal methods are useful for modelling of such systems only if rule-based systems can be also expressed in the selected formalism.

Rule-based systems can be represented in various forms, e.g. decision tables, decision trees, extended tabular trees (XTT, [7]), Petri nets [4] etc. An interesting comparison of different forms of rule-based systems description can be found in [6]. Decision tables seem to be the most popular form of rule-based systems presentation. They vary widely in the way the condition and decision entries are represented. The entries can take the form of simple true/false values, atomic values of different types, non-atomic values or even fuzzy logic formulas.

A decision table represents a set of decision rules that can be given explicitly by an expert or generated from analyzed data automatically, e.g. using rough sets approach

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 648–657, 2007.

[2]. The paper deals with a problem of including of an already constructed decision table into a hierarchical colour Petri net model (CP-net [5] or RTCP-net [10]). Both decision tables with atomic values of attributes [6] and with generalized rules [11] are considered in the paper.

The paper is organized as follows. Decision tables with atomic values of attributes are described in section 2. Generalized decision tables are presented in section 3. Hierarchical D-nets are considered in section 4. A more practical example of a decision table is presented in section 5. Computer software for decision tables called *Adder Designer* is described is section 6. The paper is shortly concluded in the final section.

2 Decision Tables with Atomic Values of Attributes

Let's recall the definition of a decision table presented in [8]. At first sight a decision table can be treated as an extension of a knowledge representation system. Such a system is a pair $\mathcal{K} = (U, A)$, where U is a nonempty, finite set called the *universe*, and A is a nonempty, finite set of *attributes*. Every attribute $a \in A$ is a function $a\colon U \to V_a$, where V_a denotes the domain of a.

To transform a knowledge representation system into a decision table, we have to distinguish two subsets of A called *conditional* (C) and *decision* (D) attributes respectively. In case of a decision table the elements of the set U denote not any real objects, but are identifiers of decision rules. Hence the symbol R will be used instead of U. Therefore, a decision table is a tuple $\mathcal{T} = (R, A, C, D)$, where $C, D \subset A$.

Such a decision table is often called a *table with atomic values of attributes* (or *simple decision table*, [6]). To construct such a decision table, we draw a column for each conditional and decision attribute. Then, for every decision rule a row should be drawn. We fill cells so as to reflect which decisions are generated for each combination of conditions. An example of a simple decision table is shown in Tab. 1.

Table 1. Example of a simple decision table ($\mathcal{T}_1$)

	a	b	c	d	e
R1	1	2	1	1	1
R2	1	2	2	2	2
R3	2	1	1	1	1
R4	2	2	2	2	2
R5	3	1	2	2	1
R6	3	2	2	2	2
R7	4	1	1	2	1
R8	4	2	2	2	1

The table $\mathcal{T}_1$ contains three conditional and two decision attributes ($C = \{a, b, c\}$, $D = \{d, e\}$). In this case, for each attribute its domain is a subset of natural numbers, but in general other types are also possible (e.g. real, boolean, enumerated types, etc.)

To include the decision table $\mathcal{T}_1$ into a Petri net model (CP-net or RTCP-net), it must be first transformed into a D-net [11]. A D-net is a non-hierarchical coloured Petri net that represents a set of decision rules. It contains two places: a *conditional place* (input place) for values of conditional attributes and a *decision place* (output place) for values of decision attributes. Types for these places are defined as the Cartesian product of domains of conditional and decision attributes respectively.

Each decision rule is represented by one transition and its input and output arcs. The conditional part of a rule is represented by the expression attached to the input arc of the corresponding transition. Similarly, the decision part is represented by the expression attached to the output arc.

D-nets are used as the bottom level pages in hierarchical models [10]. The conditional and decision places are input and output ports respectively. The superpage (see [5]) for such a D-net is used to gather all necessary information for the D-net and to distribute the results of its activity. In other words, the superpage prepares a token that represents a sequence of values of conditional attributes and that is next placed on the conditional place. If at least one transition in the D-net is enabled, the token is removed from the conditional place and a new token that represents a decision is added to the decision place. Then the superpage removes the token and brings the decision into effect. The D-net form of the decision table $\mathcal{T}_1$ is shown in Fig. 1.

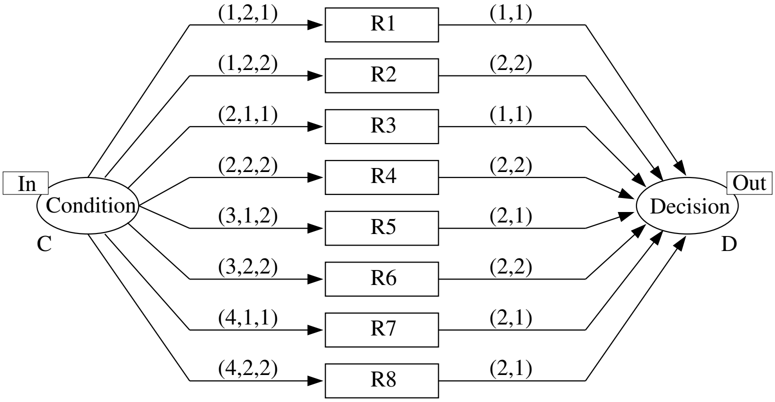


Fig. 1. D-net for the decision table $\mathcal{T}_1$

The decision table presented in Tab. 1 contains values for all conditional attributes. Methods based on the rough set theory can be used to reduce such a decision table. The reduction algorithm consists in the elimination of conditions from a decision table, which are unnecessary to make decisions specified in the table (see [8]).

Let's consider the decision tables presented in Tab. 2. Some redundant values of conditional attributes are omitted in the table. In such a case to transform a decision table into a D-net variables have to be used. In this case we need two variables for attributes a and b. An variable attached to an attribute $x \in C$ can take any value that belongs to the domain of the attribute.

Thus before the transformation algorithm can be applied, the table $\mathcal{T}_2$ is represented in the form shown in Tab. 3. For simplicity, the name of an variable is the same as the name of the corresponding attribute. The D-net form of the decision table $\mathcal{T}_2$ is shown in Fig. 2.

Table 2. Example of a simple decision table ($\mathcal{T}_2$)

	a	b	c	d
R1	1	1	1	1
R2	1	2	1	2
R3	4	1	1	2
R4	4	2	1	1
R5	2	–	1	1
R6	3	–	1	1
R7	–	–	2	2
R8	–	–	3	2

Table 3. Decision table $\mathcal{T}_2$ with variables

	a	b	c	d
R1	1	1	1	1
R2	1	2	1	2
R3	4	1	1	2
R4	4	2	1	1
R5	2	b	1	1
R6	3	b	1	1
R7	a	b	2	2
R8	a	b	3	2

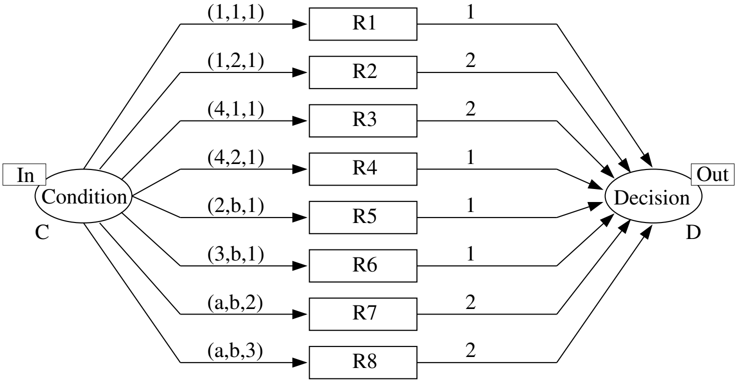


Fig. 2. D-net for the decision table $\mathcal{T}_2$

3 Generalized Decision Tables

Encoding decision tables with the use of atomic values of attributes only is not sufficient for many real applications. If the domains of attributes contain more than several values it may be really hard to cope with the number of decision rules. To handle the problem one can use formulas instead of atomic values of attributes. In such a case, a cell in a decision table will contain a formula that evaluates to a boolean value for conditional attributes, and to a single value (that belongs to the corresponding domain) for decision attributes.

The result of this approach is a decision table with generalised decision rules (or rules' patterns). Each generalised decision rule covers a set of decision rules with atomic values of attributes. Such decision tables will be called *generalized decision tables*. An example of a generalized decision table is presented in Tab. 4. Domains for these attributes are defined as follows:

$V_a = V_d = \{1, 2, 3, 4, 5\}$,
$V_b = V_e = \{\mathit{off}, \mathit{on}\}$, (Boolean values)
$V_c = \{x, y, z\}$.

Table 4. Example of a generalized decision table

	a	b	c	d	e
R1	$a < 4$	$b = on$	$c = x$	$a + 2$	on
R2	a	$b = on$	$c \neq y$	3	off
R3	$a = 5$	b	c	2	$\neg b$
R4	$a > 2$	b	$c \neq x$	$a - 2$	on
R5	$a = 2$	b	$c = x$	4	on

A generalized decision table can be also transformed into the D-net form (see Fig. 3). Formulas that describe values of conditional attributes are usually attached to the guard of the corresponding transition. The algorithm of transformation of a generalized decision table into the D-net form can be found in [11].

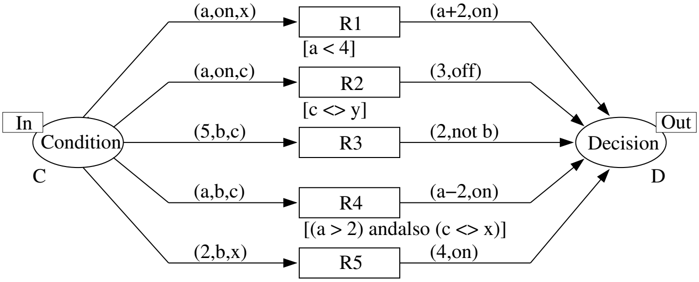


Fig. 3. D-net form of the decision table presented in Tab. 4

4 Hierarchical D-Nets

The main problem in the rule-based system design process is that it is difficult to cope with systems having more than several rules. To simplify the design process a decision table $\mathcal{T} = (R, A, C, D)$ can divided into a set of tables $\mathcal{T}_1 = (R_1, A_1, C_1, D), \cdots, \mathcal{T}_n = (R_n, A_n, C_n, D)$ such that $R_1, \cdots, R_n \subseteq R$, $R_i \cap R_j = \emptyset$ for $i \neq j$, $R = \bigcup_{i=1}^{n} R_i$, $A_1, \cdots, A_n \subseteq A$, $C_1, \cdots, C_n \subseteq C$ and $C_i = C \cap A_i$ for $i = 1, \ldots, n$.

Such decomposition for the rule-based system presented in Tab. 3 is shown in Tab. 5.

Table 5. Table $\mathcal{T}_2$ split into parts

	a	b	c	d
R1	1	1	1	1
R2	1	2	1	2
R3	4	1	1	2
R4	4	2	1	1

	a	c	d
R5	2	1	1
R6	3	1	1

	c	d
R7	2	2
R8	3	2

For each of the three decision tables a D-net (called *sub-D-net*) can be constructed as it was shown in section 2. Next such sub-D-nets are combine into one hierarchical structure. In order to design a hierarchical D-net, a superpage with a substitution transition for each sub-D-net is constructed. The superpage for the consider rule-based system is presented in Fig. 4.

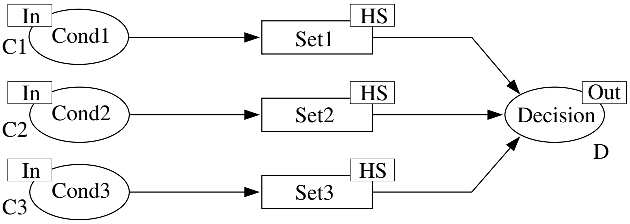


Fig. 4. Superpage for the hierarchical D-net

The superpage together with three sub-D-nets constitute a *hierarchical D-net*. In this case the superpage contains three different conditional places because we have three different sets of conditional attributes. In general the set of rules can be divided into subsets such that some of them share the same set of conditional attributes. In such a case the number of conditional places in the corresponding superpage is less than the number of sub-D-nets. In particular only one conditional place can be used.

On the other hand, the one conditional place is used if we define its colour as a union [5]. For the considered example the conditional place colour should be defined as $C = C1 \cup C2 \cup C3$. However, using of a few conditional places seems to be more practical. The general scheme of a hierarchical D-net is shown in Fig. 5.

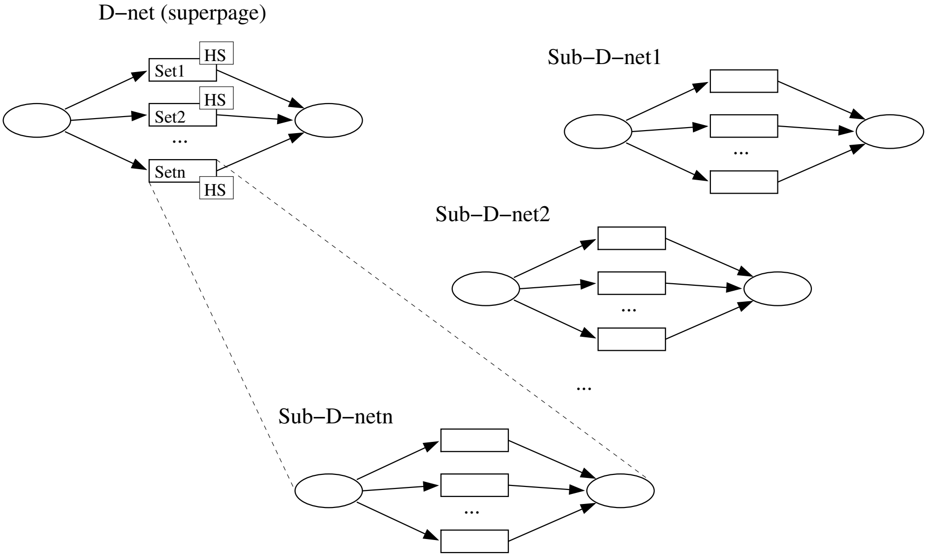


Fig. 5. General structure of a hierarchical D-net

5 Example

Let's consider an example of computer network design, presented in Fig. 6 [7]. It is a typical configuration for many security-aware small office, or company networks. The network is composed of three subnetworks: LAN (local area network), DMZ (the so-called *demilitarized zone*), and INET (Internet connection). The subnetworks are separated by a *firewall* having three network interfaces.

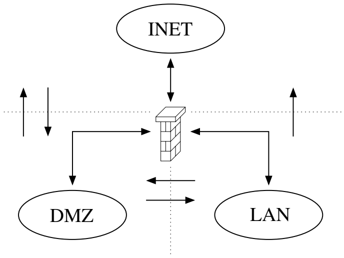


Fig. 6. Network firewall configuration

The firewall controlls the input and output and decides whether the request should be accepted or rejected. Decision table for such a firewall system contains three conditional (service, source address, destination address) and one decision attribute (routing). The attribute $Service$ stands for a type of the net service, attributes $Srcaddr$ and $Destaddr$

are connected with source and destination IP addresses respectively, and the attribute *Routing* stands for the final routing decision. Domains for these attributes are defined as follows:

$$D_{Service} = \{ssh, smtp, http, imap\},$$
$$D_{Srcaddr} = D_{Destaddr} = \{inet, dmz, lan\};$$
$$D_{Routing} = \{accept, reject\}.$$

A complete decision table for the firewall system (presented in Tab. 6) contains eleven positive and four negatives rules. The negative rules (without values of decision attributes) are used to state in an explicit way that the particular combinations of input values (values of conditional attributes) are impossible or not allowed. The negative rules are used to check whether the table is complete and are usually omitted when the corresponding D-net is generated.

D-net generated for the considered decision table is shown in Fig. 7.

Table 6. Decision table for the firewall system

Service	*Srcaddr*	*Destaddr*	*Routing*
$Service = http$	$Srcaddr = inet$	$Destaddr = dmz$	$accept$
$Service = http$	$Srcaddr = inet$	$Destaddr = lan$	$reject$
$Service = http$	$Srcaddr = lan$	$Destaddr$	$accept$
$Service = smtp$	$Srcaddr$	$Destaddr = lan$	$reject$
$Service = smtp$	$Srcaddr$	$Destaddr = dmz$	$accept$
$Service = smtp$	$Srcaddr = lan$	$Destaddr = inet$	$reject$
$Service = imap$	$Srcaddr = lan$	$Destaddr = dmz$	$accept$
$Service = imap$	$Srcaddr \neq lan$	$Destaddr$	$reject$
$Service = ssh$	$Srcaddr = inet$	$Destaddr$	$reject$
$Service = ssh$	$Srcaddr = lan$	$Destaddr$	$accept$
$Service = ssh$	$Srcaddr = dmz$	$Destaddr$	$accept$
$Service = http$	$Srcaddr = dmz$	$Destaddr$	
$Service = http$	$Srcaddr = inet$	$Destaddr = inet$	
$Service = imap$	$Srcaddr = lan$	$Destaddr \neq dmz$	
$Service = smtp$	$Srcaddr \neq lan$	$Destaddr = inet$	

6 Adder Designer

Manual analysis of a decision table can be time-consuming even for very small sets of decision rules. *Adder Designer* supports design and analysis of both simple and generalized decision tables. The tool is equipped with a decision table editor and verification procedures.

Adder Designer is a free software covered by the GNU Library General Public License. It is being implemented in the GNU/Linux environment by the use of the Qt Open Source Edition. The Qt library is freely available for the development of Open Source software for Linux, Unix, Mac OS X and Windows under the GPL license. Code written for either environment compiles and runs with the other ones. *Adder Tools home page*, hosting information about the current status of the project, is located at

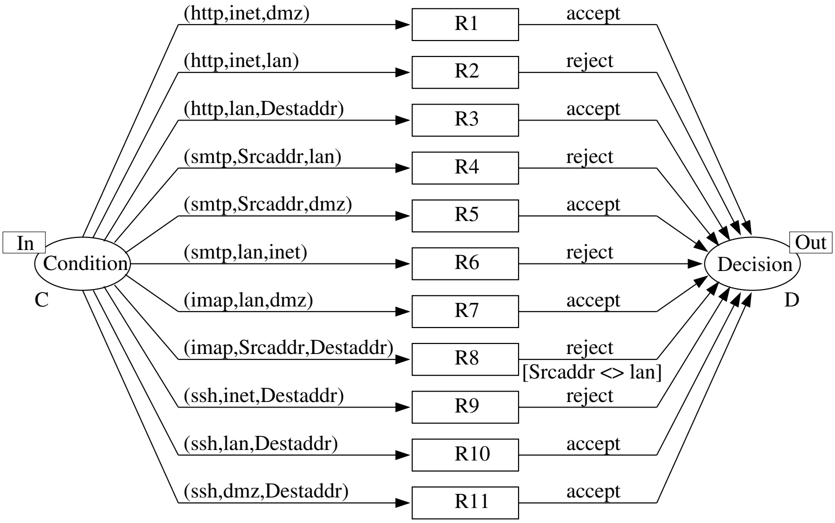
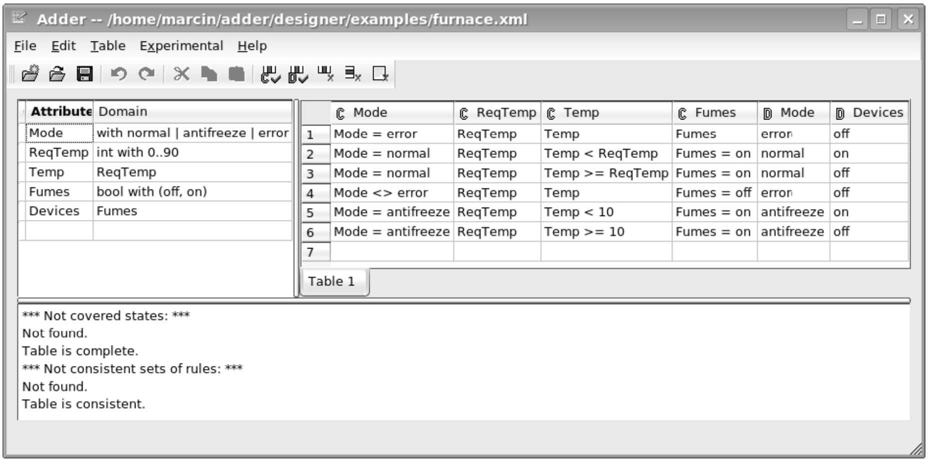


Fig. 7. D-net for the network firewall system



Attribute	Domain
Mode	with normal | antifreeze | error
ReqTemp	int with 0..90
Temp	ReqTemp
Fumes	bool with (off, on)
Devices	Fumes

	C Mode	C ReqTemp	C Temp	C Fumes	D Mode	D Devices
1	Mode = error	ReqTemp	Temp	Fumes	error	off
2	Mode = normal	ReqTemp	Temp < ReqTemp	Fumes = on	normal	on
3	Mode = normal	ReqTemp	Temp >= ReqTemp	Fumes = on	normal	off
4	Mode <> error	ReqTemp	Temp	Fumes = off	error	off
5	Mode = antifreeze	ReqTemp	Temp < 10	Fumes = on	antifreeze	on
6	Mode = antifreeze	ReqTemp	Temp >= 10	Fumes = on	antifreeze	off
7						

```
*** Not covered states: ***
Not found.
Table is complete.
*** Not consistent sets of rules: ***
Not found.
Table is consistent.
```

Fig. 8. Example of Adder Designer session

http://adder.ia.agh.edu.pl. An example of Adder Designer session is shown in Fig. 8. The figure contains a decision table for a home heating system with a boiler fueled by natural gas and results of completeness and consistency (determinism) analysis.

Using of Adder Designer for design of decision tables consists of a few steps. It is first necessary to define attributes selected to describe important features of the system under consideration. There are possible three types of domains: integer, boolean and enumerated data type. Moreover, a new domain may be defined as an alias for already defined one. Secondly, it is necessary to choose conditional and decision attributes. Each attribute can be used twice. Finally, the set of decision rules should be defined.

The verification stage is included into design process. At any time, during the design stage, users can check whether a decision table is complete, consistent (deterministic) or it contains some dependent rules. Moreover, the tool enables users to *pack* a simple decision table to a generalized one and vice versa.

The tool and the presented approach have been successfully used for developing a few practical examples of rule-based systems, e.g. for a railway traffic management system (22 attributes, 123 decision rules).

7 Summary

Methods of transformation of decision tables into a coloured Petri net form called D-net were presented in the paper. The presented approach can be used to transform into D-nets both simple and generalized decision tables. Moreover, it is also possible to construct hierarchical D-nets that can be treated as a structural form of presentation of rule based systems with many decision rules. The presented approach is supported with computer tools for design and verification of decision tables.

References

1. Adamski, M., Karatkevich, A., Węgrzyn, M. (eds.): Design of Embedded Control Systems. Springer Science+Business Media. Springer, Heidelberg (2005)
2. Bazan, J., Nguyen, S., Nguyen, T., Skowron, A., Stepaniuk, J.: Decision rules synthesis for object classification. In: Orłowska, E. (ed.) Incomplete Information: Rough Set Analysis, pp. 23–57. Physica-Verlag, Heidelberg (1998)
3. Cheng, A.M.K.: Real-time Systems. Scheduling, Analysis, and Verification. Wiley Interscience, New Jersey (2002)
4. Fryc, B., Pancerz, K., Suraj, Z.: Approximate Petri nets for rule-based decision making. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) Rough Sets and Current Trends in Computing. LNCS (LNAI), vol. 3066, pp. 733–742. Springer, Heidelberg (2004)
5. Jensen, K.: Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use, vol. 1–3. Springer, Heidelberg (1992-1997)
6. Ligęza, A.: Logical foundations of rule-based systems. Studies in Computational Intelligence, vol. 11. Springer, Heidelberg (2006)
7. Nalepa, G.J., Ligęza, A.: Designing reliable web security systems using rule-based systems approach. In: Menasalvas, E., Segovia, J., Szczepaniak, P.S. (eds.) Advances in Web Intelligence. LNCS (LNAI), vol. 2663, pp. 124–133. Springer, Heidelberg (2003)
8. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
9. Sommerville, I.: Software Engineering. Pearson Education Limited, London (2004)
10. Szpyrka, M., Szmuc, T.: Integrated approach to modelling and analysis using RTCP-nets. IFIP International Federation for Information Processing 227, 115–120 (2006)
11. Szpyrka, M., Szmuc, T.: D-nets – Petri net form of rule-based systems. Foundations of Computing and Decision Sciences 31, 157–167 (2006)

Two Types of Generalized Variable Precision Formal Concepts

Hong-Zhi Yang[1,2] and Ming-Wen Shao[3,4]

[1] Faculty of Science, Xi'an Jiaotong University, Xi'an, Shaan'xi 710049, P.R. China
[2] Pingyuan University, Xinxiang, Henan Province 453003, P.R. China
yzxz@163.com
[3] School of Information Technology, Jiangxi University of Finance & Economics, Nanchang, Jiangxi 330013, P.R. China
[4] Jiangxi Key Laboratory of Data and Knowledge Engineering, Jiangxi University of Finance & Economics, Nanchang, Jiangxi 330013, P.R. China
shaomingwen1837@163.com

Abstract. In this paper, we introduce two pairs operators in fuzzy formal contexts. Based on the proposed operators, we present two types of generalized variable precision formal concepts, i.e. property oriented crisp-fuzzy concepts and object oriented fuzzy-crisp concepts. We have different level generalized formal concepts with different precision level. Last, we discuss the relationship between different precision level generalized concepts lattices in details.

Keywords: Formal concept analysis, fuzzy formal context, concept lattice; variable precision.

1 Introduction

The theory of formal concept analysis (FCA) proposed by Wille [1,2] is initially formalized as mathematical thinking for conceptual data analysis and knowledge processing, in which the notion of formal concept originally come from formal logic. From the early of 1980s, FCA has been studied intensively, and have attracted the wide attention of numerous researchers. Over the years development both in theoretical and practical issues, FCA have grown to a powerful theory for conceptual knowledge processing. Now, FCA has been successfully used in various fields such as data mining,information retrieval, knowledge acquisition, software engineering, data base management systems [3,4,5,6,7].

In FCA, formal concepts and concept lattices are two central issues where the concept lattice is usually used represent the domains of knowledge representation and knowledge discovery. Wille's definition of a concept be a (objects, attributes) pair, the set of objects is referred to as the extension and the set of attributes as the intension of formal concept. They uniquely determine each other [1,2].

FCA is analyzed based on a formal context, which is a binary relation between a set of objects and a set of attributes with the value 0 and 1. However, in many practical applications, the binary relation is a fuzzy set represented by a membership degrees, instead of a single value in $\{0, 1\}$. For this fuzzy binary

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 658–667, 2007.

relation, several generalizations to formal concept can be found in the existent literatures [8,9,10,11,12,13]. In [9], Elloumi defined a Lukasiewicz based fuzzy Galois connection. Belohlavek [10,11]proposed fuzzy concepts in fuzzy formal context based on residuated lattice. Moreover, Georgescu and Popescu [12,13] discussed a general approach to fuzzy FCA. The constant study to the fuzzy binary relation has bringed on the notion of "fuzzy concept", which now be successfully used for the fuzzy classier and decision making [14,15].

Fan present the notion of variable precision concept lattice [16]. The notions of the object oriented formal concept and the property oriented formal concept are proposed by [17,18] and Duntsch [19] respectively. In the paper, we introduce two pairs operators in fuzzy formal contexts. Based on the proposed operators, we present two types of generalized variable precision formal concepts, i.e. property oriented crisp-fuzzy concepts and object oriented fuzzy-crisp concepts. We have different level generalized formal concepts with different precision level. The relationship between different precision level generalized concepts lattices are also discussed in details.

2 Two Kinds of Multi-level Formal Concepts

In the following, we recall the notion of residuated lattice and some of its basic properties.

Definition 1. [11] *A residuated lattice is a structure* $(L, \vee, \wedge, \otimes, \rightarrow, 0, 1)$ *such that*

(1) $(L, \vee, \wedge, 0, 1)$ *is a lattice with the least element* 0 *and the greatest element* 1*;*

(2) $(L, \otimes, 1)$ *is a commutative monoid;*

(3) *for all* $a, b, c \in L, a \leq b \rightarrow c$ *iff* $a \otimes b \leq c$*.*

Residuated lattice L is called complete if $(L, \vee, \wedge)$ is a complete lattice.

Lemma 1. [11,12] *The following hold in any complete residuated lattice:*

(1) $a \rightarrow 1 = 1$*;* $1 \rightarrow a = a$*;* $a \rightarrow b = 1$ *iff* $a \leq b$*;* $0 \otimes a = a \otimes 0 = 0$*;*

(2) $\rightarrow$ *is antitone in the first and isotone in the second argument;* $a \leq (a \rightarrow b) \rightarrow b$*;*

(3) $a \rightarrow b \leq (b \rightarrow c) \rightarrow (a \rightarrow c)$*;* $a \rightarrow b \leq (c \rightarrow a) \rightarrow (c \rightarrow b)$*;* $a \rightarrow (b \rightarrow c) = b \rightarrow (a \rightarrow c)$*;*

(4) $(\bigvee_{i\in I} a_i) \rightarrow a = \bigwedge_{i\in I}(a_i \rightarrow a)$*;* $a \rightarrow (\bigwedge_{i\in I} a_i) = \bigwedge_{i\in I}(a \rightarrow a_i)$*;*

(5) $\bigwedge_{i\in I}(a_i \rightarrow b_i) \leq (\bigwedge_{i\in I} a_i) \rightarrow (\bigwedge_{i\in I} b_i)$*;* $\bigwedge_{i\in I}(a_i \rightarrow b_i) \leq (\bigvee_{i\in I} a_i) \rightarrow (\bigvee_{i\in I} b_i)$*;*

(6) $\otimes$ *is isotone in both arguments;* $a \otimes b \leq a$*;* $a \otimes b \leq b$*;*

(7) $b \leq a \rightarrow (a \otimes b)$*;* $(a \rightarrow b) \otimes a \leq b$*;* $(a \otimes b) \rightarrow c = a \rightarrow (b \rightarrow c)$*;*

(8) $a \rightarrow b \leq (a \otimes c) \rightarrow (b \otimes c)$*;* $(a \rightarrow b) \otimes (b \rightarrow c) \leq (a \rightarrow c)$*;*

(9) $(\bigvee_{i\in I} a_i) \otimes a = \bigvee_{i\in I}(a_i \otimes a)$*;* $(\bigwedge_{i\in I} a_i) \otimes a \leq \bigwedge_{i\in I}(a_i \otimes a)$*.*

Let L be a residuated lattice. An $L - set$ A on a universe set U is any map A: $U \to L, A(x)$ being interpreted as the truth degree of the fact " x belongs to A". By L^U denote the set of all $L - set$ in U. For any $X_1, X_2 \in L^U$, $X_1 \subseteq X_2$ if and only if $X_1(x) \leq X_2(x)$ ($\forall\ x \in U$). Operations $\vee$ and $\wedge$ on L^X are defined by:

$$(X_1 \vee X_2)(x) = X_1(x) \vee X_2(x),\ (X_1 \wedge X_2)(x) = X_1(x) \wedge X_2(x), \quad \forall X_1, X_2 \in L^U.$$

Example 1. Let $U = \{x_1, x_2, \ldots, x_n\}$ be a set of n elements, An $L - set$ A on U is denoted by $A = \{x_1/A(x_1), x_2/A(x_2), \ldots, x_n/A(x_n)\}$, where $A(x)$ represents the degree to which an element $x \in U$ is an element of A.

A fuzzy formal context is defined as a triple (U, M, R), where U and M are the object and attribute sets, $R \in L^{U \times M}$ is a fuzzy relation between U and M.

Example 2. Table 1 represents a fuzzy formal context (U, M, R) with $U = \{x_1, x_2, x_3, x_4\}$ and $M = \{a, b, c, d\}$, the fuzzy relation R defined as in Table 1.

Table 1.

R	a	b	c	d
x_1	0.4	0.4	0.9	0.6
x_2	0.8	0.2	0.7	0.8
x_3	0.5	0.4	0.7	0.9
x_4	0.8	0.2	0.7	0.7

Let (U, M, R) be a fuzzy formal context and we denote the power set of U by $\mathcal{P}(U)$. For any $\delta \in (0, 1]$, a pair of operators, ${}^\Diamond$, ${}^\Box : \mathcal{P}(U) \longrightarrow L^M$, defined for $X \in \mathcal{P}(U)$ and $a \in M$ by

$$X^\Diamond(a) = \bigvee\nolimits_{x \in X}(\delta \otimes R(x, a)),$$
$$X^\Box(a) = \bigwedge\nolimits_{x \in X}(R(x, a) \to \delta).$$

Analogously, for any $B \in \mathcal{P}(M)$, a pair of approximation operators, ${}^\Diamond$, ${}^\Box$: $\mathcal{P}(M) \longrightarrow L^U$, defined for $B \in \mathcal{P}(M)$ and $x \in U$ by

$$B^\Diamond(x) = \bigvee\nolimits_{b \in B}(\delta \otimes R(x, b)),$$
$$B^\Box(x) = \bigwedge\nolimits_{b \in B}(R(x, b) \to \delta).$$

Let (U, M, R) be a fuzzy formal context and $X \subseteq L^U$, a pair of approximation operators, ${}^\triangle$, ${}^\triangledown : L^U \longrightarrow \mathcal{P}(M)$ defined by

$$X^\triangle = \{b \in M | \bigvee\nolimits_{x \in U}(X(x) \otimes R(x, b)) \geq \delta\},$$
$$X^\triangledown = \{b \in M | \bigwedge\nolimits_{x \in U}(R(x, b) \to X(x)) \geq \delta\}.$$

Analogously, for any $B \subseteq L^M$, a pair of approximation operators, ${}^{\triangle}$, ${}^{\triangledown}$: $L^M \longrightarrow \mathcal{P}(U)$ defined by

$$\begin{aligned} B^{\triangle} &= \{x \in U | \bigvee_{a \in M}(B(a) \otimes R(x,a)) \geq \delta\}, \\ B^{\triangledown} &= \{x \in U | \bigwedge_{a \in M}(R(x,a) \rightarrow B(a)) \geq \delta\}. \end{aligned}$$

The following property list the basic properties of the adjoint pair of operators.

Theorem 1. *Let (U, M, R) be a fuzzy formal context, $X, X_1, X_2 \in \mathcal{P}(U)$, B, B_1, $B_2 \in \mathcal{P}(M)$, then*

$$\begin{array}{ll} \text{(i)} & X_1 \subseteq X_2 \Rightarrow X_1^{\Diamond} \subseteq X_2^{\Diamond},\ X_2^{\Box} \subseteq X_1^{\Box}; \\ \text{(ii)} & B_1 \subseteq B_2 \Rightarrow B_1^{\Diamond} \subseteq B_2^{\Diamond},\ B_2^{\Box} \subseteq B_1^{\Box}; \\ \text{(iii)} & X \subseteq X^{\Diamond\triangledown},\ B \subseteq B^{\Diamond\triangledown}. \end{array}$$

Proof. (i) Since $X_1 \subseteq X_2$, then $\bigvee_{x \in X_1} R(x,a) \leq \bigvee_{x \in X_2} R(x,a)$. For any $a \in M$, from Lemma 1 (9) we have

$$\begin{aligned} X_1^{\Diamond}(a) &= \bigvee_{x \in X_1}(\delta \otimes R(x,a)) \\ &= \delta \otimes \bigvee_{x \in X_1} R(x,a) \\ &\leq \delta \otimes \bigvee_{x \in X_2} R(x,a) \\ &= \bigvee_{x \in X_2}(\delta \otimes R(x,a)) \\ &= X_2^{\Diamond}(a). \end{aligned}$$

It is evident that $X_2^{\Box} \subseteq X_1^{\Box}$.

(ii) It is similar to the proof of (i).

(iii) On one hand,

$$\begin{aligned} X^{\Diamond\triangledown} &= \{x \in U | \bigwedge_{a \in M}(R(x,a) \rightarrow X^{\Diamond}(a)) \geq \delta\} \\ &= \{x \in U | \bigwedge_{a \in M}(R(x,a) \rightarrow (\bigvee_{y \in X}(\delta \otimes R(y,a))) \geq \delta\}. \end{aligned}$$

On the other hand, for any $x \in X$, we have

$$\begin{aligned} \bigwedge_{a \in M}(R(x,a) \rightarrow (\bigvee_{y \in X}(\delta \otimes R(y,a)))) &\geq \bigwedge_{a \in M}(R(x,a) \rightarrow (\delta \otimes R(x,a))) \\ &\geq \bigwedge_{a \in M} \delta. \end{aligned}$$

Thus, $x \in X^{\Diamond\triangledown}$. Then, $X \subseteq X^{\Diamond\triangledown}$. And $B \subseteq B^{\Diamond\triangledown}$ can be obtained by the similar proof.

Theorem 2. *Let (U, M, R) be a fuzzy formal context, $X, X_1, X_2 \in L^U$, B, B_1, $B_2 \in L^M$, then*

$$\begin{array}{ll} \text{(i)} & X_1 \subseteq X_2 \Rightarrow X_1^{\triangle} \subseteq X_2^{\triangle},\ X_1^{\triangledown} \subseteq X_2^{\triangledown}; \\ \text{(ii)} & B_1 \subseteq B_2 \Rightarrow B_1^{\triangle} \subseteq B_2^{\triangle},\ B_1^{\triangledown} \subseteq B_2^{\triangledown}; \\ \text{(iii)} & X^{\triangledown\Diamond} \subseteq X,\ B^{\triangledown\Diamond} \subseteq B. \end{array}$$

Proof. (i) and (ii) follows immediately from Lemma 1 (2) and Lemma 1 (6).

(iii) For any $x \in X$, we have

$$\begin{aligned} X^{\nabla\Diamond}(x) &= \bigvee\nolimits_{a\in X^{\nabla}}(\delta\otimes R(x,a)) \\ &= \bigvee\nolimits_{a\in\{b\in M|\bigwedge_{y\in U}(R(y,b)\to X(y))\geq\delta\}}(\delta\otimes R(x,a)) \\ &= \bigvee\nolimits_{a\in\{b\in M|\forall y\in U, R(y,b)\otimes\delta\leq X(y)\}}(\delta\otimes R(x,a)) \\ &\leq X(x). \end{aligned}$$ □

By the similar proof we have $B^{\nabla\Diamond} \subseteq B$.

Theorem 3. *Let (U, M, R) be a fuzzy formal context, $X, X_1, X_2 \in \mathcal{P}(U)$, B, B_1, $B_2 \in \mathcal{P}(M)$, then*

$$\begin{array}{ll} \text{(i)} & X^{\Diamond\nabla\Diamond} = X^{\Diamond},\ B^{\Diamond\nabla\Diamond} = B^{\Diamond}; \\ \text{(ii)} & (X_1 \cap X_2)^{\Box} = X_1^{\Box} \wedge X_2^{\Box},\ (X_1 \cup X_2)^{\Diamond} = X_1^{\Diamond} \vee X_2^{\Diamond}; \\ \text{(iii)} & (B_1 \cap B_2)^{\Box} = B_1^{\Box} \wedge B_2^{\Box},\ (B_1 \cup B_2)^{\Diamond} = B_1^{\Diamond} \vee B_2^{\Diamond}. \end{array}$$

Proof. (i) On one hand, from Theorem 1 (i) and (iii) we have $X^{\Diamond} \leq X^{\Diamond\nabla\Diamond}$; on the other hand, from Theorem 2 (iii) we have $X^{\Diamond} \geq (X^{\Diamond})^{\nabla\Diamond}$. Thus, we have $X^{\Diamond} = X^{\Diamond\nabla\Diamond}$. And, $B^{\Diamond\nabla\Diamond} = B^{\Diamond}$ can be obtained by the similar proof.

(ii) For any $a \in M$,

$$\begin{aligned} (X_1 \cap X_2)^{\Box}(a) &= \bigwedge\nolimits_{x\in X_1\cap X_2}(R(x,a) \to \delta) \\ &= (\bigwedge\nolimits_{x\in X_1}(R(x,a) \to \delta)) \wedge (\bigwedge\nolimits_{x\in X_2}(R(x,a) \to \delta)) \\ &= X_1^{\Box}(a) \wedge X_2^{\Box}(a). \end{aligned}$$

$$\begin{aligned} (X_1 \cup X_2)^{\Diamond}(a) &= \bigvee\nolimits_{x\in X_1\cup X_2}(\delta \otimes R(x,a)) \\ &= (\bigvee\nolimits_{x\in X_1}(\delta \otimes R(x,a))) \vee (\bigvee\nolimits_{x\in X_2}(\delta \otimes R(x,a))) \\ &= X_1^{\Box}(a) \vee X_2^{\Box}(a). \end{aligned}$$

(iii) It is similar to the proof of (ii).

Theorem 4. *Let (U, M, R) be a fuzzy formal context, $X, X_1, X_2 \in L^U$, B, B_1, $B_2 \in L^M$, then*

$$\begin{array}{ll} \text{(i)} & X^{\nabla\Diamond\nabla} = X^{\nabla},\ B^{\nabla\Diamond\nabla} = B^{\nabla}; \\ \text{(ii)} & (X_1 \wedge X_2)^{\nabla} = X_1^{\nabla} \cap X_2^{\nabla},\ (X_1 \vee X_2)^{\triangle} = X_1^{\triangle} \cup X_2^{\triangle}; \\ \text{(iii)} & (B_1 \wedge B_2)^{\nabla} = B_1^{\nabla} \cap B_2^{\nabla},\ (B_1 \vee B_2)^{\triangle} = B_1^{\triangle} \cup B_2^{\triangle}. \end{array}$$

Proof. (i) On one hand, from Theorem 2 (i) and (iii) we have $X^{\nabla\Diamond\nabla} \subseteq X^{\nabla}$; on the other hand, from Theorem 1 (iii) we have $(X^{\nabla})^{\Diamond\nabla} \supseteq X^{\nabla}$. Thus, we have $X^{\nabla\Diamond\nabla} = X^{\nabla}$. And, $B^{\nabla\Diamond\nabla} = B^{\nabla}$ can be obtained by the similar proof.

(ii) From Lemma 1 (4), we have

$$\begin{aligned} (X_1 \wedge X_2)^{\nabla} &= \{b \in M| \bigwedge\nolimits_{x\in U}(R(x,b) \to (X_1 \wedge X_2)(x)) \geq \delta\} \\ &= \{b \in M| \bigwedge\nolimits_{x\in U}((R(x,b) \to X_1(x)) \wedge (R(x,b) \to X_2(x))) \geq \delta\} \\ &= \{b \in M|(\bigwedge\nolimits_{x\in U}(R(x,b) \to X_1(x)) \wedge (\bigwedge\nolimits_{x\in U}(R(x,b) \to X_2(x)))) \geq \delta\} \\ &= X_1^{\nabla} \cap X_2^{\nabla}. \end{aligned}$$

From Lemma 1 (9), we have

$$\begin{aligned}(X_1 \vee X_2)^{\triangle} &= \{b \in M| \bigvee_{x\in U}((X_1 \vee X_2)(x) \otimes R(x,b)) \geq \delta\}\\ &= \{b \in M| \bigvee_{x\in U}((X_1(x) \otimes R(x,b)) \vee (X_1(x) \otimes R(x,b))) \geq \delta\}\\ &= \{b \in M|(\bigvee_{x\in U}(X_1(x) \otimes R(x,b)) \vee (\bigvee_{x\in U}(X_1(x) \otimes R(x,b)))) \geq \delta\}\\ &= X_1^{\triangle} \cup X_2^{\triangle}.\end{aligned}$$

(iii) It is similar to the proof of (ii).

A pair (X, B), $X \in \mathcal{P}(U)$, $B \in L^M$, is called a property oriented crisp-fuzzy concept if $X^{\Diamond} = B$ and $B^{\triangledown} = X$. The set of objects X is called the extension of the property oriented crisp-fuzzy concept (X, B), and the fuzzy set of properties is called the intension. For a set of objects $X \subseteq U$ and a fuzzy set of attributes $B \subseteq L^M$, from Theorem 1 (iii) and Theorem 4 (iii) we have that $(X^{\Diamond\triangledown}, X^{\Diamond})$ and $(B^{\triangledown}, B^{\triangledown\Diamond})$ are property oriented crisp-fuzzy concepts, and we have different level property oriented fuzzy-crisp concepts by different precision level δ. For two property oriented crisp-fuzzy concepts (X_1, B_1) and (X_2, B_2), $(X_1, B_1) \leq (X_2, B_2)$, if and only if $X_1 \subseteq X_2$ (or equivalently, $B_1 \subseteq B_2$). The set of all property oriented crisp-fuzzy concepts forms a complete lattice which is denoted by $L_\delta(U, \widetilde{M}, R)$. The meet and join of the lattice is given by:

$$\begin{aligned}(X_1, B_1) \vee (X_2, B_2) &= ((X_1 \cup X_2)^{\Diamond\triangledown}, B_1 \cup B_2)\\ &= ((B_1 \cup B_2)^{\triangledown}, B_1 \cup B_2);\\ (X_1, B_1) \wedge (X_2, B_2) &= (X_1 \cap X_2,\ (B_1 \cap B_2)^{\triangledown\Diamond})\\ &= (X_1 \cap X_2, (X_1 \cap X_2)^{\Diamond}).\end{aligned}$$

A pair (X, B), $X \in L^U$, $B \in \mathcal{P}(M)$, is called an object oriented fuzzy-crisp concept if $X^{\triangledown} = B$ and $B^{\Diamond} = X$. The fuzzy set of objects X is called the extension of the object oriented fuzzy-crisp concept (X, B), and the set of properties is called the intension. For a fuzzy set of objects $X \in L^U$ and a set of attributes $B \subseteq M$, from Theorem 2 (iii) and Theorem 3 (iii) we have that $(X^{\triangledown\Diamond}, X^{\triangledown})$ and $(B^{\Diamond}, B^{\Diamond\triangledown})$ are object oriented fuzzy-crisp concepts, and we have different level object oriented fuzzy-crisp concepts by different precision level δ.For two object oriented fuzzy-crisp concepts (X_1, B_1) and (X_2, B_2), $(X_1, B_1) \leq (X_2, B_2)$, if and only if $X_1 \subseteq X_2$ (or equivalently, $B_1 \subseteq B_2$). The set of all object oriented fuzzy-crisp concepts forms a complete lattice which is denoted by $L_\delta(\widetilde{U}, M, R)$ with meet and join defined by:

$$\begin{aligned}(X_1, B_1) \vee (X_2, B_2) &= (X_1 \cup X_2, (B_1 \cup B_2)^{\Diamond\triangledown})\\ &= (X_1 \cup X_2, (X_1 \cup X_2)^{\triangledown});\\ (X_1, B_1) \wedge (X_2, B_2) &= ((X_1 \cap X_2)^{\triangledown\Diamond},\ B_1 \cap B_2)\\ &= ((B_1 \cap B_2)^{\Diamond}, B_1 \cap B_2).\end{aligned}$$

Example 3. In *Example 2*, let → be the Lukasiewicz implication, ie. for $x, y \in [0, 1]$,

$$x \to y = \begin{cases} 1, & x \leq y,\\ 1 - x + y, & x > y;\end{cases}$$

$$x \otimes y = (x + y - 1) \vee 0.$$

When $\delta = 0.9$, by computation we obtain the property oriented crisp-fuzzy concepts presented in Table 2.

Table 2. The property oriented crisp-fuzzy concepts for $\delta = 0.9$

Label	Objects × properties
FC_0	$\emptyset \times \{a/0.0, b/0.0, c/0.0, d/0.0\}$
FC_1	$\{x_1\} \times \{a/0.4, b/0.4, c/0.9, d/0.6\}$
FC_2	$\{x_3\} \times \{a/0.5, b/0.4, c/0.7, d/0.9\}$
FC_3	$\{x_4\} \times \{a/0.8, b/0.2, c/0.7, d/0.7\}$
FC_4	$\{x_1, x_3\} \times \{a/0.5, b/0.4, c/0.9, d/0.9\}$
FC_5	$\{x_1, x_4\} \times \{a/0.8, b/0.4, c/0.9, d/0.7\}$
FC_6	$\{x_2, x_4\} \times \{a/0.8, b/0.2, c/0.7, d/0.8\}$
FC_7	$\{x_1, x_2, x_4\} \times \{a/0.8, b/0.4, c/0.9, d/0.8\}$
FC_8	$\{x_2, x_3, x_4\} \times \{a/0.8, b/0.4, c/0.7, d/0.9\}$
FC_9	$\{x_1, x_2, x_3, x_4\} \times \{a/0.8, b/0.4, c/0.9, d/0.9\}$

Example 4. Continuing from *Example 3*, when $\delta = 0.9$, by calculation we obtain the object oriented fuzzy-crisp concepts presented in Table 3.

Table 3. TThe object oriented fuzzy-crisp concepts for $\delta = 0.9$

Label	Objects × properties
FC_0	$\{x_1/0.0, x_2/0.0, x_3/0.0, x_4/0.0\} \times \emptyset$
FC_1	$\{x_1/0.3, x_2/0.1, x_3/0.3, x_4/0.1\} \times \{b\}$
FC_2	$\{x_1/0.3, x_2/0.7, x_3/0.4, x_4/0.7\} \times \{a, b\}$
FC_3	$\{x_1/0.8, x_2/0.6, x_3/0.6, x_4/0.6\} \times \{b, c\}$
FC_4	$\{x_1/0.5, x_2/0.7, x_3/0.8, x_4/0.6\} \times \{b, d\}$
FC_5	$\{x_1/0.8, x_2/0.7, x_3/0.6, x_4/0.7\} \times \{a, b, c\}$
FC_6	$\{x_1/0.5, x_2/0.7, x_3/0.8, x_4/0.7\} \times \{a, b, d\}$
FC_7	$\{x_1/0.8, x_2/0.7, x_3/0.8, x_4/0.6\} \times \{b, c, d\}$
FC_8	$\{x_1/0.8, x_2/0.7, x_3/0.8, x_4/0.7\} \times \{a, b, c, d\}$

3 Relationship Between Different Precision Level Concepts Lattices

In the following, we denote the extents of the lattices $L_\delta(U, \widetilde{M}, R)$ and $L_\delta(\widetilde{U}, M, R)$ by $L_\delta^U(U, \widetilde{M}, R)$ and $L_\delta^U(\widetilde{U}, M, R)$. similarity, we denote the intents of the lattices $L_\delta(U, \widetilde{M}, R)$ and $L_\delta(\widetilde{U}, M, R)$ by $L_\delta^M(U, \widetilde{M}, R)$ and $L_\delta^M(\widetilde{U}, M, R)$.

We denote $^{\Diamond_\delta}$ as the operator $^{\Diamond}$ with the precision δ, and similarly with the operators $^{\Box}$,$^{\triangle}$ and $^{\triangledown}$.

Theorem 5. *Let (U, M, R) be a fuzzy formal context, $X \in \mathcal{P}(U)$ and $B \in \mathcal{P}(M)$. If $0 \leq \delta_1 \leq \delta_2 \leq 1$, then*

$$X^{\Diamond_{\delta_1}} \leq X^{\Diamond_{\delta_2}},\ X^{\Box_{\delta_1}} \leq X^{\Box_{\delta_2}},\ B^{\Diamond_{\delta_1}} \leq B^{\Diamond_{\delta_2}},\ B^{\Box_{\delta_1}} \leq B^{\Box_{\delta_2}}.$$

Proof. For any $x \in X$, $a \in M$,

$$\begin{aligned}\delta_1 \leq \delta_2 &\Longleftrightarrow \delta_1 \otimes R(x,a) \leq \delta_1 \otimes R(x,a)\\ &\Longleftrightarrow \bigvee_{x \in X}(\delta_1 \otimes R(x,a)) \leq \bigvee_{x \in X}(\delta_2 \otimes R(x,a))\\ &\Longleftrightarrow X^{\Diamond_{\delta_1}}(a) \leq X^{\Diamond_{\delta_2}}(a).\end{aligned}$$

From Lemma 1 (2) we have

$$\begin{aligned}\delta_1 \leq \delta_2 &\Longleftrightarrow R(x,a) \to \delta_1 \leq R(x,a) \to \delta_2\\ &\Longleftrightarrow \bigwedge_{x \in X}(R(x,a) \to \delta_1) \leq \bigwedge_{x \in X}(R(x,a) \to \delta_2)\\ &\Longleftrightarrow X^{\Box_{\delta_1}}(a) \leq X^{\Box_{\delta_2}}(a).\end{aligned}$$

By the similar proof we have $B^{\Diamond_{\delta_1}} \leq B^{\Diamond_{\delta_2}},\ B^{\Box_{\delta_1}} \leq B^{\Box_{\delta_2}}$.

Theorem 6. *Let (U, M, R) be a fuzzy formal context, $X \in L^U$ and $B \in L^M$. If $0 \leq \delta_1 \leq \delta_2 \leq 1$, then*

$$X^{\triangle_{\delta_1}} \supseteq X^{\triangle_{\delta_2}},\ X^{\nabla_{\delta_1}} \supseteq X^{\nabla_{\delta_2}},\ B^{\triangle_{\delta_1}} \supseteq B^{\triangle_{\delta_2}},\ B^{\nabla_{\delta_1}} \supseteq B^{\nabla_{\delta_2}}.$$

Proof. Since $\delta_1 \leq \delta_2$, then

$$\begin{aligned}\forall\ b \in X^{\triangle_{\delta_2}} &\Longrightarrow \bigvee_{x \in U}(X(x) \otimes R(x,b)) \geq \delta_2\\ &\Longrightarrow \bigvee_{x \in U}(X(x) \otimes R(x,b)) \geq \delta_1\\ &\Longrightarrow b \in X^{\triangle_{\delta_1}}.\end{aligned}$$

Which implies $X^{\triangle_{\delta_2}} \subseteq X^{\triangle_{\delta_1}}$.

$$\begin{aligned}\forall\ b \in X^{\nabla_{\delta_2}} &\Longrightarrow \bigwedge_{x \in U}(R(x,b) \to X(x)) \geq \delta_2\}\\ &\Longrightarrow \bigwedge_{x \in U}(R(x,b) \to X(x)) \geq \delta_1\}\\ &\Longrightarrow b \in X^{\nabla_{\delta_1}}.\end{aligned}$$

Which implies $X^{\nabla_{\delta_2}} \subseteq X^{\nabla_{\delta_1}}$. And, $B^{\triangle_{\delta_1}} \supseteq B^{\triangle_{\delta_2}},\ B^{\nabla_{\delta_1}} \supseteq B^{\nabla_{\delta_2}}$ can be obtained by the similar proof.

Theorem 7. *Let $L_\delta(U, \widetilde{M}, R)$ be a crisp-fuzzy concept lattice. If $0 \leq \delta_1 \leq \delta_2 \leq 1$, then*

$$L^U_{\delta_1}(U, \widetilde{M}, R) \subseteq L^U_{\delta_2}(U, \widetilde{M}, R).$$

Proof. For any $(X, B) \in L_{\delta_1}(U, \widetilde{M}, R)$, we have $X^{\Diamond_{\delta_1}\nabla_{\delta_1}} = X$. It is evident that $(X^{\Diamond_{\delta_2}\nabla_{\delta_2}}, X^{\Diamond_{\delta_2}}) \in L_{\delta_2}(U, \widetilde{M}, R)$. We are to prove $X^{\Diamond_{\delta_2}\nabla_{\delta_2}} = X$. On one hand, $X \subseteq X^{\Diamond_{\delta_2}\nabla_{\delta_2}}$; on the other hand,

$$\begin{aligned} X^{\Diamond_{\delta_2}\nabla_{\delta_2}} &= \{x \in U|\bigwedge_{a\in M}(R(x,a) \to X^{\Diamond_{\delta_2}}(a)) \geq \delta_2\} \\ &= \{x \in U|\bigwedge_{a\in M}(R(x,a) \to \bigvee_{y\in X}(\delta_2 \otimes R(y,a)) \geq \delta_2\} \\ &= \{x \in U|\forall\ a \in M, R(x,a) \to \bigvee_{y\in X}(\delta_2 \otimes R(y,a) \geq \delta_2\} \\ &= \{x \in U|\forall\ a \in M, R(x,a) \otimes \delta_2 \leq \bigvee_{y\in X}(\delta_2 \otimes R(y,a)\} \\ &= \{x \in U|\forall\ a \in M, R(x,a) \otimes \delta_2 \leq \delta_2 \otimes (\bigvee_{y\in X} R(y,a))\} \\ &\subseteq \{x \in U|\forall\ a \in M, R(x,a) \otimes \delta_1 \leq \delta_1 \otimes (\bigvee_{y\in X} R(y,a))\} \\ &= X^{\Diamond_{\delta_1}\nabla_{\delta_1}} \\ &= X. \end{aligned}$$

Thus, we have $X^{\Diamond_{\delta_2}\nabla_{\delta_2}} = X$, which implies $L^U_{\delta_1}(U, \widetilde{M}, R) \subseteq L^U_{\delta_2}(U, \widetilde{M}, R)$.

Theorem 8. *Let* $L_\delta(\widetilde{U}, M, R)$ *be a fuzzy-crisp concept lattice. If* $0 \leq \delta_1 \leq \delta_2 \leq 1$*, then*

$$L^M_{\delta_2}(\widetilde{U}, M, R) \subseteq L^M_{\delta_1}(\widetilde{U}, M, R).$$

Proof. It is similar to the proof of Theorem 7.

4 Conclusions

In the paper, we introduce two pairs operators in fuzzy formal contexts. Based on the proposed operators, we present two types of generalized variable precision formal concepts, i.e. property oriented crisp-fuzzy concepts and object oriented fuzzy-crisp concepts. The relationship between different precision level generalized concepts lattices are also discussed in details. By different precision level, we have different level generalized formal concepts.

Variable precision formal concept analysis is an important tool that can be applied to deal with uncertainty contained in conceptual fuzzy data analysis and knowledge processing. The proposed generalized variable precision formal concepts is a important complement to formal concept analysis. The applications of the proposed formal concepts are our next research.

Acknowledgments

This work is supported by the Postdoctoral Foundation of China (20060390467) and the Technology Program of Jiangxi Education Office of China ([2006] 321 and [2007] 275).

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-Boston (1982)
2. Gediga, B., Wille, R.: Formal Concept Analysis. Mathematic Foundations. Springer, Heidelberg (1999)

3. Carpineto, C., Romano, G.: A lattice conceptual clustering system and its application to browsing retrieval. Mach. Learning 10, 95–122 (1996)
4. Faid, M., Missaoi, R., Godin, R.: Mining complex structures using context concatenation in formal concept analysis. International KRUSE Symposium, Vancouver, BC (August 11–13, 1997)
5. Godin, R., Missaoi, R.: An incremental concept formation approach for learning from databases. Theoret. Comput. Sci. 133, 387–419 (special issues on Formal Methods in Databases and Software Engineering) (1994)
6. Harms, S.K., Deogum, J.S.: Sequential association rule mining with time lags. J. Intell. Inform. Systems 22(1), 7–22 (2004)
7. Wille, R.: Knowledge acquisition by methods of formal concept analysis. In: Diday, E. (ed.) Data Analysis, Learning Symbolic and Numeric Knowledge, Nova Science, pp. 365–380 (1989)
8. Burusco, A., Fuentes-Gonzalez, R.: Concept lattices defined from implication operators. Fuzzy Sets and systems 114(3), 431–436 (1998)
9. Elloumi, S., Jaam, J., Hasnah, A., Jaoua, A., Nafkha, I.: A multi-level conceptual data reduction approach based on the Lukasiewicz implication. Information Sciences 163, 253–262 (2004)
10. Belohlavek, R.: Fuzzy closure operators. I. J. Math. Anal. Appl. 262, 473–489 (2001)
11. Belohlavek, R.: Concept lattice and order in fuzzy logic. Annals of pure and Apll. Logic 128(1-3), 277–298 (2004)
12. Popescu, A.: A general approach to fuzzy concept. Math. Logic Quaterly 50(3), 1–17 (2001)
13. Georgescu, G., Popescu, A.: Non-dual fuzzy connections. Archive for Mathematic Logic 43(8), 1009–1039 (2004)
14. Elloumi, S., Jaoua, A.: Automatic classification using fuzzy concepts. In: Proc.JCIS, Atlantic City, USA, vol. 1, pp. 276C279 (2000)
15. Jaoua, A., Elloumi, S.: Galois connection, formal concepts and Galois lattice in real relations: application in a real classifier. The Journal of Systems and Software 60, 149–163 (2002)
16. Fan, S.Q.: Fuzzy Concept Lattice and Variable Precision Concept Lattice, Ph. M. Thesis, Faculty of Science, Xi'an Jiaotong University (2006)
17. Yao, Y.Y.: A comparative study of formal concept analysis and rough set theory in Data analysis, Rough Sets and Current Trends in Computing. In: Proceedings of 3rd International Conference, RSCT'04 (2004)
18. Yao, Y.Y.: Concept lattices in rough set theory. In: Proceedings of, Annual Meeting of the North American Fuzzy Information Processing Society, pp. 796–801 (2004)
19. Gediga, G., Duntsch, I.: Modal-style operators in qualitative data analysis. In: Proceedings of the 2002 IEEE International Conference on Data Mining, pp. 155–162 (2002)

Dynamics of Approximate Information Fusion

Patrick Doherty[1], Barbara Dunin-Kęplicz[2], and Andrzej Szałas[1,3]

[1] Dept. of Computer and Information Science, Linköping University
SE-581 83 Linköping, Sweden
[2] Institute of Informatics, Warsaw University, Warsaw, Poland,
ICS, Polish Academy of Sciences, Warsaw, Poland
and NIAS, 2242 PR Wassenaar, The Netherlands
[3] The University of Economics and Computer Science, Olsztyn, Poland

Abstract. The multi-agent system paradigm has proven to be a useful means of abstraction when considering distributed systems with interacting components. It is often the case that each component may be viewed as an intelligent agent with specific and often limited perceptual capabilities. It is also the case that these agent components may be used as information sources and such sources may be aggregated to provide global information about particular states, situations or activities in the embedding environment. This paper investigates a framework for information fusion based on the use of generalizations of rough set theory and the use of dynamic logic as a basis for aggregating similarity relations among objects where the similarity relations represent individual agents perceptual capabilities or limitations. As an added benefit, it is shown how this idea may also be integrated into description logics.

Keywords: approximate reasoning, similarity relation, information fusion.

1 Introduction

Information fusion is the process of exploiting various information sources to provide improved and possibly complete knowledge about a situation in question. Usually information sources such as agents or sensors are local and limited in capabilities, so obtaining a global description of a situation requires special aggregation operations. Many such operations have been considered in the literature. In [1] the following classes of aggregation operations have been found to be frequently used:

- operations generalizing the notion of conjunction (corresponding to the minimum)
- operations generalizing the notion of disjunction (corresponding to the maximum)
- averaging operations (providing values between the minimum and the maximum).

Since the process of information fusion is applied in so many contexts and situations, it would be desirable to have a uniform framework for defining aggregation operations on information sources such as agents or sensors which are characterized by limited perceptual capabilities. In addition, one would also like to reason about the resulting global, fused knowledge in a principled manner. In the current paper we make an attempt to contribute to the development of such a framework.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 668–677, 2007.

We will focus on fusing information obtained from imprecise measurements. According to [1], merging uncertain observations usually addresses the problem of finding the most plausible values of an observed parameter or the most credible description of a situation. In the paper, we focus on finding the most credible description of a situation as well as a description of possible situations, assuming that the imprecision of measurement devices or an agent's limited perceptual capabilities is modelled by similarity spaces [2], as considered, e.g., in [3,4] (see also the book [5]). As a side effect, the proposed approach can be combined in a natural way with the approaches for higher-level information fusion, provided in [3,4].

The approach proposed in this paper depends on using dynamic logic [6] to specify aggregation operations and reason about them[1]. To do this, we first reinterpret dynamic logic in such a way that rather than considering programs and their aggregation, we consider similarity relations and their aggregation instead. This makes sense since one would like to associate one or more similarity relations with perspective agents to model their sensory or perceptual limitations.

In the paper we will talk about *approximate* information rather than *uncertain* information, with the understanding that approximate information is obtained from uncertain information by applying approximation operators. This allows us to propose a novel interpretation of a well known formalism, which is well developed with underlying reasoning techniques which may be capitalized upon. Another positive side-effect of the proposed approach is that the basic specification of aggregated similarity relations can be directly translated into the description logic formalism (for an introduction to description logics see, e.g., [9,10]). This is useful, since description logics are one of the most frequently used knowledge representation formalisms and provide a logical basis for a variety of well known paradigms.

We will also show that given a specific type of similarity spaces, one can substantially reduce the complexity of reasoning and apply the reasoning calculus even in demanding real-time autonomous systems consisting of many independent components (agents).

The paper is structured as follows. In Section 2 we present similarity spaces. In Section 3, we provide an interpretation of dynamic logic as a calculus for similarity-based information fusion. In Section 4 ,we consider a scenario based on the use of mini UAVs (UAV is an acronym for Unmanned Aerial Vehicle). Section 5 shows the relation between the proposed approach and description logics. Finally, Section 6 concludes the paper.

2 Similarity Spaces

There is a natural generalization of relations, where instead of crisp relations one considers rough sets and relations, as introduced in [11]. These can further be generalized to approximate relations based on similarity spaces. In order to approximate relations one uses here a covering of the underlying domain by similarity-based neighborhoods (see, e.g., [12,2]). In this approach the lower and upper approximations of relations are defined via neighborhoods rather than equivalence classes which are used with

[1] A study of various operations on relations related to this context can also be found in [7]. A rough set-based approach to multiagent systems has also been considered in [8].

rough sets. Approximate relations and similarity spaces have been shown to be quite versatile in many application areas requiring the use of approximate knowledge structures [5,13,14].

There are many choices that can be made concerning the constraints one might want to place on the similarity relation used to define upper and lower approximations. For example, one might not want the relation be transitive since similar objects do not naturally chain in a transitive manner. Many of these issues are discussed in the context of rough sets (see, e.g., [11,15,16,17,18]). In order to represent arbitrary notions of similarity in a universe of individuals, similarity relations have no initial constraints.

Definition 2.1. *By a* similarity space *we mean any pair* $\langle U, \sigma\rangle$, *where* U *is a non-empty set and* $\sigma \subseteq U \times U$. *By a* neighborhood *of* u *wrt* σ *we mean* $n^{\sigma}(u) \stackrel{\text{def}}{=} \{u' \in U \mid \sigma(u, u')\}$. *For* $A \subseteq U$, *the* lower and upper approximation of A wrt σ, *denoted respectively by* A^{+}_{σ} *and* $A^{\oplus}_{\sigma}$, *are defined by* $A^{+}_{\sigma} = \{u \in U \colon n^{\sigma}(u) \subseteq A\}$, $A^{\oplus}_{\sigma} = \{u \in U \colon n^{\sigma}(u) \cap A \neq \emptyset\}$. *We also define* $A^{-}_{\sigma} \stackrel{\text{def}}{=} -A^{\oplus}_{\sigma}$ *and* $A^{\pm}_{\sigma} \stackrel{\text{def}}{=} -A^{+}_{\sigma} \cap -A^{-}_{\sigma}$. ◁

Let $S = \langle U, \sigma\rangle$ be a similarity space and let $A \subseteq U$. Then an alternative way to define upper and lower approximations used throughout this paper is as follows:

$$\begin{aligned}
A^{+}_{S} &= \{a \in A \mid \forall b\,[\sigma(a,b) \rightarrow b \in A]\}\\
A^{\oplus}_{S} &= \{a \in A \mid \exists b\,[\sigma(a,b) \wedge b \in A]\}\\
A^{-}_{S} &= \{a \in A \mid \forall b[\sigma(a,b) \rightarrow b \notin A]\}\\
A^{\pm}_{S} &= \{a \in A \mid \exists b \exists c[\sigma(a,b) \wedge\ \sigma(a,c) \wedge b \in A \wedge c \notin A]\}.
\end{aligned}$$

3 Interpretation of Dynamic Logic as Calculus for Approximate Information Fusion

3.1 Dynamic Logic

PDL has been introduced as a tool for expressing properties of programs and reasoning about them (see, e.g., [6]).

Syntax. The language of PDL consists of formulas and programs.[2] Let V_0 and P_0 be countable sets of propositions and atomic programs, respectively. Programs and formulas are built inductively from V_0 and P_0 by using *propositional connectives* ($\neg, \vee, \wedge, \rightarrow, \equiv$), *modalities* indexed by programs $[p]$, $\langle p\rangle$, *program operators* $;\,, \cup\,, \cap\,,^{*}\,,^{-1}$ and a *test operator* ?

If A, B are formulas and p, q are programs then $\neg A, A \vee B, [p]A, \langle p\rangle\, A$ are also formulas[3] and $p; q,\ p \cup q,\ p \cap q,\ p^{*},\ p^{-1},\ A?$ are programs.

In what follows we shall replace the term "programs" by "similarity relation symbols" and denote those symbols by σ with indices, if necessary, rather than by $p, q, \ldots$.

[2] We actually deal with the concurrent dynamic logic with converse, as we consider $\cap$ and $^{-1}$, too — see [19,6,20].

[3] Other typical propositional connectives, like $\wedge, \rightarrow, \equiv$ are defined as usual.

Semantics. The semantics of PDL is defined using the notion of Kripke frames of the form $\mathcal{K} = \langle U, \Pi, \Sigma \rangle$, where

- U is a set of objects
- $\Pi : V_0 \longrightarrow 2^U$ (for each proposition A, Π assigns a set of objects, for which A is TRUE)
- $\Sigma : P_0 \longrightarrow 2^{U \times U}$ (for each similarity relation symbol σ, Σ assigns a binary relation on U).

Let $\mathcal{K} = \langle U, \Pi, \Sigma \rangle$ be a Kripke structure, $a \in U$, A, B be formulas and σ_1, σ_2 be similarity relation symbols. The satisfiability relation is then defined as follows:

- $\mathcal{K}, a \models A$ iff $a \in \Pi(A)$, when $A \in V_0$
- $\mathcal{K}, a \models \neg A$ iff $\mathcal{K}, a \not\models A$
- $\mathcal{K}, a \models A \vee B$ iff $\mathcal{K}, a \models A$ or $\mathcal{K}, a \models B$
- $\mathcal{K}, a \models [\sigma]A$ iff for any $b \in U$ such that $\sigma(a, b)$ we have $\mathcal{K}, b \models [\sigma]A$
- $\mathcal{K}, a \models \langle\sigma\rangle\, A$ iff there is $b \in U$ such that $\sigma(a, b)$ and $\mathcal{K}, b \models [\sigma]A$,

where Σ is extended to cover all expressions on similarity relations recursively:

- $\Sigma(\sigma_1; \sigma_2) \stackrel{\text{def}}{=} \Sigma(\sigma_1) \circ \Sigma(\sigma_2)$, where $\circ$ is the composition of relations
- $\Sigma(\sigma_1 \cup \sigma_2) \stackrel{\text{def}}{=} \Sigma(\sigma_1) \cup \Sigma(\sigma_2)$, where $\cup$ on the righthand of equality is the union of relations
- $\Sigma(\sigma_1 \cap \sigma_2) \stackrel{\text{def}}{=} \Sigma(\sigma_1) \cap \Sigma(\sigma_2)$, where $\cap$ on the righthand of equality is the intersection of relations
- $\Sigma(\sigma^*) \stackrel{\text{def}}{=} (\Sigma(\sigma))^*$, where * on the righthand of equality is the transitive closure of a relation
- $\Sigma(\sigma^{-1}) \stackrel{\text{def}}{=} (\Sigma(\sigma))^{-1}$, where $^{-1}$ on the righthand of equality is the converse of a relation
- $\Sigma(A?) \stackrel{\text{def}}{=} \{\langle a, a \rangle \mid \mathcal{K}, a \models A\}$.

3.2 Useful Properties of Approximations Expressible in the Dynamic Logic

Observe that:

1. $[\sigma]A$ expresses the lower approximation of A wrt. σ, i.e., A^{+}_{σ}
2. $\langle\sigma\rangle\, A$ expresses the upper approximation of A wrt. σ, i.e., $A^{\oplus}_{\sigma}$.

Example 3.1.

1. $\big([\sigma_1]red \wedge \langle\sigma_2\rangle\, fast\big) \to car$ — if an object (according to σ_1) is surely red and (according to σ_2) its speed might be fast, then conclude that it is a car.
2. $\big(\langle\sigma_1\rangle\, hot \vee \langle\sigma_2\rangle\, hot\big) \to dangerous$ — if (according to σ_1 or to σ_2) an object might be hot, then conclude that it is dangerous. ◁

Expression σ^* defines the transitive closure of a relation, i.e., it makes an object o similar to an object o' if there is $k \geq 1$ and a chain of objects $o_1, \ldots, o_k$ such that $o_1 = o, o_k = o'$ and for all $1 \leq i \leq k-1$, o_i is similar to o_{i+1}, i.e., $\sigma(o_i, o_{i+1})$ holds.

Example 3.2. In rough set theory, the underlying similarity relations are equivalence relations so rather than considering arbitrary relations σ, one should consider their reflexive, symmetric and transitive closures, $(\sigma \cup id \cup \sigma^{-1})^*$, where id is the identity relation (e.g., defined by TRUE?). $\lhd$

Other typical operations on programs can be interpreted in multiagent setting as follows, where σ_1, σ_2 are similarity relations of two agents Ag_1, Ag_2:

- $\sigma_1 ; \sigma_2$ is the composition of relations, i.e., it makes an object o_1 similar to an object o_2 if agent Ag_1 finds o_1 similar to some object o' and Ag_2 considers objects o' and o_2 similar
- $\sigma_1 \cup \sigma_2$ is the set-theoretical union of relations, i.e., it makes an object o_1 similar to an object o_2 if at least one of agents Ag_1, Ag_2 considers objects o_1 and o_2 similar
- $\sigma_1 \cap \sigma_2$ is the set-theoretical intersection of relations, i.e., it makes an object o_1 similar to an object o_2 if both agents Ag_1 and Ag_2 consider objects o_1 and o_2 similar.

Example 3.3. Assume that agent Ag_1 observes objects o_1, o_2 and finds that they are of a similar color ($\sigma_1(o_1, o_2)$ holds). Assume further that Ag_2 observes objects o_2, o_3 and finds their color similar, too ($\sigma_2(o_2, o_3)$ holds). We are interested whether the color of o_1 is similar to the color of o_3. This property can be expressed by

$$\big((\sigma_1;\sigma_2) \cup (\sigma_2;\sigma_1)\big)(o_1, o_3).$$

Therefore, e.g., $\langle(\sigma_1;\sigma_2) \cup (\sigma_2;\sigma_1)\rangle\, red$ expresses that a given object might be red according to the fused knowledge of Ag_1 and Ag_2. On the other hand, $[\sigma_1 \cap \sigma_2] red$ expresses the fact that both agents find a given object to be red. $\lhd$

Test allows one to create conditional definitions, as shown in Example 3.4.

Example 3.4. In some circumstances the choice of similarity may depend on the state of the environment. For example, if the temperature is high, the observed process might be more sensitive on pressure (reflected by similarity σ_1) than in the case when the temperature is not high (reflected by similarity σ_2). Then

$$\big(high\_temp?;\sigma_1\big) \cup \big((\neg high\_temp)?;\sigma_2)\big)$$

expresses the guarded choice between σ_1 and σ_2 dependent on the temperature ("if the temperature is high then use σ_1 otherwise use σ_2"). $\lhd$

3.3 Reasoning over Concrete Similarity Spaces

The version of dynamic logic we consider is highly undecidable (see, e.g., [6]). Removing the $\cap$ operator makes the logic decidable but still quite complex. On the other hand, when we deal with concrete similarity spaces, the calculus can be used in a useful, practical manner.

The following example illustrates this idea.

Example 3.5. Consider a sensor measuring a given parameter, say ρ in the scale $[0, 10]$. Assume that the measurement error is not greater than 0.5. This means that we deal with a similarity space $\langle U, \sigma \rangle$, where $U \overset{\text{def}}{=} [0, 10]$ and $\sigma(x, y) \overset{\text{def}}{\equiv} |x - y| \leq 0.5$. Assume that the value of ρ is *acceptable* when $\rho \in [2.6, 6.8]$. Suppose that we are interested in evaluating formula $\langle \sigma \rangle\, acc$, where acc abbreviates "acceptable". Then it can easily be seen that $\langle \sigma \rangle\, acc \equiv \rho \in [2.1, 7.3]$. Similarly, $[\sigma] acc \equiv \rho \in [3.1, 6.3]$. One can, therefore use these definitions rather than modal operators. $\lhd$

When one deals with similarity spaces and relational or deductive databases, the situation becomes tractable, assuming that the underlying similarity relation is tractable. In order to compute the set of objects x satisfying $\langle \sigma \rangle\, A$ one just queries the database using first-order formula $\exists y[\sigma(x, y) \wedge A(y)]$, where y refers to objects (e.g., rows in database tables). Similarly, computing $[\sigma]A$ depends on supplying to the database the query $\forall y[\sigma(x, y) \rightarrow A(y)]$.

Operators on similarity relations allowed in the language of dynamic logic we deal with are first-order definable, except for * which, as transitive closure, also leads to tractable queries (see, e.g., [21]).

4 Mini UAV: A Case Study

4.1 A Scenario

In a given area traces of a chemical X and signs of radiation have been detected. There is a fleet of mini UAV available. They are of three kinds: capable to measure temperature, to measure radiation and, the most advanced and expensive ones, to measure the concentration of X.

Dependent on its concentration and the temperature of the environment, the chemical X can be relatively safe for humans, can be dangerous or even explosive. It also causes radiation. The goal for the fleet of UAV is to autonomously investigate the area and to report on a possible level of danger by aggregating their local knowledge.

4.2 Underlying Assumptions

Individual UAV's Level of Evaluation. The individual level concerns individual knowledge of UAVs. We assume that measurements of all UAV's are adjusted to the scale $[0, 1]$. Due to the accuracy of sensors the UAVs have certain perceptual limitations:

- the measurement error of temperature sensors is not greater than 0.02
- the measurement error of radiation sensors is not greater than 0.05
- the measurement error of concentration sensors is not greater than 0.01.

Perceptual limitations due to sensor limitations are modelled by similarity spaces with universe $[0, 1]$ and similarity relations:

$$\begin{aligned}
\sigma_t(x, y) &\overset{\text{def}}{=} |x - y| \leq 0.02 \text{ for the temperature}\\
\sigma_r(x, y) &\overset{\text{def}}{=} |x - y| \leq 0.05 \text{ for the radiation}\\
\sigma_c(x, y) &\overset{\text{def}}{=} |x - y| \leq 0.01 \text{ for the concentration of } X.
\end{aligned}$$

Table 1 provides conditions as to the evaluation of a situation from the perspective of a single UAV. Then, slightly abusing notation, for example we would have,

$$\begin{array}{lll}
\big(t \models [\sigma_t]safe\big) \equiv t \leq 0.43 & \text{and} & \big(t \models \langle\sigma_t\rangle\, expl\big) \equiv \textsc{False},\\
\big(r \models [\sigma_r]safe\big) \equiv r \leq 0.15 & \text{and} & \big(r \models \langle\sigma_r\rangle\, safe\big) \equiv r \leq 0.25,\\
\big(c \models [\sigma_c]\, expl\big) \equiv 0.71 < c & \text{and} & \big(c \models \langle\sigma_c\rangle\, expl\big) \equiv 0.69 < c.
\end{array}$$

Table 1. Individual conditions as to the danger.

Danger level	Temperature (t)	Radiation (r)	concentration of X (c)
safe for humans ($safe$)	$t \leq 0.45$	$r \leq 0.2$	$c \leq 0.4$
dangerous ($danger$)	$0.45 < t \leq 1.0$	$0.2 < r \leq 0.8$	$0.4 < c \leq 0.7$
explosive ($expl$)	never	$0.8 < r \leq 1$	$0.7 < c \leq 1.0$

Group Level of Evaluation. Here we, in fact, deal with distributed knowledge (see, e.g., [22,23] and in the context of multiagent systems [24,25]), since the actual evaluation of a situation requires fusing information from various UAVs. In our scenario we can, e.g., assume that the danger level is given in Table 2. In such a case we have to evaluate three attributes and therefore have to define a new similarity space on triples. For example, such a similarity space might be $\langle T, \sigma\rangle$, where

- $T \overset{\text{def}}{=} [0, 1] \times [0, 1] \times [0, 1]$
- $\sigma(\langle x, y, z\rangle, \langle x', y', z'\rangle) \overset{\text{def}}{=} \sigma_t(x, x') \wedge \sigma_r(y, y') \wedge \sigma_c(z, z')$.

Table 2. Group conditions as to the danger

Danger level	Group condition
safe for humans ($safe$)	$t \leq 0.3 \wedge r \leq 0.1 \wedge c \leq 0.4$
dangerous ($danger$)	$0.3 < t \leq 0.6 \wedge r \leq 0.5 \wedge c \leq 0.6$
explosive ($expl$)	in all other cases

Now, e.g.,

$$\begin{aligned}
&\big(\langle t, r, c\rangle \models [\sigma]safe\big) \equiv t \leq 0.28 \wedge r \leq 0.05 \wedge c \leq 0.39,\\
&\big(\langle t, r, c\rangle \models \langle\sigma\rangle\, danger\big) \equiv 0.28 < t \leq 0.62 \wedge r \leq 0.55 \wedge c \leq 0.61.
\end{aligned}$$

4.3 Tuning Safety Conditions to Circumstances

Observe that σ of the previous section can still be tuned. The underlying intuition is that by adding successive iterations of similarities one makes more and more situations similar to each other. For example, we might want to increase safety by iterating σ,

- σ itself may be used for the cheapest mini UAV
- $\sigma \cup (\sigma;\sigma)$ can be used for more expensive ones (measuring the concentration of X)
- $\sigma \cup (\sigma;\sigma) \cup (\sigma;\sigma;\sigma)$ can be used by humans, where a relatively highest safety is required.

5 A Relation to Description Logics

In this section we show that the approach we propose can easily be integrated with the description logic formalism.

Description logics refer to a family of formalisms concentrated around concepts, roles and individuals. There is a rich literature on description logics. For good survey papers consult [26], in particular papers [9,10,27] as well as the bibliography provided there.

Assume that sets of atomic concepts, $\mathcal{C}$, and of atomic roles, $\mathcal{R}$ are given. More complex concepts and roles are built by the use of constructors given in Table 3, where concepts are represented by unary predicates and roles are represented by binary predicates. Also, rather than using a formal semantics, we show a translation of the considered constructs into the classical first-order logic. Various description languages are distinguished by the constructors that allow to specify complex concepts and roles.

Table 3. Constructors used in description logics

Constructor name	Syntax	Translation (Tr)
concept name	A	$A(x)$
top	$\top$	TRUE
bottom	$\bot$	FALSE
complement ($\mathcal{C}$)	$\neg E$	$\neg Tr(E,x)$
conjunction	$E \sqcap F$	$Tr(E,x) \wedge Tr(F,x)$
union ($\mathcal{U}$)	$E \sqcup F$	$Tr(E,x) \vee Tr(F,x)$
universal quantification	$\forall R.E$	$\forall y[R(x,y) \rightarrow Tr(E,y)]$
existential quantification ($\mathcal{E}$)	$\exists R.E$	$\exists y[R(x,y) \wedge Tr(E,y)]$

Observe that

- $\exists\sigma.A$ mirrors the semantics of $\langle\sigma\rangle\, A$
- $\forall\sigma.A$ mirrors the semantics of $[\sigma]A$.

What then remains to show is how to deal with operations on similarity relations considered as roles. This has actually been done in [28] by introducing operations on roles corresponding to operations on programs[4]. The resulting formalism $\mathcal{TSL}$, is given by

[4] Transitive closure of roles, corresponding to $*$, has been studied also, e.g., in [29,30].

the same concept formation rules as provided in Table 3 together with allowing one to form new roles by applying operators $;\,, \sqcup, *, ^{-1}, ?$. In fact, we also need an additional operator $\sqcap$. Now the translation of dynamic logic formulas into description logic formulas, denoted below by T_δ, is rather immediate, where $\sharp \in \{^*, ^{-1}\}$, $\circ \in \{;\,, \cup, \cap\}$ and $;' \overset{\text{def}}{=} ;\,, \cup' \overset{\text{def}}{=} \sqcup, \cap' \overset{\text{def}}{=} \sqcap$:

$$\begin{array}{ll}
T_\delta(A) \overset{\text{def}}{=} A \text{ when } A \in V_0 & T_\delta(\langle C? \rangle\, A) \overset{\text{def}}{=} \exists C?.T_\delta(A) \\
T_\delta(\neg A) \overset{\text{def}}{=} \neg T_\delta(A) & T_\delta(\langle \sigma_1 \circ \sigma_2 \rangle\, A) \overset{\text{def}}{=} \exists \sigma_1 \circ' \sigma_2.T_\delta(A) \\
T_\delta(A \vee B) \overset{\text{def}}{=} T_\delta(A) \sqcup T_\delta(B) & T_\delta(\langle \sigma^\sharp \rangle\, A) \overset{\text{def}}{=} \exists \sigma^\sharp.T_\delta(A)
\end{array}$$

and similarly for the modal operator [.].

6 Conclusions

We have proposed a framework for aggregating information sources associated with agents where those information sources are inherently approximate due to the limited perceptual capabilities of the agents. These limitations are represented as similarity spaces. The dynamic logic framework is then used to aggregate information sources associated with the agents by aggregating their similarity spaces. This creates a global or fused situational context for the evaluation of formulas. These techniques have been demonstrated using a multi UAV scenario.

Acknowledgments

The research has been supported in part by an NFFP4 research grant and an SSF MOVIII strategic center research grant.

Barbara Dunin-Kęplicz would like to thank the Netherlands Institute of Advance Study in Wassenaar for the possibility to work on this subject as fellow-in-residence.

References

1. Dubois, D., Prade, H.: On the use of aggregation operations in information fusion processes. Fuzzy Sets and Systems 142, 143–161 (2004)
2. Doherty, P., Szałas, A.: On the correspondence between approximations and similarity. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) Rough Sets and Current Trends in Computing. LNCS (LNAI), vol. 3066, pp. 143–152. Springer, Heidelberg (2004)
3. Doherty, P., Łukaszewicz, W., Szałas, A.: Approximative query techniques for agents with heterogeneous ontologies and perceptive capabilities. In: Dubois, D., Welty, C., Williams, M.A. (eds.) Proceedings of 9th International Conference on the Principles of Knowledge Representation and Reasoning, KR'2004, pp. 459–468. AAAI Press, Stanford, California (2004)
4. Doherty, P., Łukaszewicz, W., Szałas, A.: Communication between agents with heterogeneous perceptual capabilities. Journal of Information Fusion 8, 56–69 (2007)
5. Doherty, P., Łukaszewicz, W., Skowron, A., Szałas, A.: Knowledge Engineering Techniques. In: A Rough Set Approach, Springer, Heidelberg (2006)
6. Demri, S., Orłowska, E.: Incomplete Information: Structure, Inference, Complexity. Springer, Heidelberg (2002)

7. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press, Cambridge (2000)
8. Rauszer, C.: Rough logic for multi-agent systems. In: Masuch, M., Polos, L. (eds.) Knowledge Representation and Reasoning Under Uncertainty. LNCS (LNAI), vol. 808, pp. 161–181. Springer, Heidelberg (1994)
9. Nardi, D., Brachman, R.J.: An introduction to description logics. pp. 5–44 [26]
10. Baader, F., Nutt, W.: Basic description logics. pp. 47–100 [26]
11. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
12. Doherty, P., Łukaszewicz, W., Szałas, A.: Tolerance spaces and approximative representational structures. In: Günter, A., Kruse, R., Neumann, B. (eds.) KI 2003: Advances in Artificial Intelligence. LNCS (LNAI), vol. 2821, pp. 475–489. Springer, Heidelberg (2003)
13. Doherty, P., Łukaszewicz, W., Szałas, A.: Approximate databases and query techniques for agents with heterogenous perceptual capabilities. In: Proceedings of the 7th Int. Conf. on Information Fusion, FUSION'2004, pp. 175–182 (2004)
14. Doherty, P., Łukaszewicz, W., Szałas, A.: Similarity, approximations and vagueness. In: Ślęzak, D., Wang, G., Szczuka, M., Düntsch, I., Yao, Y. (eds.) Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. LNCS (LNAI), vol. 3641, Springer, Heidelberg (2005)
15. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: A tutorial. pp. 3–98 [31]
16. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
17. Słowiński, R., Vanderpooten, D.: Similarity relation as a basis for rough approximations. In: Wang, P. (ed.) Advances in Machine Intelligence & Soft Computing, Raleigh NC, Bookwrights, pp. 17–33 (1997)
18. Słowiński, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Trans. on Data and Knowledge Engineering 12(2), 331–336 (2000)
19. Peleg, D.: Concurrent dynamic logic. In: STOC '85: Proceedings of the 7th Annual ACM Symposium on Theory of Computing, ACM Press, pp. 232–239. ACM Press, New York (1985)
20. Fisher, M., Ladner, R.: Propositional Dynamic Logic of regular programs. Journal of Computer and System Sciences 18, 194–211 (1979)
21. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, Reading (1996)
22. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning about Knowledge. MIT Press, Cambridge (1995)
23. Meyer, J.J., van der Hoek, W.: Epistemic Logic for AI and Theoretical Computer Science. Cambridge University Press, Cambridge (1995)
24. Dunin-Kęplicz, B., Verbrugge, R.: A tuning machine for cooperative problem solving. Fundamenta Informaticae 63, 283–307 (2004)
25. Dunin-Kęplicz, B., Verbrugge, R.: Collective intentions. Fundamenta Informaticae 51(3), 271–295 (2002)
26. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): Description Logic Handbook. Cambridge University Press, Cambridge (2002)
27. Borgida, A., Lenzerini, M., Rosati, R.: Description logics for databases. pp. 472–494 [27]
28. Schild, K.: A correspondence theory for terminological logics: preliminary report. In: Proceedings of the 12th IJCAI, pp. 466–471 (1991)
29. Baader, F.: Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. In: Proceedings of the 12th IJCAI, pp. 446–451 (1991)
30. Horrocks, I., Sattler, U.: A description logic with transitive and inverse roles and role hierarchies. Technical Report, LuFg Theoretical Computer Science, RWTH Aachen 98-05 (1998)
31. Pal, S., Skowron, A. (eds.): Rough Fuzzy Hybridization: A New Trend in Decision–Making. Springer, Heidelberg (1999)

Coevolution of a Fuzzy Rule Base for Classification Problems

Barbara Fusińska, Marek Kisiel-Dorohinicki, and Edward Nawarecki

Department of Computer Science
AGH University of Science and Technology, Kraków, Poland
{fusion,doroh,nawar}@agh.edu.pl

Abstract. In the paper a certain class of classification systems based on hybrid fuzzy-genetic approach is considered. The particular solution proposed allows for coevolution of two different populations, that search for a rule base structure and linguistic variables definitions respectively. The search is organised according to the principles of evolutionary multi-agent systems, which results in high generalisation capabilities of the system, as illustrated by the preliminary experimental results.

Keywords: data classification, fuzzy systems, evolutionary computation, multi-agents systems.

1 Introduction

Classification problems obviously arouse interest of many researchers. Among a variety of different approaches there is also a specific class of classification systems based on fuzzy and genetic techniques that operate in tandem. Such approach assumes the integration of a fuzzy rule base and inference system, which is responsible for the classification itself, and an evolutionary algorithm, which provides learning capabilities [2].

The aim of this paper is to present a specific technique of fuzzy classifiers generation by learning from examples using evolutionary processes. The novelty of the proposed approach firstly consists in using an evolutionary multi-agent system (EMAS) instead of classical evolutionary algorithms [8]. Also two coevolving populations, resulting from the proposed decomposition of the problem are introduced [21]. One population, which performs the search for the structure of a rule base, is organised according to the principles of EMAS allowing for high generalisation, so that even a smaller set of training data can assure correct assignment of unknown example. The definitions of linguistic variables used by the rules are coevolved by the second population as a classical evolutionary strategy [1]. At each step of the search, the best individuals from respective populations are used to evaluate another population, which makes the search a mutually adaptive process, and allows to get satisfactory solution of the whole problem (i.e. a complete fuzzy classifier).

The paper starts with a short introduction to a fuzzy rule-based classification systems (section 2). Then evolutionary processes are discussed as a technique

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 678–686, 2007.

of learning the FRBCS (section 3). Section 4 presents the system under consideration and the last section – preliminary experimental results, which illustrate how the proposed approach works for a few benchmark problems.

2 Fuzzy Classification Systems

In general **classification** consists in assigning certain membership classes to objects (events, phenomena), described by vectors of attributes [7]. In practice classification algorithms involve getting some data on input and putting the appropriate class on output, mostly assuming a given objects attributes set and a given class set. Of course before a classifier can give answers about the classes of presented objects, it should be trained using teaching data with correct membership classes provided. Efficient classification models characterize high generalization, which means they can assign a new (not provided in the learning phase) example to the right class. Anyway, building a proper knowledge base that can become a correct classifier strongly rely on the selection of training data with known assignments.

There are plenty of classification algorithms that use various classifier representations and different learning schemata. A classification mechanism can be constructed as a **rule-based system**, which approach dates back to early '60s, when Holland proposed the idea of a message passing, learning rule-based system, called simply a *classifier system* [5]. Rules represent the knowledge in a comprehensible form for those who will use the classification system, facilitating the use of this kind of systems as a tool in decision making processes. In addition, rules represent independent units of knowledge, so that alterations can easily take place in their contents. Classification rules can be based on fuzzy logic, which enables processing of imprecise or incomplete information, common in real classification problems. The systems that use fuzzy rules as knowledge representation are often called *Fuzzy Rule-Based Classification Systems* (FRBCS).

A **knowledge base** of FRBCS most often consists of [4]:

1. Fuzzy rules with *class as the conclusion* [3]. Conditions are defined by fuzzy sets and the conclusion is the class to be assigned to the object.
2. Fuzzy rules with *class and the membership degree as the conclusion* [6]. Conditions are defined by fuzzy sets and the conclusion is the class (with its membership degree) to be assigned to the object.
3. Fuzzy rules with *memberships degrees to all classes as the conclusion* [10]. Conditions are defined by fuzzy sets and the conclusion are membership degrees to all the classes.

A *rule base* of FRBCS is sometimes distingiushed from the so-called *data base*, which contains the fuzzy set definitions related to the fuzzy rules (fig. 1).

A classification process starts with reading an example i.e. object attribute values that may be represented as a tuple:

$$o = \langle a_1, ..., a_M \rangle \tag{1}$$

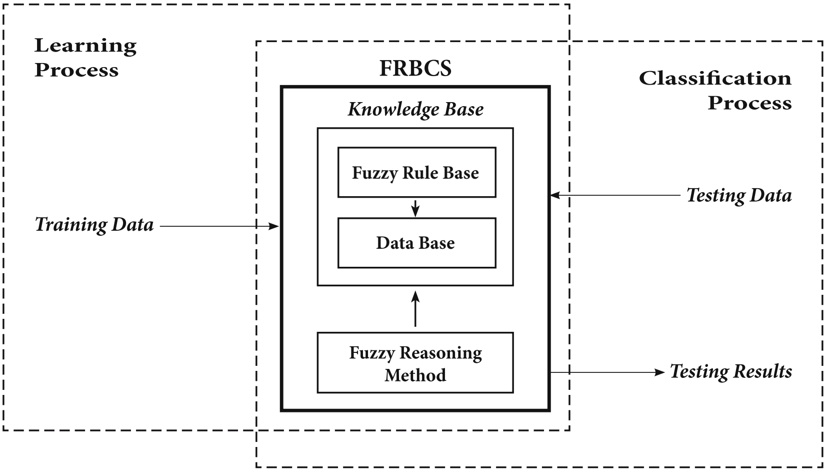


Fig. 1. The structure of FRBCS

where M is the number of attributes in the input data and a_k is the value of k-th attribute for the object o.

A *Fuzzy Reasoning Method* (FRM) is an inference procedure that derives a class to be assigned to object o by applying the rules from the knowledge base to the read data. In the first phase a *matching degree* is calculated for all the rules and the considered example. Then some rules are selected and used to draw the overall conclusion. Most often FRM considers only one rule (e.g. with the highest matching degree), which is the winning one and the response of the classifier is the class in this rule's conclusion.

3 Evolving the FRBCS

Despite apparent simplicity, constructing FRBCS for a given classification problem is not an easy task. In fact the construction of FRBCS is a supervised inductive process that fundamentally implies following tasks [4]:

1. *feature selection and extraction*,
2. *learning of fuzzy rules*, which means generation of a rule base,
3. *simplifying the rule base*,
4. *tuning the membership functions* that describe the semantics associated to the linguistic labels used by the linguistic variables.

In recent years a great number of publications have explored the use of evolutionary algorithms as a tool for designing fuzzy systems. Thus evolutionary computation can be used for completing each of the above tasks. Specific approaches differ in chromosome structure, fitness function shape, rule selection mechanism or the population behavior [4].

As far as evolutionary learning of a rule base is concerned (task 2) three approaches are often distinguished [11]:

- *Michigan approach* in which the chromosomes represent single rules and the whole population an rule base,
- in *Pittsburgh approach* each chromosome encodes a whole rule base,
- in *Iterative Rule Learning approach* each chromosome represents a single rule, but only the best individual (in iteration) is considered as the solution, discarding the remaining chromosomes in the population.

Evolutionary processes can also be applied to the data base searching for the definitions of fuzzy sets describing linguistic variables (task 4). Both these sets – rules and fuzzy set definitions – are relatively dependent, so that when one of them is changed it influences the another one. This observation leads to the discussed coevolutionary approach, which uses two populations realising tasks 2 and 4 simulatnously.

4 System Description

The discussed classification system is based on the general idea of evolving FRBCS, as described in two previous sections. Yet it introduces some essential modifications.

In the considered system rule's conditions are linguistic labels used to discretise the continuous domain of the variables, whereas conclusion is the class where the pattern belongs (1st type from the list in section 2) [3]. Each attribute value of the classified examples is assigned to one of linguistic labels (Low, Medium, High) that correspond to appropriate fuzzy sets defined separately for each attribute. Definitions of fuzzy sets can be represented by a set of parameters as shown in fig. 2 (a bigger number of sets can be represented analogically by increasing the number of parameters).

In the classification process a winning rule is assigned to the considered example according to the formula [9]:

$$arg \max_{i=1,\ldots,N} \{ \min_{j=1,\ldots,M} \{W_{ij}(a_j)\}\} \tag{2}$$

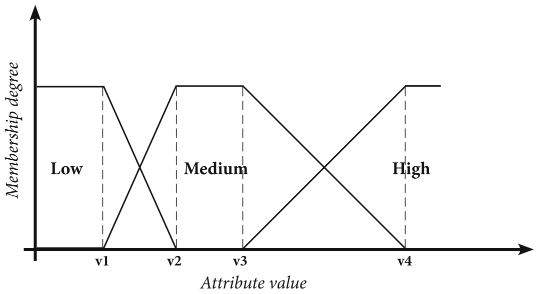


Fig. 2. Membership functions for linguistic labels in rule

where N is a number of rules in rule base, and $W_{ij}(a_j)$ denotes the membership degree of i-th fuzzy rule of the j-th attribute. The conclusion in the winning rule becomes the response of the whole classifier.

As it was already mentioned, evolutionary processes concern rules and fuzzy sets definitions. Pittsburgh approach was chosen as a fundamental model of the rule base structure evolution. Yet instead of classical evolutionary algorithms evolutionary multi-agent approach was used [8]. The **fitness** of an agent representing a whole rule base was calculated according to the following formula:

$$fitness = (TP/(TP + FN)) \cdot (TN/(TN + FP)) \tag{3}$$

where:

- TP (true positives) is the number of examples that are covered by at least one of the individual's rules and have the class indicated by those rules;
- FP (false positives) is the number of examples that are covered by at least one of the individual's rules but have a class different from the class indicated by those rules;
- FN (false negatives) is the number of examples that are not covered by any of the individual's rules but have the class indicated by those rules;
- TN (true negatives) is the number of examples that are not covered by any of the individual's rules and do not have the class indicated by those rules.

The main advantage of this approach to fitness evaluation is the fact that it considers not only number of proper and wrong examples assignment, but also linguistic variables definitions related to this assignment.

The **reproduction** of an individual agent depends on value of some probability factor. Mechanism of establishing reproduction partner in the case of positive decision is similar to the tournament selection. Individual agent is offered to cooperate with two partners. Finally there is one winner – the agent with better fitness. When these two agents are equally good there their complexity is taken into account. Recombination is done according to the one-point crossover schema where exchanged genetic material concerns whole rules. Mutation can be realised in two following ways:

- for conditions linguistic labels can be changed,
- for conclusions a class label can be changed.

Each continuous attribute is associated with its own set of membership functions, which define linguistic variables used by the rules of an agent. Simultaneous evolution of membership function definitions is realised according to a classical evolutionary strategy. The goal of this algorithm is to get better adaptation of a corresponding attribute to the structure of the rule bases used by the agents. The main advantage of this co-evolutionary approach is that the fitness of a given set of membership function definitions is evaluated across several rule sets, encoded into several different individuals, rather than on a single rule base.

5 Experimental Studies

The goal of the experiments was to evaluate the classification quality of the discussed system in comparison to other techniques. During the tests reported below there were used two data sets from the *UCI Machine Learning* repository, called *iris* and *glass types* respectively. Iris data set contains 150 examples, that may be divided to 3 linearly separable classes – 50 examples per class. All of the examples is described by 4 attributes. Glass types data set includes 214 examples, described by 9 attributes. Glass can be assigned to one of 7 (not linearly separable) classes.

5.1 Iris Set

Comparison tests were performed in two phases. The reason of this was different division ratio of the data between learning and testing sets in the compared algorithms.

Phase 1: Learning and testing data were divided in proportion 1:1. The results of the system were compared with the following methods: FCFSOM – a fuzzy classification system using self-organizing feature map [12], Nozaki method [13], Umano method [14], as presented in table 1.

Table 1. Quality comparison for iris data – phase 1

	Classifier quality [%]	
	learning set	testing set
FCSOM	99.23	94.83
Nozaki	-	93.03
Umano	-	94.43
Our system	98.07	93.48

Phase 2: In a learning set there were 70% of examples while in test set there were 30% of them. The results are presented in table 2 and were compared with following methods: C4.5 [15], CN2 [16], LVQ [17], and with another FRBCS systems: FRBCS [4], WM-FRLP [18], for which there were performed different configurations:

– classical FRM,
– FRM Normalised Sum - FRM NS,
– FRM Weighted Normalised Sum - FRM WNS,
– FRM Quasiarithmetic Mean - FRM QM.

The results presented show the high position of the system among other algorithms in both data set dividing proportions. Additionally there can be noticed the fact that quality of the system in test set is the best of the considered results.

Table 2. Quality comparison for iris data – phase 2

	Classifier quality [%]	
	learning set	testing set
C4.5	98.38	92.70
CN2	98.92	94.16
LVQ	98.55	95.72
FRBCS - FRM Classic	95.49	94.26
FRBCS - FRM NS	98.58	95.80
FRBCS - FRM WNS	97.47	94.36
FRBCS - FRM QM	95.30	94.23
WM-FRLP - FRM Classic	90.97	88.25
WM-FRLP - FRM NS	97.29	92.88
WM-FRLP - FRM WNS	98.56	94.38
WM-FRLP - FRM QM	91.18	90.34
Our system	98,20	99,20

5.2 Glass Types Set

Results collected for glass kinds set were compared with: LDA [19], SVM – for versions linear, quad and RBF [19], CART – for versions with full and best tree [19], neural nets [20], as presented in table 3.

Table 3. Quality comparison for glass kinds data

	Classifier quality [%]	
	learning set	testing set
LDA	73.74	83.33
SVN - linear	70.53	62.5
SVN - quad	73.68	75
SVN - RBF	86.84	37.5
CART - full tree	87.00	71.00
CART - best tree	81.00	67.00
Neural nets	80.95	75.00
Our system	94,12	81,90

Classification quality comparison shows that system tends to gain very good results, and proves a high position of the system among other algorithms. Based on these results it can be said that the system characterizes high adaptation to different kinds of problems together with high effectiveness, yet it must be stressed that high computational cost is paid for that.

6 Concluding Remarks

In the paper a co-evolutionary approach for discovering fuzzy classification rules was presented. The main advantage of the approach is that the fitness of a given set of membership function definitions is evaluated across several fuzzy rule sets, encoded into several different individuals, rather than on a single fuzzy set. A multi-agent environment allows for organisation of this complex process, so that the final result is a rule base and a set of membership function definitions which are well adapted to each other. It also makes the evaluation more robust.

The reported preliminary results show high classification quality for the considered problems, as compared to the results of several other approaches found in the literature which used the same data sets.

Obviously there are many directions for the future research considered, such as checking adaptation of the system with high dimensional problems or investigating the influence of increasing the number fuzzy sets per attribute.

References

1. Bäck, T.: Evolutionary algorithms in theory and practice. Oxford University Press, New York (1988)
2. González, A., Herrera, F.: Multi-stage genetic fuzzy systems based on the iterative rule learning approach, Mathware & Soft Computing, vol. 4(3) (1997)
3. González, A., Pérez, R.: Completeness and consistency conditions for learning fuzzy rules. Fuzzy Sets and Systems 96, 37–51 (1998)
4. Gordon, O., del Jesus, M., Herrera, F.: Evolutionary approaches to the learning of fuzzy rule-based classifications systems. CRC Press, Boca Raton (1999)
5. Holand, J.H.: Escaping brittleness: The possibilities of general-pupose learning algorithms applied to parallel rule-base system. In: Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (eds.) Machine Learning II, Morgan Kaufmann Publishers, San Francisco (1986)
6. Ishibuchi, H., Nozaki, K., Tanaka, H.: Distributed representation of fuzzy rules and its application to pattern classification, Fuzzy Sets and Systems, vol. 52 (1992)
7. Kasabov, N.K.: Foundation of Neutral Networks, Fuzzy Systems, and Knowledge Engineering. The MIT Press, Cambridge (1996)
8. Kisiel-Dorohinicki, M.: Agent-Oriented Model of Simulated Evolution. In: Grosky, W.I., Plášil, F. (eds.) SOFSEM 2002 LNCS, vol. 2540, Springer, Heidelberg (2002)
9. Mendes, R.R.F., de B., F., Voznika, A.A., Freitas, J.C.: Nievola, Discovering Fuzzy Classification Rules with Genetic Programming and Co-Evolution (PUC-PR, PPGIA-CCET), Curitiba - PR, Brazil (2001)
10. Punch, W.F, Goodman, E.D., Pei, M., Chia-Shun, L., Hovland, P., Enbody, R.: Further research on feature selection and classification using genetic algorithms. In: Proceedings of the Fifth International Conference on Genetic Algorithms (1993)
11. Freitas, A.A.: A survey of evolutionary algorithms for data mining and knowledge discovery, pp. 819–845. Springer, Heidelberg (2003)
12. Horikawa, S.: Fuzzy Classification System Using Self-Organizing Feature Map, Kansai General Laboratory, Research and Development Group
13. Nozaki, K., Ishibuchi, H., Tanaka, H.: Selecting fuzzy if-then rules with forgetting in fuzzy classification systems. Journal of Japan Society for Fuzzy Theory and Systems, vol. 6(3) (1994)
14. Umano, F., Hatono, T.: Extraction of fuzzy rules using fuzzy neural networks with forgetting. Transaction of Society of Instrument and Control Engineers, vol. 32(3)(1996)

15. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Francisco (1993)
16. Clark, P., Niblett, T.: Learning If Then Rules in noisy domains, TIRM86-019, The Turing Institute, Glasgow (1986)
17. Kohonen, T.: Self-Organizing Maps. Springer-Verlag Series in Information Sciences. Springer, Heidelberg (1995)
18. Chi, Z., Wu, J., Yan, H.: Handwritten numeral recognition using self-organizing maps and fuzzy rules. Pattern Recognition, vol. 28(1) (1995)
19. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning - Data Mining, Inference and Prediction. Springer, Heidelberg (2001)
20. Weiss, S.M., Kapouleas, I.: An empirical comparison of pattern recognition, neural nets and machine learning classification methods. Morgan Kauffman, San Francisco (1990)
21. Potter, M., de Jong, K.: A cooperative coevolutinary approach to function optimization. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) Parallel Problem Solving from Nature - PPSN III. LNCS, vol. 866, pp. 249–257. Springer, Heidelberg (1994)

Towards Agent-Based Evolutionary Planning in Transportation Systems*

Jarosław Koźlak, Marek Kisiel-Dorohinicki, and Edward Nawarecki

Institute of Computer Science
AGH University of Science and Technology, Kraków, Poland
{kozlak,doroh,nawar}@agh.edu.pl

Abstract. In this paper problems of planning in transportation systems based on Pickup and Delivery Problem with Time Windows (PDPTW) are discussed. The results of two variants of evolutionary algorithms illustrate the pros and cons of using different approaches, and their cooperation in hybrid island model indicates how they can help each other in achieving better solutions. This leads to the general idea of an agent-based cooperative system, in which many different techniques may be used simultaneously, exchanging the obtained solutions. Experimental study of such a system that uses evolutionary algorithms and tabu search concludes the work.

Keywords: transport planning and scheduling, Pickup and Delivery Problem with Time Windows, evolutionary algorithms, multi-agent systems.

1 Introduction

It is rather obvious that effective organisation of transportation systems allows companies to highly limit sustained costs and be more competitive on the market. This could hardly be achieved without adequate tools, which should support transport planning on the basis of acquired knowledge on available resources, incoming requests and road network structure. Critical situations analysis seems to be of vast importance for such planning. Yet, even though there is a wide selection of planning techniques, most of them assume a complete description of both resources and requests available a priori. Thus it is very difficult (or even impossible) to apply them to dynamic problems, and even more difficult, with unsure and incomplete knowledge.

The goal of the research partially reported in this paper is to create concepts and tools, that should manage planning in dynamic environments of multi-agent systems in the face of crisis, considering transportation systems as a particular case. Based on a general scheme of crises management in MAS, as well as preliminary results obtained in the field of transportation systems [7], several possible variants of planning-support techniques were already considered [2]. In this paper special attention is paid to evolutionary techniques as a tool for solving static transportation problems, moving to cooperative systems, which should be flexible enough to be used in dynamic environments.

* This work was partially sponsored by State Committee for Scientific Research (KBN) grant no. 3 T11C 025 27.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 687–696, 2007.

Still the paper does not touch dynamic problems, but rather shows how different techniques or different configurations of similar techniques can help one another, attaining better results than when used alone.

Section 2 introduces a particular transportation problem considered in the paper, that is Pickup and Delivery Problem with Time Windows (PDPTW). In section 3 there is a discussion of evolutionary algorithms dedicated to solving transportation problems, with special attention to the advantages and shortcomings of different approaches. Selected experimental results that illustrate the described algorithms are presented in the next section. Section 5 introduces the idea of a cooperative system that allows for exchanging of solutions between different algorithms solving transportation problems, and finally section 6 provides an experimental study of the system at work.

2 Research on Transportation Problems

Typical transportation problems are based on a set of requests being realised by a set of available vehicles. Vehicles are characterised by their capacity and speed, and requests by the required capacity of vehicles and a time period (known as *time window*) within which the pickup and delivery operations have to take place. In a more widely researched *Vehicle Routing Problem with Time Windows* (VRPTW) with each transport request only one location point (either pickup or delivery) is associated, but in *Pickup and Delivery Problem with Time Windows* (PDPTW) each request is characterised by both a pickup and delivery location. The quality of a solution depends on the number of vehicles used and the total distance travelled. Sometimes, to express the quality of a solution, a total travel time of vehicles and a total waiting time of vehicles before the start of any time window is also considered. In problems *with hard time windows*, it is absolutely necessary that pickup and delivery operations start in the given time window. In problems *with soft time windows*, pickup and delivery operations may start after the end of this time period, but in estimating the quality of a solution, a penalty for the delay may be taken into account (higher delay may result in higher penalties). The problems have numerous practical applications — for example in planning sea and air transport, different kinds of cargo services and transport services on demand (for example transport of handicapped for treatment) or taxi-share services.

A set of benchmark tests for VRPTW was proposed by Solomon and extended by Gehring and Homberger. Li and Lim proposed a similar set of benchmark problems to verify the quality of the algorithms for PDPTW [9]. Benchmarks are divided into different groups depending on the number of requests to be served and locations to be visited (about 100, 200, 400 locations etc.). For each group six classes of tests are distinguished: on one hand due to the characteristics of time windows (problems with small time windows and a short scheduling horizon — LR1, LC1, LRC1, as well as with large time windows and a long planning horizon — LC2, LR2, LRC2), on the other due to the spatial distribution of requests (request locations may be grouped into clusters — LC1, LC2, evenly distributed — LR1, LR2, and there are also mixed problems with some request locations in clusters and some randomly distributed — LRC1, LRC2).

Due to the complexity of the described transportation problems (mainly on account of many constraints) nowadays the most promising approximate solutions provide heuristic approaches (accurate solutions are not attainable because of NP-hardness of the problem). The majority of algorithms are based on the generation of an initial solution using some simple heuristics (like insertion heuristic, sweep heuristic or partition heuristic), which is optimised afterwards using some metaheuristics. The proposed algorithms for VRPTW are numerous and it would be difficult to list them here. However it is worth mentioning that when comparing different VRPTW solving algorithms [1], hybrid evolutionary approaches achieve the best results. The approach based on the tabu search and simulated annealing [5] provided the best solutions obtained so far for PDPTW. Many other interesting approaches to PDPTW are also based on tabu search, e.g. [6,4].

3 Evolutionary Approach to Transportation Problems

Evolutionary algorithms are based on iterative transformation of the *population of individuals* potential solutions of the given problem. Evolution consists on generating consecutive generations, using so called *genetic operators* (or *variation operators*) and the *selection* mechanism.

Most evolutionary algorithms for transportation problems use direct representation of solutions [1] – each individual consists of consecutive locations assigned to particular routes. Such representation assumes no *coding*, which results in genetic operators operating directly on solutions. This guarantees the generation of acceptable solutions, which is easily achieved introducing genetic operators based on existing optimisation algorithms dedicated to transportation problems (e.g. pointed out in the previous section). Also the initial population is not generated randomly, as for typical evolutionary algorithms, but by using some existing construction heuristics. Several criteria considered for transportation problems are often aggregated (e.g. as a weighted sum) into a single value, which may be used as the fitness of individuals.

The discussed approach [3] is based on GENEROUS algorithm, which uses direct representation as described above. Two recombination operators: based on sequence (SBX) and route exchange (RBX), allow an improvement of the total distance, yet can hardly reduce the number of vehicles. Thus two mutation operators: one level (1M) and two level (2M) exchange, aim at emptying (the shortest) routes. Third mutation operator works as a local optimiser based on *or-opt* technique. If the solution cannot be repaired (there are unserved locations), it is rejected and the whole process is repeated [8].

This algorithm was adapted for the PDPTW problem, leaving the same representation, as well as slightly modified SBX recombination and 1M mutation operators. Also, additional recombination operator for exchanging best routes and mutation operators based on the concept of ghost routes were introduced. The initial population is generated using a clustering technique in the first phase and a modified sweep heuristic [5] to fill in the population up to the assumed size in the second phase. Tournament selection was used with individual comparison based on three criteria: the number of routes, a total distance, and a total waiting time, considered one by one in the given order.

In general the process of evolution should tend to generate better individuals and finally to find the needed (usually approximate) problem solution, which quality depends on the operators used and the parameters of the algorithm. Yet, evolutionary computation often suffers from the loss of population diversity, which practically hinders further search. This means that the algorithm locates the basin of attraction of some local optimum instead of a global one. This is especially important considering transportation problems with direct representation, because of introduced constraints, which often eliminate many new individuals from the population.

That is why a second considered variant of evolutionary algorithm utilized a partial representation of the solution, which consisted of only pickup locations. For such representation genetic operators for travelling salesman problem might be used. Also the initial population could be generated randomly. An *insertion heuristics* [5] allowed for transformation of every individual into a feasible complete solution. It was chosen because of its low computational complexity (it must be used for every individual in every generation), yet unfortunately permitted different individuals to be transformed into the same, often weak solution. Selection and fitness evaluation was realised in the same way as for the previous algorithm.

4 Experimental Comparative Study of the Evolutionary Approaches

Various experimental studies were conducted in order to compare the performance of the above-described algorithms [3]. Below, only selected results allowing for drawing preliminary conclusions are presented. The results were obtained for 100-location problems with even distribution of request locations, with small (LR1) and large (LR2) time windows. Tables 1 and 2 show the benchmark results [9] and the best individual obtained averaged over 3 independent runs of each algorithm, with the population of 125 individuals evolving for 125 generations.

Table 1. Results obtained for problems with small time windows (LR1)

	benchmark		GEN1		GEN2		GEN1+GEN2	
problem	routes	distance	routes	distance	routes	distance	routes	distance
lr101	19	1650.8	19	1744.5	19	1650.8	19	1650.8
lr102	17	1487.6	17	1580.9	17	1575.1	17	1523.9
lr103	13	1292.7	14	1550.1	13	1421.2	13	1369.8
lr104	9	1013.4	10.7	1149.9	11	1244.7	9.5	1037.2
TOTAL	58	5444.4	60.7	6025.4	60	5891.7	58.5	5581.8

For problems with small time windows (table 1) the results were quite good – both algorithms could find solutions very close to the benchmark ones (but one must remember that one vehicle more in the obtained solution gives a considerable relative difference to the benchmark value). The situation was slightly different for large time windows (table 2) – even though the number of vehicles obtained by both algorithms was still comparable to the benchmark result, the total distance was worse for the algorithm with direct representation (GEN1), and much worse for the algorithm with partial representation (GEN2).

Table 2. Results obtained for problems with large time windows (LR2)

	benchmark		GEN1		GEN2		GEN1+GEN2	
problem	routes	distance	routes	distance	routes	distance	routes	distance
lr201	4	1253.2	4	1419.1	4	1923.5	4	1328.4
lr202	3	1197.7	4	1398.9	4	1734.9	4	1341.2
lr203	3	949.4	3	1224.4	3	1849.6	3	1115.3
lr204	2	849.1	3	1099.7	3	1494.1	3	1110.9
TOTAL	12	4249.4	14	5142.1	14	7002.1	14	4895.9

The reasons for the weak results obtained seem to be different for the algorithms discussed. As already suggested and illustrated by figures 1, the first algorithm suffered from the lack of diversity in the evolving population, which inhibited its search capabilities from ca. 40-50 generation. One may notice that the second algorithm maintained the diversity for the whole run. The reason for weak results in this case was the heuristics used to generate complete solutions, as suggested in the previous section.

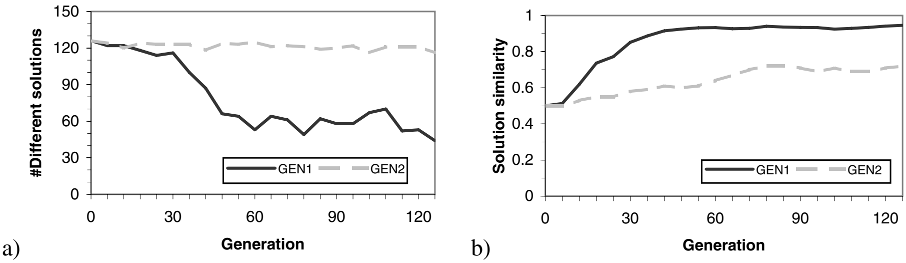


Fig. 1. The number of different solutions (a) and the similarity of solutions (b) in algorithms GEN1 and GEN2

The obvious conclusion drawn from these experiments was to use both algorithms simultaneously, allowing to exchange the solutions during the search. This meets the idea of the hybrid island model of parallel evolutionary algorithm, assuming that migration operator is responsible for conversion of the solutions between representations used by both algorithms. The results presented in the third part of tables 1 and 2 are quite promising and initially confirm the correctness of the approach.

5 Solution at a Cooperative Level

As it was illustrated in the previous section different algorithms applied to transportation problems have different strengths and weaknesses, e.g. some may be better suited to solving problems with small time windows and other for problems with large time windows. This is also confirmed by benchmark results – the best known solutions for a given test case are often obtained by different algorithms [9]. Preliminary results obtained for the discussed dual-population evolutionary algorithm indicated that cooperation of different approaches should allow to achieve more flexibility and produce better results for a variety of test cases.

That is why the environment was developed that facilitates the cooperation of different algorithms solving PDPTW (with both hard and soft time windows), by means of exchanging solutions or even parts of solutions (routes, requests served in the routes). The system model and architecture is based on a multi-agent approach and consist of several embedded sub-environments which contain computational agents [3]. There are two kinds of agent groups:

- standard groups – applying the algorithms proposed by [5] based on tabu search and simulated annealing,
- evolutionary groups - take advantage of algorithms presented above.

Of course the optimisation is performed simultaneously by different agents using different algorithms.

The quality function used has the following form:

$$f = \alpha N + \beta D + \gamma CD + \delta WT + \varepsilon P \tag{1}$$

where: NV – the number of vehicles, TD – total distance, SD – total service realisation time, WT – total waiting time, P – total lateness, α – weighing factor of the number of vehicles (in tests equals 5000000), β – weighing factor of the total distance (in tests equals 1000), γ – weighing factor of total service realisation time (in tests equals 1), δ – weighing factor of total waiting time (in tests equals 0.001), ε –weighing factor of penalty caused by lateness (in tests equals 100). The quality of the solution decreases with the increase of the value of the f function.

The important characteristic feature of the presented approach is a cooperative aspect of the computation process. Agents which represent different algorithms find the routes and requests having the worst influence on the quality of solution (have the highest impact on the quality function). Each agent is then informed by other agents about similar routes to their worst ones (identified by the central point calculated as the average of respective coordinates of request/delivery locations present in the given route), and about the routes, where the other agents placed the most costly requests that were analysed. On the basis of the obtained suggestions, the agent may modify its route or even construct a new one, selecting the request from the route being replaced or from other accessible routes and moving the other requests from the route being removed to other feasible positions in other routes.

6 Results of the Cooperative Approach

The goal of the tests performed was to compare the quality of solutions offered by the discussed cooperative algorithm with the quality offered by the considered metaheuristics used alone, as well as the quality of the best known solutions. Numerous tests were performed for different sets of Li-Lim benchmarks, but as the space in this paper is somewhat limited, only the most interesting results concerning the number of vehicles and total travel distance are presented.

To search a wide part of the solution space, different quality functions were applied in the particular agents. These differences are consequences of different weights (sometimes randomly generated) of particular elements of the quality function. Thanks to this approach, it was possible to take into consideration different kinds of solutions, for example the ones that attached greater significance to the number of vehicles used, a total distance or the arrival on time at service points. The difference between the results may also be influenced by the fact that the cooperative approach used soft time windows and thus allowed solutions with vehicles arriving late at service points, but their lateness was penalised by an important factor.

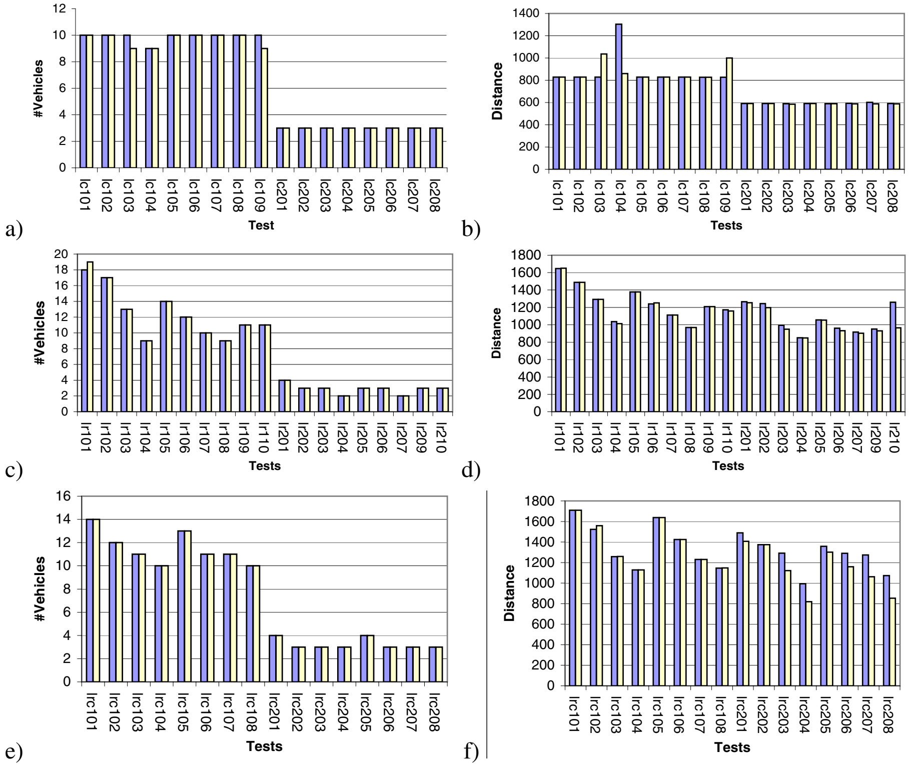


Fig. 2. Results for 100 locations: vehicles (a) and total distance (b) for LC1/LC2, vehicles (c) and total distance (d) for LR1/LR2 and vehicles (e) and total distance (f) for LRC1/LCR2; dark bars – results of our cooperative algorithm, fair bars – the best known solutions

The computational environment was composed of two groups of three agents of different types (tabu and evolutionary). If the basic algorithms were able to find the best known solution, the meta-algorithms were unable to find a better one, unless it accepted some lateness and penalty factor associated with it. In the situations when basic algorithms were not able to find the best known solutions, the meta-algorithm sometimes guaranteed an increase of solution quality. The best benefits of the introduced approach appeared in the problems with small time windows in the LR type problems. One can also notice that the worst results were obtained for LRC problems and for long time windows.

Figures 2 and 3 show the results obtained by the cooperative algorithm in comparison to the best known solutions for benchmark problems with 100 and 200 request locations. The figures include the results for cases with clusters, with even spatial distribution and mixed clusters/distributed. For each figure, the results for small and large time windows are presented. The number of used vehicles and a total travel distance for each group of tests are presented in separate figures.

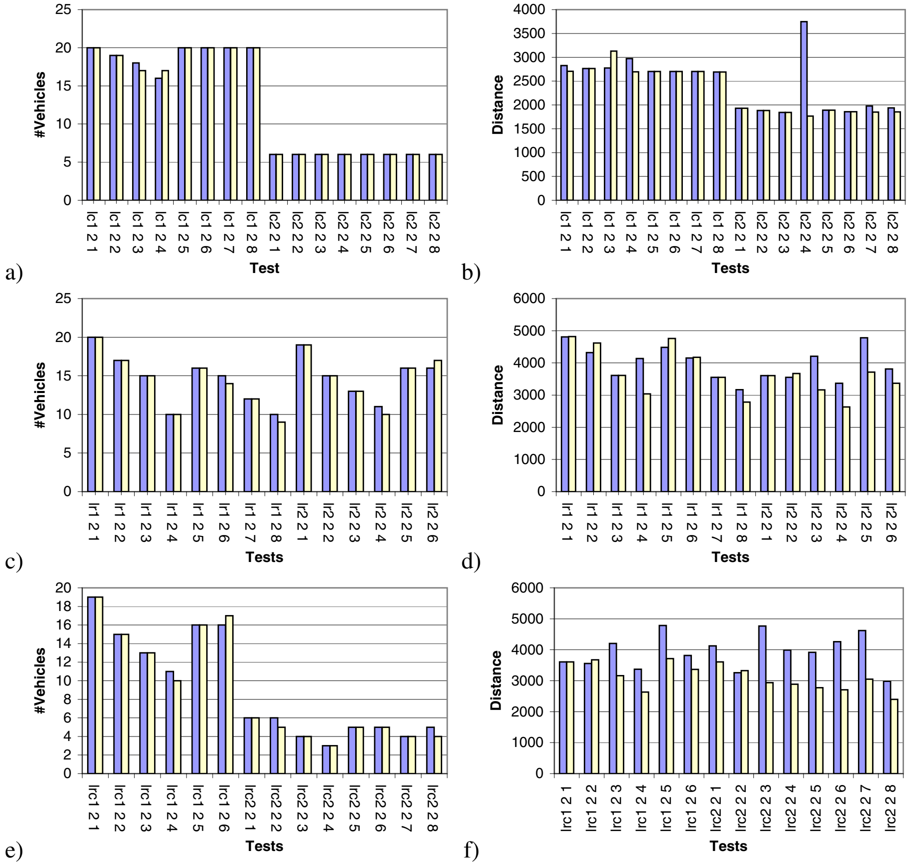


Fig. 3. Results for 200 locations: vehicles (a) and total distance (b) for LC1/LC2, vehicles (c) and total distance (d) for LR1/LR2 and vehicles (e) and total distance (f) for LRC1/LCR2; dark bars – results of the cooperative algorithm, fair bars – the best known solutions

In table 3 the results obtained for cooperative algorithm are compared with the results obtained using only evolutionary island model based on algorithms presented in the previous section and the multi-agent systems which uses only tabu-search heuristics. In the columns concerning the distances ”+” (”-”) means that cooperative algorithm obtained better (worse) results with respect the total travel distance in the given percentage of tests of the considered test class. In the columns concerning the number of routes additional information is provided. It concerns a difference between the number of routes used by the cooperative solution and basic solutions (”+” – less routes, ”-” – more routes).

Note that in the case of island model of evolutionary algorithm and multi-agent system consisting of tabu agents sometimes not all benchmark problems in the given class were solved. The table shows that mixing the solutions obtained from evolutionary and tabu algorithms using the cooperative algorithm in general gives better results.

Table 3. Results obtained for cooperative approach in comparison to evolutionary algorithm and agent-based tabu search

	cooperative/evolutionary		cooperative/agent-based tabu	
problem	routes	distance	routes	distance
100 LC1	(25%,-1)	(25%+),(25%-)		(11%+)
100 LC2		(50%+)		(25%+),(25%-)
100 LR1	(25%,+1)	(75%+),(25%-)	(10%,+1)	(60%+)
100 LR2	(25%,+1)	(100%+)		
100 LRC1	(25%,+1)	(100%+)	(12.5%,+1)	(75%+)
100 LRC2	(25%,+1)	(75%+), (25%-)		(37.5%+)
200 LC1	(25%,+1)	(75%+), (25%-)	(12.5,%+1)	(50%+), (12.5%-)
200 LC2		(50%+), (25%-)	(50%+)	(37.5%+), (12.5%-)
200 LR1	(25%,+2),(25%,+1)	(75%+)	(50%,+2)	(75%+), (25%-)
200 LR2	(25%,+1)	(50%+), (50%-)	—	—
200 LRC1	(25%,+2),(25%,+1)	(50%+),(50%-)	—	—
200 LRC2	(75%, +1)	(25%+),(75%-)	—	—

The final total results are as follows:

- 33% was equal to the best known solutions,
- 22% was better than the best known solution, after the application of soft-time windows and calculation of penalties,
- 14% was worse than the best know solutions obtained so far, considering the number of vehicles,
- 36% of results were worse then the best known solutions considering the total travel distance.

7 Concluding Remarks

In this paper two different approaches of growing complexity for solving transportation problems were presented. The results obtained using both systems do not differ significantly from the best known solutions for the existing set of benchmark problems. The cooperative approach not only allows to get slightly better results but also proves much more flexible. This is due to the use of soft time windows, because often it is not possible to strictly predict times of the particular activities (like travel time, pickup time, delivery time) or the definition of changes in the problem (due to breaking down of the cars or withdrawing of requests). In relation to this, considering the possibility of development of plans based on different definitions of quality function may make it possible to find solutions which are more resistant to critical situations. It also makes it possible to develop a set of plans which afterwards may be adapted to the current conditions with respect to new or unpredicted events arising. Additionally, it may constitute a basis for the development of solutions for dynamic problems (when new requests arrive simultaneously while the vehicles are serving the previously accepted requests), which will be the subject of further research.

References

1. Braysy, O.: Genetic algorithms for the vehicle routing problem with time windows. Arpakannus. Special issue on Bioinformatics and Genetic Algorithms 1, 33–38 (2001)
2. Dreżewski, R., Kisiel-Dorohinicki, M., Koźlak, J.: Agent-based and evolutionary planning techniques supporting crises management in transportation systems. In: Dolgui, A., Morel, G., Pereira, C.E. (eds.) Information Control Problems in Manufacturing 2006. Information Systems, Control and Interoperability, vol. 1, Elsevier, Amsterdam (2006)
3. Lalewicz, M., Wójcik, J., Rola, L., Srebro, M., Koźlak, J., Kisiel-Dorohinicki, M.: Evolutionary and hybrid approaches for PDPTW (in polish: Ewolucyjne i hybrydowe podejścia do PDPTW). Technical Report 5/2005, Department of Computer Science, AGH-UST, Kraków (2005)
4. Lau, H., Liang, Z.: Pickup and delivery with time windows : Algorithms and test case generation. In: Proceeedings of 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'01), Dallas, USA (May 2001)
5. Li, H., Lim, A.: A metaheuristic for the pickup and delivery problem with time windows. In: Proceedings of 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'01), Dallas, USA (2001)
6. Nanry, W., Barnes, J.: Solving the pickup and delivery problem with time windows using reactive tabu search. Transportation Research 34(Part B), 107–121 (2000)
7. Nawarecki, E., Koźlak, J., Dobrowolski, G., Kisiel-Dorohinicki, M.: Discovery of crises via agent-based simulation of a transportation system. In: Pěchouček, M., Petta, P., Varga, L.Z. (eds.) Multi-Agent Systems and Applications IV. LNCS (LNAI), vol. 3690, pp. 132–141. Springer, Heidelberg (2005)
8. Potvin, J.Y., Bengio, S.: The vehicle routing problem with time windows - part ii: Genetic search. INFORMS Journal on Computing 8, 165–172 (1996)
9. Benchmarks - vehicle routing and travelling salesperson problems. `http://www.sintef.no/static/am/opti/projects/top/`

Exploiting Rough Argumentation in an Online Dispute Resolution Mediator*

Ioan Alfred Letia and Adrian Groza

Technical University of Cluj-Napoca
Department of Computer Science
Baritiu 28, RO-400391 Cluj-Napoca, Romania
{letia,adrian}@cs-gw.utcluj.ro

Abstract. Online dispute resolution is becoming the main method when dealing with a conflict in e-commerce. Our framework exploits the argumentation semantics of defeasible logic, shown to be a suitable choice for legal reasoning. We introduce the rough set theory within defeasible logic for handling the gradual information revealed in a legal dispute. The rough sets are being used in the generation of defeasible theories from available cases, but also in the inference rules required for the argumentation process. The framework can cover both aspects of the law: case based reasoning and legal syllogism.

1 Introduction

Online Dispute Resolution (ODR) promises to become the predominant approach to settle e-commerce disputes, after ten years of fast and sustained development [1]. Our goal is to provide a flexible decision support framework, according to the current practice in law, with potential benefits to the e-commerce and legal communities [2,3].

Flexibility in configuring ODR systems is both an opportunity and a challenge. The opportunity is that any business can, quite quickly, have its own "court" specialized in disputes that might occur in its specific business domain. The challenge is that the technical instrumentation must simultaneously satisfy the business viewpoint asking for trust [4] and the legal viewpoint, which requires accordance with the current practice in law. One aspect covers how to combine different knowledge sources such as the legal framework, contractual clauses, and data representing precedent litigation cases for assisting the resolution process.

We base our framework on rough set theory and defeasible logic, enriched with level of certainty to handle practical applications. After the presentation of the formalism by defining both sustaining and defeating rules for a claim, we show how available cases can be used to generate defeasible theories. Considering judicial decisions, four components are particularly relevant: the inference method, the minimul level of certainty required to accept evidence, the selection of relevant arguments, and the derivation of the outcome.

* Part of this work was supported by the grant 27702-990 from the National Research Council of the Romanian Ministry for Education and Research.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 697–706, 2007.

2 Argumentation Framework

Our framework exploits the argumentation semantics of defeasible logic which is proved to be the most suitable choice for legal reasoning [5]. A theory in defeasible logic is a structure $\langle F,\ R\rangle$ formed by a finite set of facts $f(\beta) \in F$, and a finite set of rules $r(\gamma, cov) \in R$, having the *certainty factors* β, γ and the *coverage factor* $cov \in [0..1]$. A fact $f(\beta) \in F$ is *strict* if $\beta = 1$ and *defeasible* if $\beta < 1$. The rules are split in two disjoint sets: the set of support rules R_{sup} which can be used to infer conclusions and the set of defeaters R_{def} that can be use only to block the derivation of conclusions. A rule $r(\gamma, cov) \in R_{sup}$ is *strict* if and only if $\gamma = 1$. The set of strict rules is represented as $R_s = \{r(\gamma) \in R_{sup} | \gamma = 1\}$. A rule $r(\gamma) \in R_{sup}$ is *defeasible* if and only if $\gamma < 1$. The set of defeasible rules is represented as $R_d = \{r(\gamma) \in R_{sup} | \gamma < 1\}$.

Strict rules are rules in the classical sense, that is whenever the premises are indisputable, then so is the conclusion, while defeasible rules are rules that can be defeated by contrary evidence. Defeaters are rules that cannot be used to draw conclusions, they are use for preventing conclusions. A defeasible conclusion q can be defeated either by inferring the opposite $\sim q$ with a superior certainty factor (rebuttal defeater), or by attacking the link between the premises and the conclusion q (undercutting defeater[1]). The problem arises since a defeater of the consequent q attacks all rules which sustain q and there is no mean to attack a single rule sustaining the respective conclusion. To handle this we introduce negated rules, with the following notation: $\nrightarrow$ for $\neg\ (a \rightarrow b)$, meaning that "a does not strictly determine b", $\nRightarrow$ for $\neg\ (a \Rightarrow b)$, expressing that "a does not defeasibly determine b", and $\not\rightsquigarrow$ for $\neg\ (a \rightsquigarrow b)$, meaning that "$a$ does not defeat b". R_{ns} denotes the set of negated strict rules, R_{nd} the set of negated defeasible rules, and R_{ndef} the set of negated defeaters.

Table 1. Attacking a sentence φ depends on its type

φ	$\sim\varphi$
q	$\neg q,\ X \rightarrow \neg q,\ X \Rightarrow \neg q$
$A \rightarrow q$	$\neg q,\ X \rightarrow \neg q,\ A \nrightarrow q$
$A \Rightarrow q$	$\neg q,\ X \rightarrow \neg q,\ X \Rightarrow \neg q,\ X \rightsquigarrow \neg q,\ A \nRightarrow q$
$A \rightsquigarrow q$	$A \not\rightsquigarrow q$
$A \nrightarrow q$	$A \rightarrow q$
$A \nRightarrow q$	$A \Rightarrow q$
$A \not\rightsquigarrow q$	$A \rightsquigarrow q$

The type of counterargument depends on the type of the current sentence φ: fact, support rule, defeater (table 1), where A represents the set of antecedents sustaining q and X represents a different set of premises supporting the opposite

[1] Intuitively, an undercutting defeater argues that the conclusion is not sufficiently supported by its premises.

conclusion. *"A defeasible implies q" $(A \Rightarrow q)$* can be attacked either by i) simply claiming the opposite fact $\neg q$, ii) deriving (strictly or defeasibly) the opposite conclusion based on a different set of premises X $(X \rightarrow \neg q,\ X \Rightarrow \neg q)$, iii) blocking the derivation of the consequent $(X \rightsquigarrow \neg q)$, or iv) claiming the set A does not suffice to sustain the consequent q $(A \not\Rightarrow q)$.

3 From Data to Defeasible Theory

Consider a dispute scenario where the initial information provided by the plaintiff reveals that the item was delivered in time, but supplementary charge has occurred. As no clause has been signed regarding another remedy in case of a dispute issue, the customer considers he is right to claim the money back.

Collecting Similar Cases. The first phase consists of collecting all the cases with the similar attributes and the respective remedy in each case. The decision table we consider consists of the 98 cases for dispute resolution (table 2). The condition attributes regards the payment arrangements, the delivery status, and the existence or not of a contractual clause stipulating that no refund will be considered, but only the item replacement. The column labeled "remedy" contains the output of the resolutions, while N represents the number of similar cases, considering the given attributes.

Table 2. Dispute resolution cases for the *refund* claim

#	payment	contractual_clause	delivery	remedy	N
1	supplementary charge	yes	partial	refund	8
2	incorrect price	yes	on time	refund	10
3	supplementary charge	no	on time	refund	4
4	initial price	yes	partial	no refund	50
5	incorrect price	no	delayed	refund	6
6	supplementary charge	no	on time	no refund	20

The current level of abstraction contains inconsistent data: facts 3 and 6 where the same premises imply opposite decisions. These cases contain hidden reasons which have influenced the outcome and which might be revealed in the case of a low level dispute analysis. At this phase, the attributes *payment*, *contractual_clause*, and *delivery* describe the set of refund decisions approximately[2]. Under the rough set theory, the lower approximation $B_a = \{1, 2, 5\}$ represents the maximal set of facts that can certainly be classified as with the

[2] The report *The European Online Marketplace: Consumer Complaints 2005* conveyed that 46% of the complaints regard delivery issues, 8% payment arrangements, and 8% quality of the items.

refund outcome. The upper approximation $B^a = \{1, 2, 5, 3, 6\}$ contains the possible cases where the refund decision might be taken. The difference $B^a \setminus B_a = B = \{3, 6\}$ is called boundary region of the set $\{1, 2, 3, 5\}$ in which the refund decision was taken. A set is rough if its boundary region is not empty [6].

Computing the Reducts. The next step in data analysis is to find a minimal subset of data that preserves the degree of consistency. This reduct represent the essential data used to derive the corresponding defeasible theory. Following [6], table 3 presents one such reduct.

Table 3. Dispute resolution cases with consistency

#	**payment**	**contractual_clause**	**remedy**	**N**
1'	supplementary charge	yes	refund	8
2'	incorrect price	-	refund	10
3'	supplementary charge	no	refund	4
4'	initial price	-	no refund	50
5'	incorrect price	-	refund	6
6'	supplementary charge	no	no refund	20

Every fact in the data table determines a decision rule. For instance the fact 1' is synonym with the rule r'_1 : $supplementary\_charge \wedge clause \Rightarrow refund$. In order to provide explanation of decisions in terms of conditions one can define inverse decision rules [6]: r''_1 : $refund \Rightarrow supplementary\_charge \wedge no\_refund\_clause$. Explanation capabilities are necessary [4] in such a dispute resolution system to provide trustworthiness in the outcome.

Generating Defeasible Rules. We use the conditional probabilities in [6] to derive defeasible rules. The *support* of a rule $r : p \Rightarrow q$ represents the number of cases in the decision system that pose both properties p and q, where p is the conjunction of premises. The certainty factor represents the accuracy of the rule $p \Rightarrow q$, defined as $\gamma(r) = \gamma(q|p) = support(p, q)/support(p)$. The coverage factor reflects how well a specific case is replicable $cov(r) = cov(p|q) = support(p, q)/support(q)$. The certainty factor helps to introduce defeasible logic. A rule with $\gamma = 1$ defines a strict rule, while a rule with $\gamma < 1$ a defeasible rule. Strict rules act in the lower approximation region, while defeasible rules in the boundary region. When new information is available, it guides the process into a refined stage. The new facts are modeled with defeaters (rebuttal, undercutting or negated rules), which may block the derivation of some defeasible conclusions. In the rough set interpretation, the initial boundary region will be adjusted according to the new information, for which the defeater semantics helps to adapt the model built so far. Thus, an incorrect classification in the boundary region can be challenged by an undercutting defeater which attacks the link between the premises and the conclusion. From table 3 the pairs $\langle \gamma, cov \rangle$ are computed for each rule[3], with the defeasible theory generated as illustrated in figure 1. The frequency of the rules generated from all the computed reducts can be used to define an importance measure of the rule [7].

$$\begin{array}{l}
r'_1\langle 1, 0.29\rangle : \mathit{supplementary\_charge}, \mathit{no\_refund\_clause} \rightarrow \mathit{refund}\\
r'_2\langle 1, 0.57\rangle : \mathit{incorrect\_price} \rightarrow \mathit{refund}\\
r'_3\langle 0.27, 0.14\rangle : \mathit{supplementary\_charge}, \mathit{no\_clause} \Rightarrow \neg \mathit{refund}\\
r'_4\langle 1, 0.71\rangle : \mathit{supplementary\_charge} \rightarrow \mathit{refund}\\
r'_6\langle 0.83, 0.29\rangle : \mathit{supplementary\_charge}, \mathit{no\_clause} \Rightarrow \neg \mathit{refund}
\end{array}$$

Fig. 1. Defeasible theory generated from available data

Handling Exceptions. Usually the rules that apply to only few cases are seeded out [8]. In our approach marginal rules can be seen rather as exceptions and modeled with defeaters. When a court distinguishes a case it points to some features that make that case different. If we can find such attributes for a particular case, we can formalize it with undercutting defeaters. If it cannot be distinguished by the precedent case it remains a counterexample. Here, the marginal rules are given by the convergence factor. If we establish a threshold of 0.2 only rule 3' with a $\mathit{cov} = 0.14$ is considered marginal. For this particular rule, suppose that 3 out of the $N = 4$ cases in table 2 have, compared to the rule 6', a supplementary attribute *item_broken_by_client* (not used in the first phase of approximation). This exception is modeled with the undercutting defeater r'_{31}, with the certainty factor $3/4 = 0.75$ and the coverage $1/28 = 0.035$. For the remaining rule of the $N = 4$ cases there is no known distinguished attribute.

$$\begin{array}{l}
r'_1\langle 1, 0.29\rangle : \mathit{supplementary\_charge}, \mathit{no\_refund\_clause} \rightarrow \mathit{refund}\\
r'_2\langle 1, 0.57\rangle : \mathit{incorrect\_price} \rightarrow \mathit{refund}\\
r'_{31}\langle 0.75, 0.035\rangle : \mathit{item\_broken\_by\_client} \leadsto \mathit{refund}\\
r'_{32}\langle 0.127, 0.105\rangle : \mathit{supplementary\_charge}, \mathit{no\_clause} \not\Rightarrow \neg \mathit{refund}\\
r'_4\langle 1, 0.71\rangle : \mathit{supplementary\_charge} \rightarrow \mathit{refund}\\
r'_6\langle 0.873, 0.29\rangle : \mathit{supplementary\_charge}, \mathit{no\_clause} \Rightarrow \neg \mathit{refund}
\end{array}$$

Fig. 2. Handling exceptions and counterexamples

[3] The number of all cases satisfying the decision attribute *refund* is 8+1+4+6=28, while for ¬*refund* is 20+50=70. Each judicial case has some meta-data attached, such as court which filed the case or data of judgment. The indiscernability relation of the cases in the rough set approach can be considered for both attributes of the case and these meta-data. In the above computation, we consider all the cases having the same relevance. By taking meta-data into consideration, one can introduce legal strategies such as *legis posterior* or *legis superior*. Under the *legis posterior* doctrine, instead of simply counting the cases, one may compute a weighted sum with the contribution of each case. According to the *legis superior* principle, the outcome imposed by the stronger court takes precedence when computing the certainty factor of a defeasible rule.

Therefore it represents a counterexample and is modeled with the negated rule r'_{32} in figure 2. The increase in the certainty factor of the rule r'_6 covers the idea in common low that exception proves the rule in the case not excepted[4]. Note also, the high confidence assigned to the defeater r'_{31}, value reflecting one of the legal principles for conflict resolution, known as *legis specialis*[5].

4 Mediation in ODR

Since the inclination of mediators to accept as evidence the facts conveyed by the disputants vary, the following components are relevant for the overall process.

4.1 Inference Rules in Argumentation

Given a conclusion q that can be derived based on strict premises in the rule $r[\gamma]$, meaning that the consequent is inferred to a degree of γ_r, the same conclusion might also be derived up to a certain degree $\beta_q \leq \gamma_r$ using defeasible premises. There are two inherited sources of uncertainty: either the premises represent vague concepts or they represent a clearly defined concept, but the facts can only be approximately assigned to the concept represented by the premises. Consequently, the reliance on each conclusion depends both on the certainty factor of the rule (representing how strong the premises sustain the conclusion) and also on the membership function characterizing the premises. We consider two complementary inference methods: fuzzy-based, and rough-sets-based.

Fuzzy Inference. Using the weakest link principle for deductive arguments [9], the conclusion q_0 is as good as the weakest premise, given by $min(\beta_1, ..., \beta_k)$, where β_i is the fuzzy membership. Additionally, the strength of the consequent is also influenced by the certainty factor γ of the inferencing rule, with $\beta_0 = min(\beta_i, \gamma), i \in [1..k]$ (in the fuzzy approach) leading to the following rule.

$$q_0[\beta_0] \xleftarrow{\gamma} \bigwedge_{i \in [1..k]} q_i[\beta_i]$$

Rough Inference. The approximation of similarity [10] considers that it would be correct to reason with the neighbors of perceived values, instead of a crisp probability value. The conclusion q_1 would inherit the inaccuracy of the perceived premises and their associated tolerance spaces. One must assume that every

[4] Known as *Exceptio probam regulam in casibus not exceptis.* Another interpretation in legal practice is: if an excepting clause makes it impermissible when there is no excepting clause, that it is necessary that it is permissible, which fits perfectly with the semantics of defeaters. The "necessary" term is relaxed by introducing the certainty factor.

[5] Under this doctrine the most specific norm takes precedence.

attribute in table 3 has a lower and upper approximation, taking into account the inherent perceptual or contextual limitations, in the latter case two intervals being defined.

In the case of continuous variables two intervals are defined, using the double interval notation $\langle UAI : LAI \rangle$ [11], with $UAI = [u^s, u^e]$ representing the interval where the fact is defeasibly derived, and $LAI = [l^s, l^e]$ the interval where the fact is certainly true. With $u^s_0 = max(u^s_i)$, $u^e_0 = min(u^e_i)$, $l^s_0 = max(l^s_i)$, $u^e_0 = min(u^e_i)$, $i \in [1, k]$ we have the following rough rule.

$$q_0[u^s_0, u^e_0] : [l^s_0, l^e_0] \xleftarrow{\gamma} \bigwedge_{i \in [1..k]} q_k[u^s_k, u^e_k] : [l^s_k, l^e_k]$$

The certainty factor within the UAI is computed according to a rough membership function meeting three constraints: complementarity, nonmonotonicy, border conditions [11]. The function is designed by the mediator taking in consideration statistical data and some sense of symmetry.

4.2 Level of Acceptance

The next step consists in determining the minimum degree of certainty assigned to a defeasible premise within the UAI in order to act as a valid antecedent when it fires a rule. For each antecedent a supporting the consequent q, we use a rough membership coefficient derived from the initial dataset: $\beta^a_{min}(q) = support(a)/support(a, q)$. For instance, in table 3, $\beta^{supplementary\_charge}_{min}(refund)$ $= (8+4)/(8+4+20) = 0.375$ and for $\beta^{contractual\_clause}_{min}(refund) = 4/(4+20) =$ 0.17. This parameter acts as a guideline in the process of accepting a fact as reliable for the current situation, based on the intuition that the attributes which highly influence the outcome must meet a similar level of certainty to be accepted in the argumentation process. Some adjustments, in the spirit of tolerance spaces [10] may be useful, by considering the $\alpha \times \beta^a_{min}(outcome)$, where α depends both on the reliance or importance given to the dataset in the current dispute and on the phase of the current dispute, in both cases $\alpha \longrightarrow 1$[6]. Having designed a rough membership function for each attribute, the mediator can extract the certainty factor β_q for that fact q. If $\beta_q \geq \alpha \times \beta^q_{min}(outcome)$, the premise will be accepted in the argumentation process.

4.3 Accrual of Arguments

The same conclusion q can be sustained by several pro and counter arguments with different degree of reliance $\beta_q \geq \alpha \times \beta^q_{min}(outcome)$. The accrual of argu-

[6] It is possible to define thresholds to decide, during dispute resolution phases whether a given claim is fulfilled or not. The certainty factor of the conclusion must meet the level of confidence accepted for each dispute phase: scintilla of evidence in dispute commencement (20%), preponderence of evidence in discovery phase (50%), clear and convincing evidence (75%) in arbitration phase, and beyond reasonable doubt in post-trial or appeal(95%), thresholds corresponding to credulous, caution, or respectively skeptical attitudes [12].

ments does not hold in our approach, since the types of defeaters are treated differently to achieve different patterns of defeasible reasoning [9]. The strongest undercutting defeater contributes to the decrease of the certainty factor and if the remained strength of the conclusion overwhelms the most powerful rebuttal defeater, the respective conclusion is derived.

$$\beta_q = \begin{cases} max(\beta_{q_i}) - max(\beta_{\sim q_j}) & \mathit{if}\ max(\beta_{q_i}) - max(\beta_{\sim q_j}) > max(\beta_{\sim q_k}) \\ 0 & \text{otherwise} \end{cases}$$

4.4 Defeasible Derivation of a Consequent

Having defeasible rules and accepted valid premises with a level of certainty, the next step consists in inferring the possible consequences. Next we present the derivation formula of a consequent according to the argumentation semantics of the defeasible logic. A conclusion in a defeasible logic theory is a tagged literal which can have the following forms: i) $+\Delta q : \Leftrightarrow q$ is definitely provable using only strict facts and rules (figure 3); ii) $-\Delta q \Leftrightarrow q$ is not definitely provable; iii) $+\partial q : \Leftrightarrow q$ is defeasible provable (figure 4); iv) $-\partial q \Leftrightarrow q$ is not defeasible provable. A conclusion q is strictly provable (figure 3) if (1) q is a strict fact valid or (2) there exists a strict rule r with conclusion q which rule (2.1) have all its antecedents a valid and (2.2) there is no a strict negated rule ns, attacking the rule r.

```
$+\Delta$:
    If $P(i+1) = +\Delta q$ then
        (1) $\exists q(\beta) \in F$ and $\beta = 1$ or
        (2) $\exists r \in R_s[q]$ so that
            (2.1)$\forall a \in A(r) + \Delta a \in P(1..i)$
            (2.2) $\nexists ns \in R_{ns}[r]$
```

Fig. 3. Definite proof for the consequent q

The sentence q is defeasibly provable[7] (figure 4) if (1) it is strictly provable, or (2) there is a valid support for q either (2.1) it is a defeasible valid fact, or (2.2) there exists a rule with all premises valid sustaining that conclusion q and it is not defeated by (2.3) a negated rule with a stronger certainty factor, or (2.4) by an undercutting defeater *def* where (2.4.1) the defeater has an antecedent a which cannot be derived, or (2.4.2) there exists a negated defeater stronger than *def*, and (2.5) for all rebuttal defeaters d either (2.5.1) there is a negated rule which defeats d or (2.5.2) the support for conclusion q after it is attacked by the undercutter defeaters remains stronger than all the valid rebuttal defeaters. The non strict order relation in (2.3), (2.4.2), and (2.5.2) does not provide a skeptical

[7] The answer to a defeasible query is based on premises situated in the boundary region, similar to the concept of approximation queries [10].

```
$+\partial$:
  If $P(i+1) = +\partial q$ then
  (1) $+\Delta q \in P(1..i)$ or
  (2) q is supported
      (2.1) $\exists q(\beta_q) \in F$ and $\beta^q_{min} < \beta_q < 1$ or
      (2.2) $\exists r(\gamma_r) \in R_{sup}[q]$, $\forall a \in A(r)$ so that $+\partial a \in P(1..i)$, $\beta_a \geq \beta^a_{min}$, not defeated
      (2.3) $\forall nd(\gamma_{nd}) \in R_{nd}[r] \cup R_{ns}[r], \gamma_r \geq \gamma_{nd}$ and
      (2.4) $\forall def(\gamma_{def}) \in R_{def}[q]$ or
          (2.4.1) $\exists a \in A(def) - \partial a$ or
          (2.4.2) $\exists ndef(\gamma_{ndef}) \in R_{ndef}[def],\ \gamma_{ndef} \geq \gamma_{def}$ and
      (2.5) $\forall d(\gamma_d) \in R_{sup}[\sim q]$ with $\forall a(\beta_a) \in A(d)$, $+\partial a$ and $\beta_a \geq \beta^a_{min}$ either
          (2.5.1) $\exists nnd(\gamma_{nnd}) \in R_{nd}[d] \cup R_{ns}[d], \gamma_{nnd} > \gamma d$, or
          (2.5.2) $\gamma_r - \gamma_{def} \geq \gamma_d$
```

Fig. 4. Defeasible derivation of consequence q

reasoning mechanism, meaning that both of q and $\sim q$ may be derived when they have equal support[8].

5 Discussion and Related Work

The essential advantage of our approach compared to existing ones for deriving defeasible theories from data [3,13] is that we do not assume that the available information is consistent. The reasoning pattern in HeRO [3] is conservative since it does not draw any conclusion in case of doubt, a very hard (skeptical) constraint for ODR systems, which has been relaxed in our framework.

The Apriori algorithm used to generate association rules can facilitate the discovery of defeasible rules [13] by suggesting hypotheses, the candidate defeasible rules reduced later by applying support, confidence and interest metrics. In the rough set approach the irrelevant defeasible rule candidates are not even computed, because they are derived from reducts which contain only the important attributes. Our approach is adapted to the constraints of practical applications where the information is revealed gradually and in the first phases indiscernability is a fact of life.

To the best of our knowledge this is the first attempt to combine rough set theory with defeasible reasoning, aiming to cover both aspects of the law: case based reasoning and legal syllogism. Given the current demand for ODR technologies, our future work will enrich the framework with explanation capabilities of the outcome. The variant of defeasible logic proposed here offers a rich knowledge representation formalism and a clearly interpreted theory, with adequate argumentative semantics for legal reasoning.

[8] In some cases the law gives no straight answer and consequently judges can legitimately decide either way. Allowing ambiguity propagation increases the number of inferred conclusions, useful in the argumentation process, in ODR systems oriented towards solution rather than finding the degree of guilt.

References

1. Tyler, M.C., Bretherton, D.: Seventy-six and counting: An analysis of ODR sites. In: Workshop on Online Dispute Resolution at the International Conference on Artificial Intelligence and Law, Edinburgh, UK, pp. 13–28 (2003)
2. Walton, D., Godden, D.: Persuasion dialogues in online dispute resolution. Artificial Intelligence and Law 13, 273–295 (2006)
3. Johnston, B., Governatori, G.: An algorithm for the induction of defeasible logic theories from databases. In: 14th Australasian Database Conference, Darlinghurst, Australia, pp. 75–83 (2003)
4. Rule, C., Friedberg, L.: The appropriate role of dispute resolution in building trust online. Artificial Intelligence and Law 13, 193–205 (2006)
5. Hage, J.: Law and defeasibility. Artificial Intelligence and Law 11, 221–243 (2003)
6. Pawlak, Z.: A primer on rough sets: a new approach to drawing conclusion from data. Cardozo Law Review 22, 1407–1415 (2002)
7. Li, J., Cercone, N.: Introducing a rule importance measure. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, pp. 167–189. Springer, Heidelberg (2006)
8. Szczuka, M.: Techniques for managing rule-based concept approximation. In: International Workshop on Soft Computing at Intelligent Agent Technology, Compiegne, France, pp. 31–35 (2005)
9. Pollock, J.L.: Defeasible reasoning with variable degrees of justification. Artificial Intelligence 133, 233–282 (2001)
10. Doherty, P., Lukaszewicz, W., Szalas, A.: Communication between agents with heterogeneous perceptual capabilities. Information Fusion 8, 56–69 (2007)
11. Rebolledo, M.: Rough intervals – enhancing intervals for qualitative modeling of technical systems. Artificial Intelligence 170, 667–685 (2006)
12. Parsons, S., Wooldridge, M., Amgoud, L.: Properties and complexity of some formal inter-agent dialogues. Journal of Logic and Computation 13, 347–376 (2003)
13. Governatori, G., Stranieri, A.: Towards the application of association rules for defeasible rules discovery. In: Verheij, B., Lodder, A., Loui, R.P., Muntjerwerff, A.J. (eds.) Legal Knowledge and Information Systems, pp. 63–75. IOS Press, Amsterdam (2001)

Semantic Service Discovery with QoS Measurement in Universal Network

Ying Zhang, Houkuan Huang, Youli Qu, and Xiang Zhao

School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, P.R. China
dearzppzpp@gmail.com, hkhuang@center.njtu.edu.cn, ylqu@bjtu.edu.cn, zhaotomx@sohu.com

Abstract. The service of Universal Network is different from that of current network, because the former has QoS (Quality of Service) grading. Therefore, service discovery of Universal Network is quite distinct from that of present network. In this paper, we present service discovery with QoS measurement to adapt to Universal Network. Many researches adopt semantic web technology-OWL-S (Web Ontology Language for Services), which is innovative for service discovery. With the aim of service discovery in Universal Network, we append QoS descriptions to OWL-S. Such OWL-S with QoS information is called OWL-QoS which is the groundwork for service discovery. Moreover, we also propose a matching algorithm that allows matching on the bases of capabilities and QoS descriptions of services.

Keywords: service discovery, semantic service, QoS, Universal Network.

1 Introduction

The Universal Network, which combines Telecom Network with IP Network, is under developing. Providing services based on it for clients is a core of research in the world. Providing QoS grading is one of the most important features in the Universal network. Present Internet supplies best-effort services, which can not meet users' requirements. Users often need service with specified QoS.

The promotion of services has stimulated providers to develop and publish their services. Consequently, service requesters can discover services which they want through looking up registry center. In the Universal Network, Services with distinct QoS will be published in the registry center, and requesters can get services with distinct QoS. Thus, discovering services now requires more sophisticated pattern.

In this paper we concentrate on discovering services with QoS measurement in Universal Network for satisfying requesters' high-grade demand. In practice, we divide service discovery into two steps. In the first step, a requester discovers the service using the basic ability description-what the service does, its input and output parameters, preconditions, and effects [1]. This step satisfies the requester's basic need. The second step is to identify sufficiently similar service

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 707–715, 2007.

for the requester with QoS measurement. This second step is the emphasis of our research.

At present, many service discovery processes use keyword-matching technique to find published services. This method often discontents requester with so many unrelated results and leads to a bit of manual work to choose the proper service according to its semantics. In order to realize automatic discovery, we adopt semantic web technique-OWL-S, which is innovative for service discovery. With OWL-S markup of services, the information necessary for service discovery could be specified as computer-interpretable semantic markup at the service registry or ontology-enhanced search engine could be used to locate the services automatically [2].

OWL-S provides three essential types of knowledge about a service: a service profile (what the service does), a service model (how the service works), and a service grounding (how to use the service). The service profile describes what the service can do, for purposes of advertising, discovery, and matchmaking [3]. It describes the basic ability of service, so it is helpful for fulfilling the requirement of the first step of service discovery mentioned afore.

In order to accomplish the second step of service discovery, we propose to add QoS descriptions to OWL-S to specify the service's QoS information in Universal Network for satisfying users' high-class requirement.

The rest of this paper is organized as follows: Section 2 introduces the architecture of the Universal Network. Section 3 gives details about OWL- QoS which is ontology with QoS descriptions. Section 4 discusses a matching algorithm between advertisements and requests described in OWL- QoS that recognizes various degrees of matching. Section 5 provides concluding remarks.

2 Services in the Universal Network

In the present Information Network, one kind of network mostly supports one kind of service. For instance, Telecom Network basically faces phonetic business while IP Network mainly supports data business. Due to the limitation of the original model, the existing network can not satisfy diversified requirements essentially. It is very meaningful to form Universal Network. The universal Network Model is specified in [16] and [17].

According to [4], we give the definition of the service in the Universal Network: Service is self-contained, modular applications that can be described, published, located, and invoked over the Universal Network.

Providing various qualities of services for different users is one of the greatest features of the Universal Network.Quality depends on user's request and pay. Paying more can get more. It means if user wants to have high quality service then he should pay more.

QoS refers to connectivity, security and so on. The details will be discussed in section 3.1. Pay means the spending of the user. For example, how much money is invested.

3 Ontology with QoS

A fundamental component of the Semantic Web will be the markup of services to make the computer-interpretable, use-apparent, and agent-ready. The Web Ontology Language for Semantic Web Services (OWL-S) supports automatic service discovery via matchmakers through its service profile language constructure [5]. For the sake of automatic service discovery, we adopt OWL-S markup for describing services and so we can call the services marked by OWL-S as semantic services. We can also call service discovery with OWL-S as semantic service discovery. For the purpose of discovery with proper QoS in Universal Network, we append service's quality description to service profile. We will discuss QoS in detail as follows.

3.1 The Definition and Classification of QoS

QoS Definition[6]. According to ITU-T QoS Study Group the term of QoS is defined as: "collective effect of service performances that determine the degree of satisfaction by a user of the service" (ITU-T R.E.800) [7].

International Organization for Standardization has proposed another definition in ISO/IEC 10746-2 [8] for the term of QoS as follows:

"a set of qualities related to the collective behavior of one or more objects" and the Internet Engineering Task Force (IETF) Network Working Group has proposed the following QoS definition in RFC 2386 [9]:

"a set of service requirements to be met by the network while transporting a flow"

We can see that ITU-T Study Group gives definition from the user's point of view while IETF Network Working Group defines QoS from network's point of view.

QoS Classification. From network's point of view, we provide QoS with several properties: Delay, Loss Probability, and QoS Spectrum.

Delay is classified into four grading:

1. Delay Time $\leqslant$ 1s(It is applied to Interactive Service)
2. 1s $<$ Delay Time $\leqslant$ 3s (It is applied to Response-Service)
3. 3s $<$ Delay Time $\leqslant$ 10s (It is applied to Timely Service)
4. Delay Time $>$ 10s (It is applied to Delay-Insensitive Service)

Loss Probability has three classes according to threshold which is given by the Universal Network as f ollows:

1. Loss Probability $\ll \tau$
2. Loss Probability is approximate to τ
3. Loss Probability $\gg \tau$

While Expedited Forwarding, Assured Forwarding and Best Effort [10] belong to QoS Spectrum.

Properties of QoS are not only Delay, Loss Probability, and QoS Spectrum, but also Connectivity, Security and Trustworthy Degree from user's point of view.

Table 1 shows the mapping from user's view to network's view. This mapping is prepared for the matching of user's request and network's supply.

Table 1. Mapping from user's view to network's view

QoS properties from user's view		QoS properties from network's view	
Connectivity	Excellent	Delay	Delay Time $\leqslant$ 1s
	Good		1s $<$ Delay Time $\leqslant$ 3s
	bad		Delay Time $>$ 10s
Security	High	Loss Probability	Loss Probability $\ll \tau$
	Medium		Loss Probability $\approx \tau$
	Low		Loss Probability $\gg \tau$
Trustworthy Degree	High	QoS Spectrum	Expedited Forwarding
	Medium		Assured Forwarding
	Low		Best Effort

3.2 OWL-QoS

We call our ontology OWL-QoS which is designed for Universal Network; it is a complementary ontology that provides detailed QoS information for OWL-S.

Original Service Profile. An OWL-S Profile describes three types of information: the organization that supplies the service, the function of the service, and the features of the service. The provider information consists of contact information that refers to the entity that provides the service. Specifically, the functional description of the service specifies the input required by the service, the output generated, the preconditions required by the service and the expected effects. The features specify the category of a given service, quality rating of the service and so on [2].

The provider information and the feature descriptions are nonfunctional aspects of the description, while the function of the service is functional aspect of the description.

Appending QoS specification to Service Profile. As service profile mostly supports automatic discovery of the service, we add QoS specification to it for adapting to the service discovery of the Universal Network and it forms OWL-QoS.

The new service profile model which includes QoS is depicted as Fig.1.

Others are classes and properties which are the same as those in [2]. Class QoS is the common superclass for all QoS specification. RequesterQoS and ProviderQoS are subclasses of class QoS. RequesterQoS is the requester's QoS description and ProviderQoS is the QoS description from the Universal network's viewpoint.

ProviderQoS has three object properties which are mentioned afore: Delay, Loss Probability and QoS Spectrum. DelayValue and LossProValue which are data properties belong to class Delay and class LossPro respectively. Class QoSSpe owns three data properties: ExpeditedForw, AssuredForw and Best Effort.

RequesterQoS also has three object properties mentioned in section 3.1. They are Connectivity, Security and Trustworthy Degree. Data properties such as Excellent, Good and Bad belong to Class Connectivity. Both Class Security and Class Trustworthy Degree have three identical data properties: High, Medium and Low.

We give definitions of classes QoS, ProviderQoS, and Delay in profile as follows. Definitions of RequesterQoS and ProviderQoS are identical while the definitions of Loss Probability, QoS Spectrum, Connectivity, Security and Trustworthy Degree are similar to the definition of class Delay.

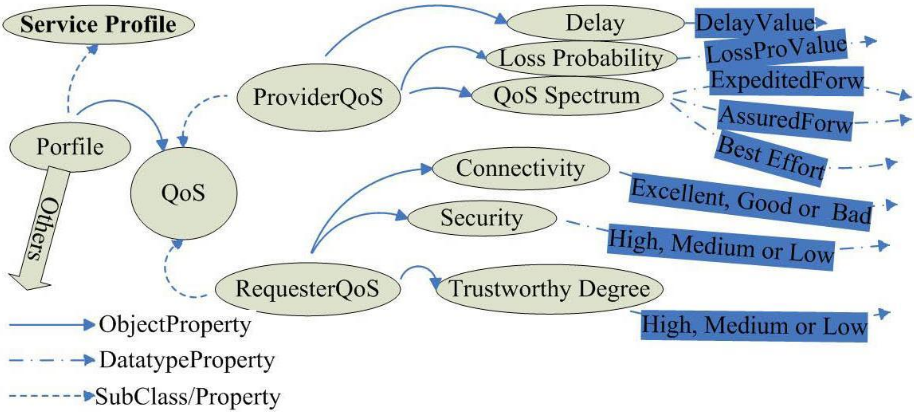


Fig. 1. Service profile model with QoS

4 Semantic Service Discovery with OWL-QoS

Semantic service discovery is a process for location of semantic services that can provide a particular class of service capabilities and a specified QoS, while adhering to some client-specified constrats [2]. Using OWL-QoS markup of services of Universal Network, the information that is useful for service discovery could be specified as computer-interpretable semantic markup at the service web sites [11]. A server could proactively advertise itself in OWL-QoS with a service registry, which is also called middle agent [2, 12, 13, 14], while requesters can find the needed services when they query the registry. So OWL-QoS is helpful for automatic service discovery.

4.1 Semantic Service Discovery Model

Because the services of the Universal Network are graded according to the properties of QoS, it is essential to use profile with QoS for semantic service discovery. Fig.2 shows the semantic service discovery model with QoS measurement.

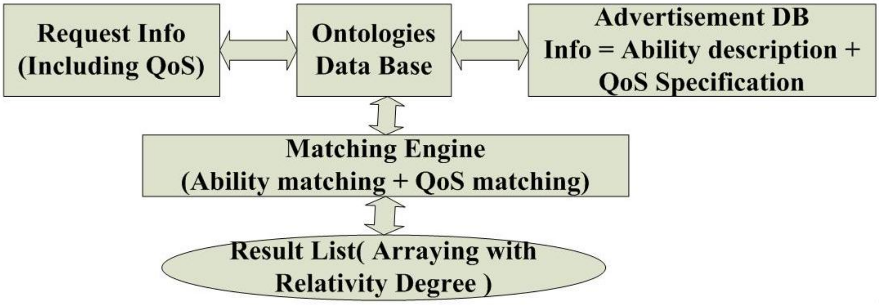

Request Info (Including QoS)
Ontologies Data Base
Advertisement DB Info = Ability description + QoS Specification
Matching Engine (Ability matching + QoS matching)
Result List(Arraying with Relativity Degree)

Fig. 2. Service profile model with QoS

Advertisements DB stores advertisements provided by service providers and the advertisement info includes basic ability description and QoS specification of service. After receiving a request, the Matching Engine chooses the advertisements that are relevant for the current request from Advertisements DB according to Matching Algorithm. This Matching Algorithm will be detailed in the next section. It includes QoS matching as its one point.

4.2 Matching Algorithm with QoS Measurement

Matching algorithm is a key for semantic service discovery. Some earlier algorithms are too restrictive. An advertisement matches a request, when the advertisement and the request describe exactly the same service. This is too restrictive, because advertisers and requesters have no prior agreement on how a service is represented and they have different objectives. Such restrictive a match inevitably bounds to fail to recognize similarities between advertisements.

We give a flexible matching algorithm. The result of the match is not a hard true or false, however it relies on the degree of similarity between the concepts in the match. The matching algorithm composes of two parts: basic ability matching and QoS matching.

In this section we will discuss the matching algorithm in some detail. At first we will present the main program. The request is matched against all advertisements stored in Advertisements DB in Fig.2. When a match between the request and any of the advertisement is found, it is recorded and scored to find the matches with the highest degree [15].

```
match(Request) {
   ResultMatch = empty
    forall Adv in advertisements do {
      if Match(Request, Adv) then
         ResultMatch.add(Request, Adv) }
         return Sort(ResultMatch);}
```

A match between an advertisement and a request consists of the match of all the outputs of the request against the outputs of the advertisement; all the inputs of the advertisement against the inputs of the request; all the QoS requirements of the request against the QoS requirements of the advertisement. The algorithm for output matching is described as follows. The degree of success is due to the degree of match detected. The QoS matching algorithm is the same as output matching's. The matching algorithm of inputs is similar to that of outputs, but with the reversed order of the request and the advertisement, i.e. the advertisement's inputs are matched against the request's inputs [15].

```
OutputMatch(OutputsRequest,OutputsAdvertisement)
  globaldegreeMatch =Excellent
  forall OutR in outputsRequest do
  find OutA in OutputsAdvertisement such that
        degreeMatch =maxdegreeMatch(OutR,OutA)
    if (degreeMatch = error) return Failed
    if (degreeMatch < globaldegreeMatch)
     globaldegreeMatch = degreeMatch
    return globaldegreeMatch;
```

The degree of match between two outputs, two inputs and two QoS depends on the relation between the concepts associated with those inputs, outputs and QoS. We give rules for outputs matching degree and QoS matching degree respectively. The inputs matching degree rules are identical to ouputs'.

```
degreeMatch(OutR,OutA)
  if OutA = OutR then return Excellent
  if OutR is a subclassOf OutA then return Exact
  if OutA subsumes OutR then return PlugIn
  if OutR subsumes OutA then return Subsumes
  otherwise Error

degreeMatch(QoSR,QoSA)
  if QoSA = QoSR then return Excellent
  if |QoSR-QoSA| = 1  then return Distinguishing
  if |QoSR-QoSA| > 1 then return Distinguishing
  otherwise Error
```

In the above QoS degree matching, |QoSR-QoSA| = 1 means that according to table 1, the distance between QoSR and QoSA is equal to 1. For instance, if requester requires that the connectivity isent and the advertisement provides 1s<Delay Time $\leqslant$ 3s, then the distance between QoS of requester and QoS of provider amounts to 1.

At last, the sorting rule is showed. It reflects that it will select the match with the highest score in the outputs firstly. If the outputs' matching scores are equal, then choose the match with the highest score in the QoS. Finally input matching is used only to break ties between equally scoring outputs and equally scoring QoS.

```
SortingRule(Match1,Match2) {
   if Match1.output > Match2.output then Match1 > Match2

   if Match1.output = Match2.output & Match1.QoS >
      Match2.QoS then Match1 > Match2

   if Match1.output = Match2.output & Match1.QoS =
      Match2.Oos & Match1.input > Match2.input then
      Match1 > Match2

   if Match1.output = Match2.output & Match1.QoS =
      Match2.Qos & Match1.input = Match2.input then
      Match1 = Match2}
```

5 Conclusion

Services are classified and have various qualities in Universal Network, so service discovery of Universal Network is different from that of current Network. This paper contributes to this challenge by presenting semantic descriptions of services with QoS measurement, and we call it OWL-QoS. Describing a matching algorithm with OWL-QoS is another contribution of this paper. This algorithm allows matching of advertisements and requests not only on the bases of the capabilities that they describe, but also on QoS which they specify. In the future, the research on Universal Network will still be a point. As part of our future work, we would like to delve into automatic service invocation, composition and interoperation in Universal Network.

Acknowledgments. Our research is supported by the national Grand Fundamental Research 973 Program of China under Grant (No.2007CB307100, No.2007CB307106).

References

1. Zhou, C., Chia, L.-T., Lee, B.-S.: Semantics in service discovery and QoS measurement. IEEE IT professional 7(2), 29–34 (2005)
2. Martin, D. (ed.): OWL-S: Semantic Markup for Web Services OWL-S Semantic Markup for Web Services.htm (2003) http://www.daml.org/services/owl-s/

3. McIlraith, S.A., Martin, D.L.: Bringing semantics to Web services. IEEE Intelligent Systems 18(1), 90–93 (2003)
4. Gottschalk, K. (ed.): Web Services Architecture Overview: The Next Stage of Evolution for E-Business. http://www-106.ibm.com/developerworks/library/w-ovr
5. Sycara, K., Paolucci, M., Soudry, J., Srinivasan, N.: Dynamic discovery and coordination of agent-based semantic Web services. IEEE Internet Computing 8(3), 66–73 (2004)
6. Muntean, C.H.: Ph.D.thesis: Quality of Experience Aware Adaptive Hypermedia System.
http://www.eeng.dcu.ie/~pel/graduates/graduate-cristina-hava.html
7. ITU-T Recommendation E.800: Terms and Definitions Related to Quality Service and Network Performance Including Dependability (1994)
8. ISO/IEC 10746-2: Information Technology-Open Distributed Processing-Reference Model:Foundations.International Standards Organisation (1996)
9. Crawley, E.S., Nair, R., Rajagopalan, B., Sandick, H.: A Framework for QoS-based Routing in the Internet. RFC 2386. http://www.ietf.org/rfc/rfc2386.txt
10. IETF, An Architecture for Differentiated Services [I], RFC2475 (1998)
11. McIlraith, S.A., Son, T.C., Zeng, H.: Semantic Web services. IEEE Intelligent Systems 16(2), 46–53 (2001)
12. Decker, K., Sycara, K., Williamson, M.: Middle-agents for the Internet. In: IJCAI97 (1997)
13. Wong, H.-C., Sycara, K.: A Taxonomy of Middle-agents for the Internet. In: ICMAS'2000 (2000)
14. Martin, D., Cheyer, A., Moran, D.: The Open Agent Architecture: A Framework for Building Distributed Software Systems. Applied Artificial Intelligence 13(1-2), 92–128 (1999)
15. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.: Semantic Matching of Web Services Capabilities. In: Horrocks, I., Hendler, J. (eds.) The Semantic Web - ISWC 2002. LNCS, vol. 2342, pp. 333–347. Springer, Heidelberg (2002)
16. Zhang, H.: An architecture of Universal Network Services. Patent Application No. 200510134579.1
17. Zhang, H.: An architecture of Universal Network Services. Patent Application No. 200510134579.1

Rough Sets in the Interpretation of Statistical Tests Outcomes for Genes Under Hypothetical Balancing Selection

Krzysztof Cyran

Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
krzysztof.cyran@polsl.pl

Abstract. Detection of natural selection at the molecular level is one of the crucial problems in contemporary population genetics. There exists a number of statistical tests designed for it, however, the interpretation of the outcomes is often obscure, because of the existence of factors like population growth, migration and recombination. In his earlier work, the author has proposed the multi-null methodology, and he applied it for four genes implicated in human familial cancer: ATM, RECQL, WRN and BLM. Because of high computational effort required for estimating critical values under nonclassical nulls, mentioned methodology is not appropriate for selection screening. In the current paper, the author presents novel, rough set based methodology, helpful in the interpretation of tests outcomes applied versus only classical nulls. This method does not require long-lasting simulations and, as it is shown in the paper, it gives reliable results.

Keywords: rough sets, natural selection, ATM, BLM, RECQL, WRN, neutrality tests.

1 Introduction

Widely accepted Kimuras neutral model of evolution [1] states that, at the molecular level, the majority of genetic variation is caused by the selectively neutral forces like silent mutations and a genetic drift. Nevertheless, the model does not contradict the existence of selection at molecular level, although the role of it is not so important, as it had been thought before Kimuras work. When this work was published and, after some discussion, accepted, the majority of the genome was assumed to be selectively neutral. However, it is obvious that some mutations must be deleterious (and in fact we know many of such mutations causing serious genetic dysfunctions), some must be selectively positive (at least when the environment is changing) and some are known to be responsible for a phenomenon called balancing selection. Perhaps the most representative example of a positive selection is the ASMP locus, which is a major contributor to the brain size in primates [2,3]. Yet, even if the number of positive selections found grows

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 716–725, 2007.

up, the evidence for balancing selection is not so numerous. Therefore, the detection of the signatures for balancing selection operating at the molecular level remains one of the crucial problems in contemporary population genetics.

There exists a number of statistical non-neutrality tests [4,5,6,7] designed for the detection of such a selection in a gene under study. However, the interpretation of the outcomes of tests is hard because of the existence of factors like population growth, migration and recombination, which are not included in classical null hypothesis [8]. In his earlier work (published in part in [9] and in part unpublished), the author has proposed the multi-null hypothesis methodology, and using it, he was able to detect the signatures of a balancing selection in genes implicated in human familial cancer: in ATM (ataxia-telangiectasia mutated) and in a helicase involved in a repair of the DNA called RECQL. He also confirmed no evidence of such a selection in two other DNA helicases: WRN (Werners syndrome, see [10]) and BLM (Blooms syndrome, see [11]).

Because of high computational effort required for computing (by computer simulations) the critical values of the tests under nonclassical null hypotheses, the methodology proposed earlier is not appropriate as a screening tool. In a current paper the author presents rough set based methodology, helpful in the interpretation of tests outcomes, applied versus only classical nulls. The use of rough set theory for knowledge processing was dicated by the fact that test outcomes can be naturally discretized to a few values only, such as statistically non signinficant, statistically signinficant, or strongly statisctically significant. Moreover, since the critical values for classical null hypotheses are known, this method does not require time-consuming computer simulations and, as it is shown in the paper, it gives relatively reliable results.

2 Materials and Methods

As genetic material for this study, there was taken the single nucleotide polymorphisms (SNP) data, taken from the intronic regions of target genes. They form haplotypes, which can be used as tools to investigate the genetic diversity and possible disease associations. The first locus analyzed is ataxia-telangiectasia mutated (ATM) [12,13]. The ATM gene product is a member of a family of large proteins implicated in the regulation of the cell cycle and in the response to DNA damage. The other three genes include: human helicase RECQL, Blooms syndrome (BLM) and Werners syndrome (WRN). The products of these three genes are DNA helicases, enzymes involved in various types of DNA repair, including mismatch repair, nucleotide excision repair, and direct repair. A number of interesting facts about these genes were determined, including the question of selection signatures, addressed by the author and his co-workers [9].

The ATM gene, located in human chromosomal region 11q22-q23, spans 184 kb of genomic DNA. The intron-exon structure of the WRN locus spanning 186 kb at 8p12-p11.2 includes 35 exons, with the coding sequence beginning in the second exon. RECQL is composed of 15 exons, located at 12p12-p11 and spans 180 kb, whereas BLM, mapped to 15q26.1, has 22 exons and spans 154 kb. Blood samples for this study were collected from the residents of Houston, TX, from four major ethnic groups: Caucasians, Asians, Hispanics, and African-Americans.

To detect departures from the neutral model, the following statistics were used: Tajimas (1989) T (for uniformity, we follow here the nomenclature of Fu [5] and Wall [7]), Fu and Lis (1993) F^* and D^*, Kellys (1997) Z_{nS} and Walls (1999) Q and B, as well as Strobecks S test. The definitions of these statistics can be found in original works of the inventors, as well as, in a brief form, in Cyran et. al. (2004) pilot study [9].

In this study the rough set based method is used to simplify the process of determining whether the given gene is exhibiting the signatures of balancing selection or not. Such a selection (if present) is reflected by statistically significant departures from the null of the Tajimas and Fus tests towards positive values. However, not all such departures are indeed caused by a balancing selection [8], since such factors like population change in time, migration between subpopulations and a recombination can be reflected by similar outcomes of these tests. Therefore, a wide range of tests was included and the problem with the interpretation of their combinations occurred.

In order to apply a rough set based methodology, the decision table was built with tests outcomes treated as conditional attributes and a decision about the balancing selection treated as the only decision attribute. Fortunately, basing on previous studies, using multi-null methodology and heavy computer simulations, the author was able to determine the value of this decision attribute for given combination of conditional attributes. The purpose of this work was to propose and verify that the automatic and reliable interpretation of the battery of tests outcomes (perhaps without using all of them) can be done without application of the time consuming multi-null strategy. Therefore, to find the required set of tests, which is informative about the problem, there was applied the notion of a relative reduct with respect to decision attribute. Also, in order to obtain as simple decision rules as possible, the relative value reducts were used for particular elements of the Universe. To study the generalization properties and to estimate the decision error, the jack-knife crossvalidation technique was used.

3 Results and Discussion

The haplotypes for particular loci were inferred and their frequencies were estimated by using the Expectation-Maximization algorithm [14]. The results of tests T, D^*, F^*, S, Q, B and Z_{nS}, together with the decision concerning the evidence of balancing selection based on multi-null methodology, are given in Table 1.

The rough set based analysis of the Decision Table 1 reveals that there exist two relative reducts: $RED_1 = \{D^*, T, Z_{nS}\}$ and $RED_2 = \{D^*, T, F^*\}$. It is clearly visible that the core set is composed of tests D^* and T, whereas tests Z_{nS} and F^* can be chosen arbitrarily, according to the automatic data analysis. However, since it is known that both Fus tests F^* and D^* are examples of tests belonging to the same family, and therefore their outcomes are rather strongly correlated, it is advantageous to choose Kellys Z_{nS} instead of F^* test. It is so, because Z_{nS} outcomes are theoretically less correlated with outcomes of test D^*, belonging, as it was stated above, to the core and therefore required in any reduct. Generally, the same rule should be applicable also to the cases when the number of reducts is larger than two. However, the actual choice of the appropriate reduct in such a case can be more difficult, and the advise of a genetician should be of great relevance. The Decision Table 1 with set of conditional attributes reduced to the set RED_1 is presented in Table 2.

Table 1. The outcomes of the statistical tests for the classical null hypothesis. The table includes: Fus D^* test Walls B test, Walls Q test, Tajimas T test (known also as Tajimas D), Strobecks S test, Kellys Z_{nS} test, and Fus F^* test. The values of the test are: Non significant (NS) when $p > 0.05$, significant (*) if $0.01 < p < 0.05$, and strongly significant (**) when $p < 0.01$. The last column indicates the evidence or no evidence of balancing selection, based on the detailed analysis according to multi-null methodology.

		D^*	B	Q	T	S	Z_{nS}	F^*	Balancing selection
ATM	AfAm	*	NS	NS	*	NS	NS	*	Yes
	Cauc	*	NS	NS	**	**	*	**	Yes
	Asian	NS	NS	NS	*	NS	*	NS	Yes
	Hispanic	*	NS	NS	**	NS	*	*	Yes
RECQL	AfAm	NS	NS	NS	**	NS	NS	NS	Yes
	Cauc	*	NS	NS	**	NS	NS	**	Yes
	Asian	NS	*	*	*	NS	*	NS	Yes
	Hispanic	*	NS	NS	**	NS	NS	*	Yes
WRN	AfAm	NS	NS	NS	NS	NS	NS	NS	No
	Cauc	*	NS	NS	NS	NS	NS	NS	No
	Asian	*	NS	NS	NS	NS	NS	NS	No
	Hispanic	NS	NS	NS	NS	NS	NS	NS	No
BLM	AfAm	NS	NS	NS	NS	NS	NS	NS	No
	Cauc	NS	NS	NS	*	NS	NS	*	No
	Asian	NS	NS	NS	NS	NS	NS	NS	No
	Hispanic	NS	NS	NS	NS	NS	NS	NS	No

After a reduction of the set of informative tests to set $RED_1 = \{D^*, T, Z_{nS}\}$, there was considered the problem of coverage of the discrete space generated by these statistics, by the examples included in the training set. The results are given in Table 3, and they reveal that in such a space the fraction of points, which are included in training data, is only 30%. The next step was to apply the notion of the relative value reducts to particular decision rules in the Decision Table 2. The resulting new Decision Table is presented in Table 4. Basing on this table, the Decision Algorithm 1 was obtained. Note that this algorithm is simplified as compared to the algorithm that corresponds to the Decision

Table 2. The Decision Table, in which the set of tests is reduced to relative reduct RED_1 composed of tests: D^*, T, and Z_{nS}

		D^*	T	Z_{nS}	Balancing selection
ATM	AfAm	*	*	NS	Yes
	Cauc	*	**	*	Yes
	Asian	NS	*	*	Yes
	Hispanic	*	**	*	Yes
RECQL	AfAm	NS	**	NS	Yes
	Cauc	*	**	NS	Yes
	Asian	NS	*	*	Yes
	Hispanic	*	**	NS	Yes
WRN	AfAm	NS	NS	NS	No
	Cauc	*	NS	NS	No
	Asian	*	NS	NS	No
	Hispanic	NS	NS	NS	No
BLM	AfAm	NS	NS	NS	No
	Cauc	NS	*	NS	No
	Asian	NS	NS	NS	No
	Hispanic	NS	NS	NS	No

Table 2. At the same time, it is more general, which can be observed in Table 5, presenting the information analogous to Table 3. In Table 5, the coverage of points is based on the number of points which are classified with the use of the simplified Algorithm 1. One should notice that the fraction of points covered by algorithm is 74%, however, since 11% is classified as both with and without the evidence of balancing selection, therefore only 63% of the points could be treated as really covered.

Algorithm 1

```
BALANCING_SELECTION If: T = ** or (T = * and D* = *) or ZnS = *
NO_SELECTION If: T = NS or (T = * and D* = NS and ZnS = NS)
```

Purely automatic knowledge processing technique resulting in Algorithm 1, can be further improved by supplying it with the additional information, concerning the domain under study. It is clearly true that, if a balancing selection is determined by the statistical significance of the given test, then such a selection is even more probable when the outcome of this test is strongly statistically significant.

Therefore, instead of equalities in Algorithm 1, there are proposed inequalities in the generalized version referred to as Algorithm 2. Such inequality means that the given test is at least of the value of statistical significance shown to the right of the inequality sign, but it can obviously be also more significant. In other words, the main difference between Algorithm 2, as compared to the Algorithm

Table 3. The discrete space of three tests: D^*, T and Z_{nS}. The domain of each test outcome (coordinate) is composed of three values: ** (strong statistical significance $p < 0.01$), * (statistical significance $0.01 < p < 0.05$), and NS (no significance $p > 0.05$). The given point in a space is assigned to: S (the evidence of balancing selection), N (no evidence of balancing selection) or empty cell (point not covered by the training data). The assignment is done basing on raw training data with conditional part reduced to the relative reduct RED_1 . Note that the fraction of points covered by training examples is only 30%.

		T								
		$**$			$**$			NS		
		Z_{nS}			Z_{nS}			Z_{nS}		
		$**$	$*$	NS	$**$	$*$	NS	$**$	$*$	NS
D^*	$**$									
	$*$		S	S			S			N
	NS			S		S	N			N

Table 4. The set of tests is reduced to reflect the relative reduct composed of tests: D^*, T, and Z_{nS}, and additionally the notion of relative value reduct is used to further reduce the complexity of separate rows in a decision table

		D^*	T	Z_{nS}	Balancing selection
ATM	AfAm	*	*		Yes
	Cauc		**		Yes
	Asian			*	Yes
	Hispanic		**		Yes
RECQL	AfAm		**		Yes
	Cauc		**		Yes
	Asian			*	Yes
	Hispanic		**		Yes
WRN	AfAm		NS		No
	Cauc		NS		No
	Asian		NS		No
	Hispanic		NS		No
BLM	AfAm		NS		No
	Cauc	NS	*	NS	No
	Asian		NS		No
	Hispanic		NS		No

1, is that instead of formulas of the type $testoutcome = *$ it uses formulas of the type $testoutcome >= *$, meaning that the test outcome is at least significant (and perhaps strongly significant).

Algorithm 2 deals also with the problem of contradiction, and in such a case, it generates no decision about the evidence of balancing selection in a gene under study. The problem of covering points in a discrete space generated by three

Table 5. The discrete space of three tests: $D*$, T and Z_{nS}, forming a relative reduct. The domain of each test outcome (coordinate) is composed of three values: ** (strong statistical significance $p < 0.01$), * (statistical significance $0.01 < p < 0.05$), and NS (no significance $p > 0.05$). The given point in a space is assigned to S and N (with the meaning identical to that given in the caption of Table 3), or "-" having the meaning of contradiction between evidence and no evidence of the balancing selection. The space is filled basing on the simplified Decision Algorithm 1, which uses the relative value reducts varying among different training examples. Note that the fraction of points covered is now 74%, but it includes 11% denoting the contradicting decisions, and such a case should be treated as the lack of decision. Therefore, the real fraction of points assigned with some decision is now 63%.

		T								
		$**$			$**$			NS		
		Z_{nS}			Z_{nS}			Z_{nS}		
		$**$	$*$	NS	$**$	$*$	NS	$**$	$*$	NS
D^*	$**$	S	S	S					-	N
	$*$	S	S	S	S	S	S		-	N
	NS	S	S	S		S	N		-	N

tests in Algorithm 2 is presented in Table 6. This table shows that all points are covered by Algorithm 2, yet since 22% are designated as contradictions, therefore 78% points in a space are really recognizable by this algorithm.

Moreover, the remainding fraction of 22% of points with no decision assigned to them, are such points which denote situations that are extremely rare from genetics point of view. Namely, these are the situations where the outcome of the Tajima test T is non significant and, at the same time, the outcome of the Kelly Z_{nS} test is significant or even strongly significant. Such a situation has never happened for any gene, for any population and for any of the null hypothesis, considered in the detailed multi-null study. Therefore, even if one cannot totally exclude such situations from theoretical point of view, in practice one meets them very rarely.

Algorithm 2

```
BALANCING_SELECTION := False; NO_DECISION := False;
If T >= ** or (T >= * and D* >= *) or ZnS >= * then
   BALANCING_SELECTION := True;
If T = NS or (T = * and D* = NS and ZnS = NS) then
   If BALANCING_SELECTION then
      NO_DECISION := True
   else
      BALANCING_SELECTION := False;
```

The comparison of Table 3 with Tables 5 and 6 shows the degree of generalization (into unknown combinations of the tests outcomes). It was increased by the application of rough set theory (Table 5) and by additional genetic knowledge (Table 6). Both these strategies, when applied together, resulted in a relatively

Table 6. The discrete space of three tests: D^*, T and Z_{nS}, forming a relative reduct. The domain of each test outcome (coordinate) is composed of three values: ** (strong statistical significance $p < 0.01$), * (statistical significance $0.01 < p < 0.05$), and NS (no significance $p > 0.05$). The given point in a space is assigned to S, N or "-" (with the meaning identical to that given in captions of Tables 3 and **??**). If any character is in parentheses, it means, that the point is assigned to the given value not automatically. Rather, the simple reasoning is used. It states that the selection is even more probable for given test showing strong significance (**), when automatic knowledge acquisition indicated such selection for this test being just significant (*) with the values of other tests unchanged. The assignment in Table 6 is done basing on the Decision Algorithm 2, which, similarly to Algorithm 1, uses the relative value reducts varying among different training examples. Note that the fraction of points covered by the algorithm is now 100%, but 22% denotes the contradiction in the decision, and such a case should be treated as the lack of decision. Therefore, the fraction of points really assigned with the decision is now 78%.

		T								
		**			**			NS		
		Z_{nS}			Z_{nS}			Z_{nS}		
		**	*	NS	**	*	NS	**	*	NS
	**	S	S	S	(S)	(S)	(S)	(-)	-	N
D^*	*	S	S	S	S	S	S	(-)	-	N
	NS	S	S	S	(S)	S	N	(-)	-	N

high increase of covering of the space generated by test outcomes (from 30% covered by the training examples, to 78% covered by the Algorithm 2).

However, here the question could be raised, what is the probability of correct generalization into unknown situations. To study this problem, there was applied automatic knowledge extraction procedure presented above, in the so-called jack-knife cross-validation, which is known to be unbiased in estimating the decision error of any classifier. Classical jack-knife strategy assumes that the training is performed basing on all-but-one training examples, and that the testing is done for the excluded example. After iterating this procedure N times (where N is the number of training facts), the average of decision errors in separate iterations is an unbiased estimate of the decision error. However, in case considered such a strategy could give too optimistic results, because training facts describing one gene in four different populations are not independent, and even after excluding one of them some knowledge about it is passed to the classifier. That is why, to be rigorous about the conclusions, the author decided to exclude from the iterations all four examples concerning one particular gene, and perform training basing on examples concerning three remaining genes.

The detailed presentation of results of cross-validation is beyond the scope of this paper. Here, the author would only like to point out that relatively large decrease of the number of training examples, which was the result of the assumed strategy, could produce pessimistic estimates of the decision error. However, it proved that even such pessimistic estimate as can be seen in Table 7, is small enough (12.5% with a variation between iterations equal to 0.0156) to claim that the proposed methodology could be utilized as useful tool in looking for candidates for more detailed analysis with computationally more requiring strategy, like the multi-null methodology. The last statement is based on the fact that as much as 87.5% correct recognitions of balancing selection for unknown genes were done when the proposed rough set based methodology was applied, with completely no need for performing long-lasting computer simulations for calculation of critical values of tests under non-classical null hypotheses (as required by multi-null methodology). The results of cross-validation procedure are also summarized in a form of confusion matrix in Table 8.

Table 7. The results of the cross-validation in a modified jack-knife strategy

Iteration	Errors in populations				Percentage of	Decision
without gene	African-American	Caucasian	Asian	Hispanic	correct decisions	Error
ATM	*Y*	*N*	*N*	*N*	75%	25%
RECQL	*N*	*N*	*N*	*N*	100%	0%
WRN	*N*	*N*	*N*	*N*	100%	0%
BLM	*N*	*Y*	*N*	*N*	75%	25%
Average:					**87.5%**	**12.5%**

Table 8. The confussion matrix of the cross-validation test

		Prediction	
		Lack of balancing selection	Balancing selection
Actual value	Lack of balancing selection	7	1
	Balancing selection	1	7

4 Conclusion

Since the time of Kimura's book [1] the search for the signatures of natural selection at molecular level has become one of important directions in genetics. However, many non-neutrality tests generate similar patterns for such depatures from neutral model like population growth or substructure of population. Since these factors influence different tests in a different way, the battery of tests can be more informative than any separate one. The problem of interpretation of a battery of such tests was considered in a paper. It proved that the rough set based decision making system can correctly (i.e with the concordance with time consuming mulit-null methodology) recognize 87.5% of cases of balancing selection for genes not used in a training.

Acknowledgments. The author would like to acknowledge the financial support of this work in the genetics related part under the MNiSW grant number

3T11F 010 29, and in the rough set related part under author's activities BW2007 at SUT. The author would like also to thank to Prof. M. Kimmel from Rice University in Houston for long discussions and advise concerning the multi-null methodology using statistical non-neutrality tests.

References

1. Kimura, M.: The Neutral Theory of Molecular Evolution. Cambridge University Press, Cambridge (1983)
2. Zhang, J.: Evolution of the Human ASPM Gene, a Major Determinant of Brain Size. Genetics 165, 2063–2070 (2003)
3. Evans, P.D., Anderson, J.R., Vallender, E.J., Gilbert, S.L., Malcom, Ch.M. et al.: Adaptive Evolution of ASPM, a Major Determinant of Cerebral Cortical Size in Humans. Human Molecular Genetics 13, 489–494 (2004)
4. Fu, Y.X., Li, W.H.: Statistical Tests of Neutrality of Mutations. Genetics 133, 693–709 (1993)
5. Fu, Y.X.: Statistical Tests of Neutrality of Mutations Against Population Growth, Hitchhiking and Background Selection. Genetics 147, 915–925 (1997)
6. Kelly, J.K.: A Test of Neutrality Based on Interlocus Associations. Genetics 146, 1197–1206 (1997)
7. Wall, J.D.: Recombination and the Power of Statistical Tests of Neutrality. Genet. Res. 74, 65–79 (1999)
8. Nielsen, R.: Statistical Tests of Selective Neutrality in the Age of Genomics. Heredity 86, 641–647 (2001)
9. Cyran, K.A., Polaska, J., Kimmel, M.: Testing for Signatures of Natural Selection at Molecular Genes Level. J. Med. Inf. Techn. 8, 31–39 (2004)
10. Dhillon, K.K., Sidorova, J., Saintigny, Y., Poot, M., Gollahon, K., Rabinovitch, P.S., Mon-nat Jr., R.J.: Functional Role of the Werner Syndrome RecQ Helicase in Human Fibroblasts. Aging Cell 6, 53–61 (2007)
11. Karmakar, P., Seki, M., Kanamori, M., Hashiguchi, K., Ohtsuki, M., Murata, E., Inoue, E., Tada, S., Lan, L., Yasui, A., Enomoto, T.: BLM is an Early Responder to DNA Double-strand Breaks. Biochem. Biophys. Res. Commun. 348, 62–69 (2006)
12. Golding, S.E., Rosenberg, E., Neill, S., Dent, P., Povirk, L.F., Valerie, K.: Extracellular Signal-Related Kinase Positively Regulates Ataxia Telangiectasia Mutated, Homologous Recombination Repair, and the DNA Damage Response. Cancer Res. 67, 1046–1053 (2007)
13. Schneider, J., Philipp, M., Yamini, P., Dork, T., Woitowitz, H.J.: ATM Gene Mutations in Former Uranium Miners of SDAG Wismut: a Pilot Study. Oncol. Rep. 17, 477–482 (2007)
14. Polanska, J.: The EM Algorithm and its Implementation for the Estimation of the Frequencies of SNP-Haplotypes. Int. J. Appl. Math. Comp. Sci. 13, 419–429 (2003)

Indiscernibility Relation for Continuous Attributes: Application in Image Recognition

Krzysztof Cyran and Urszula Stanczyk

Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
{krzysztof.cyran,urszula.stanczyk}@polsl.pl

Abstract. The paper presents an application of rough sets in a problem defined for the continuous feature space used by hybrid, high speed, pattern recognition system. The feature extraction part of this system is built as a holographic ring-wedge detector based on binary grating. Such feature extractor can be optimized and we apply for this purpose automatic knowledge acquisition and processing. Features from optimized extractor are then classified with the use of probabilistic neural network classifier. The methodology, proposed by one of the authors in earlier works, has been further enhanced here by application of modified indiscernibility relation. Modified version of this relation makes possible natural application of discrete type rough knowledge representation to problems defined in continuous space. We present an application of modified indiscernibility relation in the domain of image recognition.

Keywords: indiscernibility relation, ring-wedge detectors, probabilistic neural networks, image recognition, evolutionary optimization.

1 Introduction

The pattern recognition problems are of great importance among applications of machine learning. Therefore, many researchers and companies have focused their attention on trying to build system which is reliable, fast and easily adaptable. Hybrid solutions, designed to perform heavy computations in optical mode with practically no time delays, and to post-process the optical results in classical computers, are considered in the paper as the target for rough based optimization of the feature extractor. We present enhancements in the system capable of size, rotation and shift invariant recognition of the input images.

As the feature extractor of the system a holographic ring-wedge detector (HRWD) has been proposed [1]. The first complete recognition system was composed of a commercially available ring wedge-detector (RWD) and a neural network [2]. The system, however, was lacking the possibility of adaptation because of application of RWD instead of HRWD fitted specifically to given application. According to optical characteristics the HRWD is a grating-based diffractive optical variable device (DOVD) [3]. The first propositions of the methodology suitable for optimization of HRWD considered the choice of the objective function. The analysis of this problem led Cyran, Mrozek and Jaroszewicz [4,5] to

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 726–735, 2007.

choose as an objective the quality of approximation of the decision attribute by conditional attributes. This notion is defined within the theory of rough sets, originated by Pawlak [6]. The subsequent development of the theory was due to work of such researches like (among others) Mrozek [7], Skowron and Grzymala-Busse [8], Ziarko who proposed the variable precision model [9], or Pawlak and Skowron [10,11,12].

The optimization of HRWD was successfully applied to ANN-based system for recognition of the type of subsurface stress in materials with embedded optical fiber [13,14,15] or in systems designed by Podeszwa, Jaroszewicz et al. [16,17] for monitoring airplane engines wear. The purely optical version of considered here recognition system was also studied [18] but its practical applicability is limited by the slow development of technology of optical neural networks. The serious problems with obtaining non linear activation function in an optically implemented artificial neuron are the main reason for such a situation.

Remarkably, neural network is not the only type of classifiers that could be used in classification of features generated by HRWD. Moreover, the first version of optimization procedure was better designed for the rough set based classifiers. The fact is due to identical, discrete nature of knowledge representation in the theory of rough sets applied both in HRWD optimization and in subsequent rough set based classification. The general ideas concerning the design of such a rough classifier, as well as the project of the fast rough classifier implemented as a programming logic device (PLD) can be found in [19,20].

However, systems obtained in these early works were suboptimal. This suboptimality was a simple consequence of the fact, that the feature space generated optically by the HRWD is always a continuous space. Therefore, a separate discretization of each conditional attribute, required by rough set based machine learning methods, introduced highly nonlinear transformation and potentially lost some useful information.

Natural enhancement could be possible if both, classifier and optimization procedure worked in a continuous space. The first can be easily achieved by the application of a probabilistic neural network classifier, but the latter demands a modification of the indiscernibility relation in rough set theory.

We propose here to use one of possible generalizations of the indiscernibility relation, allowing for natural processing of real valued attributes. Such a modification improves the results of evolutionary optimization of HRWD and makes possible to avoid highly non linear transformation of separate features corresponding to conditional attributes in the theory of rough sets.

The paper explains the mutual relationship between classical and modified indiscernibility relations in the section 2. In the section 3 there is presented the high speed HRWD-ANN based pattern recognition system, utilizing optimization with modified indiscernibility relation. This section starts with foundations of the system considered and continues with description of experimental results, comparing the enhanced methodology with that published before. The discussion and final conclusions are presented in the section 4.

2 Modified Indiscernibility Relation

In our previous work we proposed to use as the objective function the approximation of the decision attribute by conditional attributes. It is equivalent to the statement, that the consistency measure of a decision table has been used, as these two notions are different expressions of the same concept. Such a objective was proposed to be used in evolutionary optimization of HRWD. The motivations supporting such a criterion have been considered in [4]. Even if they seem to be reasonable, both from theoretical and experimental perspective, yet assuming classical definition of indiscernibility relation, they always resulted in suboptimal solutions.

Taking above into consideration, we decided to use such an indiscernibility relation that defines two objects as indiscernible if they belong to the same clusters in a continuous space (or subspaces). It makes possible to avoid the need of independent discretization of individual features when calculating the rough set based objective function. The consistency measure of decision table, used as a criterion in optimization for classification, can be easily computed based on modified indiscernibility relation, satisfying the demand that this relation should be an equivalence relation, i.e. it should be reflexive, symmetric and transitive. More general indiscernibility relations have been also proposed, however, for classification the equivalence relation seems to be the most natural one.

Formally, let us consider the information system $S =< U, Q, v, f >$ composed of universe U, set of atributes Q, information function f, and a mapping v which associates each attribute $q \in Q$ with its domainV_q. Let the process of discretization be denoted as a vector function Λ: $\Re^{card(C)} \rightarrow \{1, 2, \ldots, \xi\}$,where ξ denotes the number of clusters covering the domain of attributes $q \in C$. Furthermore, let discretization of any individual attribute $q \in C$ be denoted as a scalar function Λ: $\Re \rightarrow \{1, 2, \ldots, \xi\}$. Then, the classical indiscernibility relation is defined as:

$$x \; IND_0 \, (\Lambda[C]) \, y \Leftrightarrow \forall q \in C, \;\; f(x, \Lambda[q]) = f(y, \Lambda[q]) \qquad (1)$$

To introduce formally the modification that we propose to use in our pattern recognizer, let us change the notation of indiscernibility relation from classical form (1) dependent on unstructured set of attributes, to a version being dependent on a family (set) of sets of attributes. Such modification makes it possible to introduce hierarchy into originally unstructured set of attributes. If we denote $\mathbf{C} = \{C_1, C_2, \ldots, C_N\}$ as a family of disjoint sets of attributes $C_n \subseteq Q$ then the originally unstructured set of attributes $C \subseteq Q$ is equal to the union of elements of the family $\mathbf{C}$, i.e. $C = \bigcup_{C_n \in \mathbf{C}}, C_n$. Furthermore, let the indiscernibility relation be dependent on family $\mathbf{C}$ instead of being dependent on a set C. Note, that both sets $\mathbf{C}$ and C contain the same collection of single attributes, however $\mathbf{C}$ includes additional, second order structure, and C is a normal set of attributes. If this structure proves to be irrelevant for the problem considered, it can be simply ignored and we can go back to the classical version of indiscernibility relation IND_0. Let the modified indiscernibility relation $IND_1(\mathbf{C}) \in U \times U$ be defined as

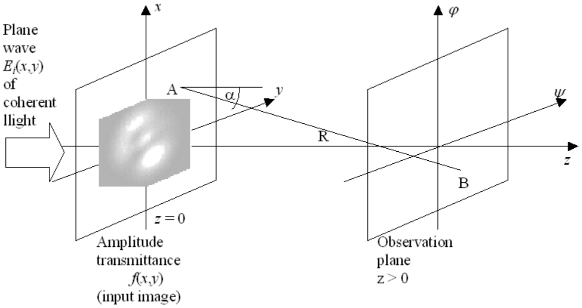
$$x\ IND_1\,(\mathbf{C})\,y \Leftrightarrow \forall C_n \in \mathbf{C},\ Cluster(x, C_n) = Cluster(y, C_n) \qquad (2)$$

where $x, y \in U$, and $Cluster(x, C_n)$ denotes the cluster, that the element x belongs to. There are two opposite cases influencing the exact meaning of this relation. The first is obtained when the family $\mathbf{C}$ is composed of exactly one set of all conditional attributes C, and the second is when the family $\mathbf{C}$ is composed of $N = card(C)$ one-element sets containing different conditional attributes $q \in C$. The classical form IND_0 of the indiscernibility relation is obviously the latter special case of modified version IND_1, which can be denoted as $IND_0(\Lambda[C]) \equiv IND_1(\mathbf{C}) \Leftrightarrow \mathbf{C} = \left\{\{q_n\}\colon C = \bigcup_{q_n \in C}, \{q_n\}\right\} \wedge Clus(x, \{q\}) = f(x, \Lambda[q])$. In such a case the clustering and discretization is performed separately for each continuous attribute.

3 Application into Image Recognition System

As it was already stated, the feature extraction part of the system considered is performed by optical methods. The information required for understanding of Fraunhofer diffraction pattern sampling performed by HRWD, is presented in this paper, as briefly as possible.

Let us assume that some amplitude transmittance denoted by $f(x, y)$ represents the transparent image to be recognized. Let it be placed in a plane of the Cartesian coordinate system, perpendicular to axis z and such that coordinate $z = 0$. This situation is represented by the left part of the Fig. 1.



x
φ
Plane wave $E_i(x,y)$ of coherent llight
A
α
y
ψ
R
z
B
z = 0
Amplitude transmittance f(x,y) (input image)
Observation plane z > 0

Fig. 1. The input plane with the transparent image $f(x, y)$ and the observation plane with observation point **B**, are parallel to each other and perpendicular to the optical axis represented by the axis z of Cartesian coordinate system. Modified basing on [20].

The input image is illuminated by a plane wave $E_p(x, y) = E_0 exp(ikr)$ of coherent light. In the above equation $k = [k_x, k_y, k_z]$ and $r = [r_x, r_y, r_z]$ are are wave and spatial vectors, respectively. Note, that this is the situation of illuminating the hologram with amplitude transmittance f by a reference beam (plane wave). The field directly behind the considered amplitude transmittance is given by the equation $E(x, y) = f(x, y)E_p(x, y)$. At observation plane, that is for points with coordinate $z > 0$ the integration over all spherical waves emitted from points $(x, y, 0)$ results in Fresnel-Kirchhoff integral, given by the equation [14,20,21]:

$$E(\psi, \varphi, z) = \frac{iE_0}{\lambda} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \frac{e^{-ikR}}{R} \left(\frac{1}{2} + \frac{1}{2} \cos \alpha \right) dxdy \qquad (3)$$

In (3) R and α denote radius and angle respectively. They extend from the point, say $\mathbf{A}$, of emission of light with length λ to the point, say $\mathbf{B}$, of observation. For points with coordinate $z \to \infty$, i.e. when $z \gg \pi(x^2 + y^2)/\lambda$, so called Fraunhofer approximation of the integral is (3) given by [14,20,21]:

$$E(\psi, \varphi, z) = \frac{iE_0}{\lambda z} e^{-\frac{i\pi}{\lambda z}(\psi^2+\varphi^2)} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{\frac{i2\pi}{\lambda z}(x\psi+y\varphi)} dxdy \qquad (4)$$

Given in Fig. 1 basic optical setup generating Fraunhofer approximation of Fresnel-Kirchhoff integral is of no practical use, since this approximation is observable for really great distances between input and observation planes (to satisfy the assumption that $z \to \infty$). Therefore, additional element, namely the spherical lens is very often applied. Such lens generates a phase delay $\Delta\emptyset(x, y)$, and thus it can be treated as a transparency with a complex amplitude transmittance given by [14,20,21]:

$$t_L(x, y) = e^{\frac{i\pi}{\lambda f}(x^2+y^2)} \qquad (5)$$

For a setup composed of the input image $f(x, y)$ and the lens $t_L(x, y)$, the Fresnel approximation becomes a Fraunhofer approximation in a back focal plane ($z = f$) of the lens. In other words, Fraunhofer approximation of the Fresnel-Kirchhoff integral is brought by the spherical lens from the infinity to its back focal plane, which is expressed by [14,20,21]:

$$E(\psi, \varphi, z = f) = \frac{iE_0}{\lambda f} e^{-\frac{i\pi}{\lambda f}(\psi^2+\varphi^2)} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{\frac{i2\pi}{\lambda f}(x\psi+y\varphi)} dxdy \qquad (6)$$

One can observe that equation (6) is a Fourier transform of the input image $f(x, y)$ up to the multiplication of the result with a spherical phase factor. However, since only the intensity can be directly observed or recorded, therefore in practical applications the spherical phase factor can be omitted. The resulting intensity, which is a Fourier power spectrum $F^2(u, v)$ of the input transmittance $f(x, y)$, is called Fraunhofer diffraction pattern and the back focal plane of the lens is often referred to as a Fourier plane.

After presenting the processing of the Fourier transform by the spherical lens, let us consider the operation of the holographic ring-wedge detector. It is a circular element composed of rings and wedges, covered with rectangular diffraction gratings of different spatial frequency and orientation [1]. When HRWD is placed in a Fourier plane, then it samples the Fraunhofer diffraction pattern by integrating the power spectrum over rings and wedges. In this way, each region of HRWD generates exactly one real-valued feature, and the value of this feature is equal to the integral of the power spectrum integrated over that region.

Assuming that the Fraunhofer diffraction pattern is expressed in polar coordinates (ρ, θ), features generated by rings R_i and features generated by wedges W_j have the values [14,20]:

$$R_i = \int\limits_0^{\pi} \int\limits_{\rho_i}^{\rho_{i+1}} F^2(\rho, \theta) d\rho \, d\theta, \; W_j = \int\limits_0^{R} \int\limits_{\theta_j}^{\theta_{j+1}} F^2(\rho, \theta) d\rho \, d\theta \tag{7}$$

In above equation R is the radius of the HRWD element, ρ_i $(i = 1, \ldots, N_R)$ are radii of rings and θ_j $(j = 1, \ldots, N_W)$ are angles of wedges. The numbers N_R and N_W represent total numbers of rings and wedges respectively. Taking into consideration well-known properties of Fourier transform and specific shapes of HRWD regions it is clear, that the entire feature vector contains elements shift and rotation invariant, but scale dependent (generated by rings), and shift and scale invariant, but rotation dependent (generated by wedges). The answer to the question: which invariances and which dependencies in a feature vector are informative for the image recognition, is a task dependent problem, which can be automatically resolved by a properly trained classifier.

When all classes can be represented by single clusters, the choice of the objective function in optimization of feature extraction is simple. In such a case the distance between clusters can be used as an objective function. In more complex problems, however, one class can be represented by points, which not necessarily form single cluster in a feature space. These are so called multimodal distributions of data, and certainly they are more general as compared to single-cluster per class distribution. However, even in the case of multimodal distribution of data the distance between one class clusters belonging to different classes can be used as an auxiliary criterion in bi-objective optimization. The definition of the good main objective function is not a trivial task in such a situation.

In our previous work [4,5] we proposed the objective function as a coefficient defined in the theory of rough sets. More specifically it was the consistency measure $\gamma_C(D^*)$ of the decision table with conditional attributes corresponding to rings and wedges of HRWD, whereas the decision attribute was the class of the image. In this study, we used this coefficient as a main criterion in the evolutionary optimization.

As it was already stated, such an objective, when maximized, leads to the feature space optimal for the rough classifiers. However, for ANN-based classifiers the feature space is suboptimal. The reason for this lies in the requirement of discretization light intensities independently of each other, as demanded by classical definition of indiscernibility relation in rough set theory. Now, we propose to use the same criterion, however, computed for modified version of indiscernibility relation defined by formula (2), and additionally to use auxiliary criterion to separate one-class clusters in a feature space.

Defined above enhanced objective function is not differentiable and therefore, gradient-based search method should be excluded. However, the HRWD can be optimized in a framework of an evolutionary algorithm. As genetic operations classical one-point recombination and uniform mutation have been used. The selection was proportional, however it was used in the elitist model propagating best solution from generation to generation with probability 1. The scalarization of objectives was done basing on the weighted average for the purpose of proportional selection, and basing on the lexicographic order for the purpose of elitist model.

4 Results and Conclusions

We verified the recognition abilities of presented system using PNN for classification. In order to show that presented methodology used for optimization of HRWD is general and does not make any specific assumptions about the type of images, we considered three different image recognition problems: 1) the recognition of speckle structures from the output of the optical fiber, 2) the recognition of the type of the vehicle, and 3) the recognition of the type of road obstacle based on infrared images.

In the first problem the experiments were conducted for a set of 128 images of speckle patterns generated by intermodal interference occurring in optical fiber and belonging to eight classes. We obtained the Fraunhofer diffraction patterns of input images by calculating the intensity patterns from discrete Fourier transform equivalent to (6). To compute the feature vectors we applied discrete forms of equations (7).

To cross-validate the quality of recognition we applied modified jack-knife methodology. In classical jack-knife strategy all-but-one examples are used for a training, and the recognition of the example excluded from the training data is iterated for all examples. However, in each iteration we excluded 8 images, each belonging to different class. In this way we achieved 8-fold increase in the computational time, required to cross-validate the results. The time course of evolutionary optimization is given in the Fig. 2.

In the recognition of the type of vehicles we used a set of 128 images belonging to 4 different classes: buses, trucks, vans and cars. However, in this experiment each vehicle was represented by a set composed of 4 images shifted slightly, to reflect the different conditions of exposure. Since the images of the same vehicle are positively correlated therefore we always excluded from the training set all 4 images of the same car. To perform time-efficient cross-validation we excluded in all iterations images of four vehicles belonging to different classes (that is we excluded in fact 16 images). Therefore, we had to iterate such a procedure 8 times. The optimization of the HRWD in this experiment is presented in Fig. 3.

The images in the experiment of recognition of the type of the road obstacle in the infrared light belonged to two classes: living creatures and others. We

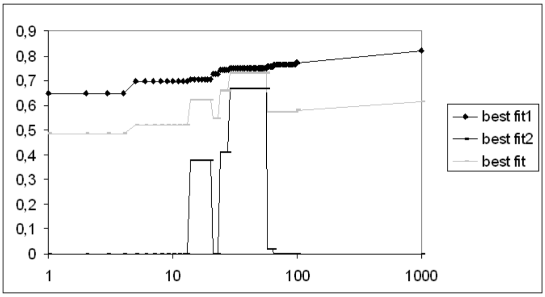


Fig. 2. Experiment with the recognition of speckle structures. Evolutionary optimization of HRWD as a time course of the objective function for the best individual in a population. Curve *bestfit*1 indicates the course of the main rough set based criterion, curve *bestfit*2 represents the course of the additional distance based criterion, and curve denoted *bestfit*3 is a weighted average of these criterions. Plot is presented in a logarithm scale of time.

used a set of 64 such images. In the cross-validation, in each of 32 iterations two images belonging to different classes were excluded. The optimization in this experiment is presented in a Fig. 4.

Observe that even if the behavior of the second, distance based, criterion denoted by *bestfit*2 is different among Fig. 2, 3, and 4, the rough set based criterion qualitatively looks similar. In all these courses, it follows a monotonic growth, approximated by a straight line in the plot with log-time scale. Therefore, in a linear time scale, the increase of this criterion can be approximated by a logarithmic curve, reflecting the fact, that the more optimized the current solution, the more difficult it is to optimize it further.

The cross-validation procedures described above indicated that the normalized decision errors in these experiments were ranging from 14 to 5 percent suggesting over-all relatively good recognition abilities of the system considered, especially when it is taken into mind that the system is not dedicated for recognition of specific types of images.

The theory of rough sets (and some generalizations of it, like variable precision model) has been successfully applied in many machine learning problems. However, it is well known drawback of classical versions of these theories, that they deal with continuous attributes in an unnatural way. To overcome this disadvantage we applied indiscernibility relation based on structural family of subsets of conditional attributes ([2]), which is equivalently valid for classical theory of rough sets as well as for the variable precision model, most often used in machine learning concerning large data sets.

Such modified indiscernibility relation introduces the flexibility in applying a particular case of it to the given application. In the case of continuous attributes it allows for multidimensional cluster analysis, as opposed to one-dimensional analyses required by classical form of indiscernibility relation. At the same time,

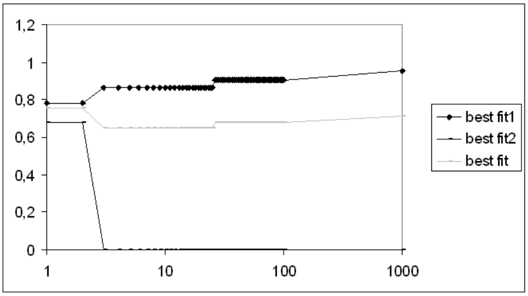


Fig. 3. Experiment with the recognition of the type of vehicles. Further explanations like in capture of Fig. 2.

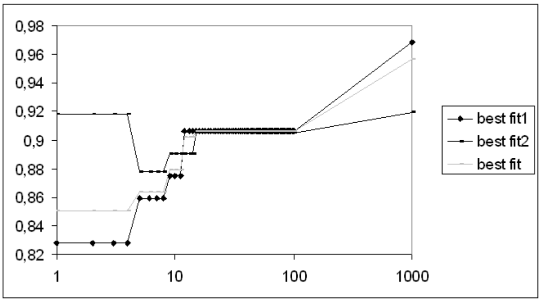


Fig. 4. Experiment with the recognition of the road obstacle in infrared light. Further explanations like in capture of Fig. 2.

the modified version remains the equivalence relation, which seems to be a natural choice in a classification problems.

When using modified relation IND_1, in majority of cases the cluster analysis should be performed in a space generated by all attributes. This corresponds to a family **C**, which consists of only one set composed of all conditional attributes. However, in this experimental study we used a family $\mathbf{C} = \{C_R, C_W\}$ composed of two sets containing 8 elements each ($card(C) = 16$, $card(C_R) = card(C_W) = 8$, $card(\mathbf{C}) = \mathbf{2}$). Such a structure is suggested by the architecture and properties of the HRWD that we used as a feature extractor.

References

1. Casasent, D., Song, J.: A Computer Generated Hologram for Diffraction-Pattern Sampling. Proc. SPIE 523, 227–236 (1985)
2. George, N., Wang, S.: Neural Networks Applied to Diffraction-Pattern Sampling. Appl. Opt. 33, 3127–3134 (1994)
3. Cyran, K.A., Niedziela, T., Jaroszewicz, L.R.: Grating-based DOVDs in High Speed Semantic Pattern Recognition. Holography 12, 10–12 (2001)

4. Cyran, K.A., Mrozek, A.: Rough Sets in Hybrid Methods for Pattern Recognition. Int. J. Intel. Sys. 16, 149–168 (2001)
5. Jaroszewicz, L.R., Cyran, K.A., Podeszwa, T.: Optimized CGH-based Pattern Recognizer. Opt. Appl. 30, 317–333 (2000)
6. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Boston (1991)
7. Mrozek, A.: A New Method for Discovering Rules from Examples in Expert Systems. Man-Machine Studies 36, 127–143 (1992)
8. Skowron, A., Grzymala-Busse, J.W.: From Rough Set Theory to Evidence Theory. In: Yager, R.R., Ferdizzi, M., Kacprzyk, J. (eds.) Advances in Dempster Shafer Theory of Evidence, John Wiley & Sons, New York (1994)
9. Ziarko, W.: Variable Precision Rough Set Model. J. Comp. Sys. Sci. 40, 39–59 (1993)
10. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences. An International Journal, Elsevier 177, 3–27 (2007)
11. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences. An International Journal. Elsevier 177, 28–40 (2007)
12. Pawlak, Z., Skowron, A.: Rough sets and Boolean reasoning. Information Sciences. An International Journal. Elsevier 177, 41–73 (2007)
13. Cyran, K.A., Stanczyk, U., Jaroszewicz, L.R.: Subsurface Stress Monitoring System based on Holographic Ring-Wedge Detector and Neural Network. In: McNulty, G.J. (ed.) Quality, Reliability and Maintenance, pp. 65–68. Professional Engineering Publishing, Bury St Edmunds London (2002)
14. Cyran, K.A., Niedziela, T., Jaroszewicz, J.R., Podeszwa, T.: Neural Classifiers in Diffraction Image Processing. In: Proc. Int. Conf. Comp. Vision Graph, Zakopane Poland, pp. 223–228 (2002)
15. Cyran, K.A., Jaroszewicz, L.R., Niedziela, T.: Neural Network based Automatic Diffraction Pattern Recognition. Opto-elect. Rev. 9, 301–307 (2001)
16. Podeszwa, T., Jaroszewicz, L.R., Cyran, K.A.: Fiberscope based Engine Condition Monitoring System. Proc. SPIE. 5124, 299–303 (2003)
17. Jaroszewicz, L.R., Merta, I., Podeszwa, T., Cyran, K.A.: Airplane Engine Condition Monitoring System based on Artificial Neural Network. In: McNulty, G.J. (ed.) Quality, Reliability and Maintenance, pp. 179–182. Professional Engineering Publishing, Bury St Edmunds London (2002)
18. Cyran, K.A., Jaroszewicz, L.R.: Concurrent Signal Processing in Optimized Hybrid CGH-ANN System. Opt. Appl. 31, 681–689 (2001)
19. Cyran, K.A., Jaroszewicz, L.R.: Rough Set based Classifiction of Interferometric Images. In: Jacquot, P., Fournier, J.M. (eds.) Interferometry in Speckle Light. Theory and Applictions, pp. 413–420. Springer, Heidelberg (2000)
20. Cyran, K.A.: PLD-based Rough Classifier of Fraunhofer Diffraction Pattern. In: Proc. Int. Conf. Comp. Comm. Contr. Tech., Orlando, pp. 163–168 (2003)
21. Kreis, T.: Holographic Interferometry – Principles and Methods. Akademie Verlag Series in Optical Metrology, vol. 1. Akademie-Verlag, Heidelberg (1996)

Clustering of Leaf-Labelled Trees on Free Leafset*

Jakub Koperwas and Krzysztof Walczak

Institute of Computer Science, Warsaw University of Technology, Nowowiejska 15/19,
00-665 Warsaw, Poland
J.Koperwas@elka.pw.edu.pl, K.Walczak@ii.pw.edu.pl

Abstract. This paper focuses on the clustering of leaf-labelled trees on free leafset. It extends the previously proposed algorithms, designed for trees on the same leafset. The term z-equality is proposed and all the necessary consensus and distance notions are redefined with respect to z-equality. The clustering algorithms that focus on maximizing the quality measure for two representative trees are described, together with the measure itself. Finally, the promising results of experiments on tandem duplication trees are presented.

1 Introduction

This paper is a part of a larger work on applying data mining techniques to tree data - tree mining. Tree mining techniques have large applications in bioinformatics, image processing, text mining and others. This paper concentrates on clustering techniques for leaf-labelled trees, which have their main applications in the bioinformatics field. Previously in [1], we have presented techniques for clustering leaf-labelled trees, where all the trees where built on the same leafset. In this paper we enhance these methods so that they can be used for trees which do not contain exactly the same leafsets. We call them trees on a free leafset. In the first part of the paper we enhance the basic notions considering a tree representation, distance measure and consensus methods so that they are applicable to trees with free leafset. We introduce z-distance and z-consensus methods. The next section concentrates on the clustering of leaf-labelled trees with a free leafset. We show how to construct the algorithms for strict and majority rule consensus tree as a representative tree. We also discuss the quality measure used for assessing the clustering. Finally, we describe the results of clustering of tandem duplication trees, which are the leaf-labelled trees on a free leafset taken from bioinformatics field.

2 Basic Notions

2.1 Splits

One of the most popular leaf-labelled tree representations is the set of splits, which highlights the leaf-labelled trees interpretation as a space partition.

* The research has been partially supported by grant No 3 T11C 002 29 received from Polish Ministry of Education and Science.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 736–745, 2007.

Definition 1 (Split). *Split $A|B$ (of a tree T with leafset L), corresponding to an edge e is a pair of leafsets A and B, which originated by splitting tree T into two disconnected trees whilst removing an edge e from a tree T, $A \cup B = L$.*

If $|A|$ or $|B|$ is equal to 1, the split is trivial. Split $A|B$ is a valid split if both sets A and B are non-empty. The splits of tree T_1 from Fig. 1 are: $a|bcde, b|acde, c|abde, d|abce, e|abcd, abe|cd, be|acd$; among them $abe|cd, be|acd$ are non-trivial splits.

Definition 2 (Split Equality). *Two splits $A|B$ and $C|D$ are considered equal iff ($A = C$ and $B = D$) or ($A = D$ and $B = C$).*

The trees with free leafset cannot be compared easily if they are not built on the same leafset. In particular, the conventional distance or consensus techniques cannot be used, because splits, built on a different leafset cannot be equal. On the other hand, there is a need to compare such trees to determine whether they share common information or not. We present therefore, the restricted equality as an efficient and well-interpretable method of comparing two trees on free leafset.

Definition 3 (Restricted Split). *Split s_1 is a restricted version of split s_2 on the leafset z if it is built with removing leafs not in z from s_2: $s_2^z = s_1$.*

Split restriction is a complementary term to the term restricted tree described in [2]. It can be shown that the restricted tree of a tree T is built of restricted splits of a tree T on the same set z.

Definition 4 (Restricted Split Equality(z-equality)). *Splits s_1 and s_2 are restrictedly equal on the leafset z, if their restricted versions on the leafset z are equal: $s_1 =^z s_2 \iff s_1^z = s_2^z$.*

For example: $abc|def$ and $fabc|deg$ are restrictedly equal on the leafset $abcde$, because their corresponding restricted splits: $abc|de$ and $abc|de$ are equal, however they are not equal on a leafset $abcdef$ because their corresponding restricted splits: $abc|def$ and $fabc|de$ are not equal.

Definition 5 (Split Coherence). *Splits s_1 and s_2 are coherent if they are z-equal on the leafset z that is an intersection of their leafsets*
$s_1 \sim s_2 \iff s_1 =^z s_2 \wedge z = L(s_1) \cap L(s_2)$.

Z-equality/coherence relations as opposed to normal split equality relations do not determine whether two splits carry the same information but whether two splits do not contain contradictory information with respect to given leafset. For example $abc|def$ and $fabc|deg$ are not equal but they are restrictedly equal on the leafset $abcde$, which means that set of leaves $abcde$ in both splits is divided identically. Both the z-equality and the coherence are the equivalence relations.

2.2 Distance Between Leaf-Labelled Trees

One of the most popular distances for leaf labelled trees is a Robinson-Foulds distance. R-F distance between two trees T_1 and T_2 with set of splits S_1 and S_2 respectively is defined as follows:

$$d_{R-F}(T_1, T_2) = |S_1 \cup S_2| - |S_1 \cap S_2|. \tag{1}$$

For the reasons described earlier, classic R-F distance will not work if even one leaf is not present in both of the compared trees. Therefore, we extend the R-F distance with respect to leaf-labelled trees on free leafset.

Definition 6 (z-distance). *The z-distance for a given z is number of splits that are not z-equal on some leafset z.*

$$\begin{aligned} d_z(T_1, T_2) = |S_1 \div^z S_2| = |S_1 \cup^z S_2| - |S_1 \cap^z S_2|, \\ , where \\ S_1 \cup^z S_2 = \{s : (r \in S_1 \vee r \in S_2) \wedge (s = r^z)\}, \\ S_1 \cap^z S_2 = \{s : (r \in S_1 \wedge r \in S_2) \wedge (s = r^z)\}. \end{aligned} \tag{2}$$

Let us consider trees form Fig. 1 as an example. They are built on the following splits:
T_1 : $a|bcde, b|acde, c|abde, d|abce, e|abcd, abe|cd, be|acd.$
T_2 : $a|bcdef$, $b|acdef$, $c|abdef$, $d|abcef$, $e|abcdf$, $f|abcde$, $ab|cdef$, $ef|abcd$, $def|abcd$.
The z-distance, where $z = abcd$, is counted as follows:
The restricted splits are the following:
T_1 : $a|bcd, b|acd, c|abd, d|abc, ab|cd$. T_2 : $a|bcd, b|acd, c|abd, d|abc, ab|cd$.
Therefore the z-distance on set $abcd$ equals 0.
Z-distance on set $abcde$ is equal to 4 the same as for set $abcdexy$.
It may seem more natural to count the distance for two trees where z contains common leaves of compared trees i.e. with respect to coherence relation rather than z-equality. However, the distance defined in this way could not meet triangle inequality, therefore it is not a metrics. There are more possible ways to define the distance between leaf labelled trees on free leafset. However the z-distance is both efficient and has a good interpretation. The value of z-distance for two trees indicates the amount of contradictory information in those trees, with respect to a given leafset. For an interpretation in phylogenetic analysis we may imagine that we have two species trees that share common taxa a, b, c, d among others, that are not shared. Counting z-distance on $abcd$, we want to check how much the information about relations of these particular taxa differ in given trees. Z-distance is a natural extension of R-F distance, because for trees with the same leafset it will give the same result.

2.3 Consensus Methods Extensions for Free Leafset

Consensus methods in phylogenetic analysis are used to extract common information from set of trees and represent it as a single tree. The most popular are a

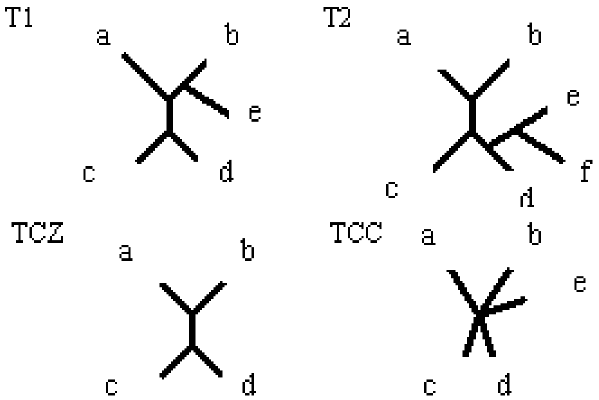


Fig. 1. Two leaf-labelled trees on free leafset and their z-restricted on a leafset *abcd* and common-restricted strict consensus trees

strict consensus tree and a majority rule consensus tree. Strict consensus tree is built of splits that occur in all of the input trees. Majority-rule consensus tree is built of splits that occur in the majority of the input trees. Consensus methods used for trees with free leafset will result in empty consensus tree split-set (always for strict and often for a majority-rule). Therefore we extend these terms with respect to restricted splits.

Definition 7 (z-restricted Strict Consensus Tree). *For a profile of trees* $T_1, \ldots, T_n$ *z-restricted strict consensus tree is built of valid splits s such that s is restrictedly equal on z to at least one split in each tree, in other words, split s is a restricted version of at least one split in each tree on leafset z.*

$$T_{zc}(T_1, \ldots, T_m) : S_{zc} = \left(\bigcap_{i=1}^{m} \right)^{z} S_i. \tag{3}$$

Definition 8 (Common-restricted Strict Consensus Tree). *Common-restricted consensus tree is a z-restricted consensus tree where z is an intersection of all corresponding leafsets* $L_1, \ldots, L_n$.

In order to construct z-restricted or common-restricted tree, we restrict all splits to a leafset z, and count classic consensus tree. For trees from the Fig. 1, the z-restricted strict consensus tree on a leafset *abcd* contains $a|bcd, b|acd, c|abd, d|abc$, $ab|cd$ (see Fig. 1 - TCZ) and the common-restricted strict consensus tree consists of: $a|bcde, b|acde, c|abd, d|abceande|abcd$ (see Fig. 1 - TCC)

Property 1. For any given set of trees $T_1, \ldots, T_m$ and a set z.

$$T_{zc}(T_1, \ldots, T_m) = T_{zc}(T_1, T_{zc}(T_2, \ldots, T_m)), \tag{4}$$

where T_{zc} is z-restricted strict consensus tree on leafset z.

Proof.

$$\begin{array}{r} T_{zc}(T_1,\ldots,T_m) : S_{zc} = (\bigcap_{i=1}^{m})^z S_i = \bigcap_{i=1}^{m} S_i^z, \\ T_{zc}(T_1, T_{zc}(T_2,\ldots,T_m)) : S_{zc} = (\bigcap_{i=2}^{m} S_i^z) \cap S_1^z = \bigcap_{i=1}^{m} S_i^z. \end{array} \quad (5)$$

Definition 9 (z-restricted Majority-rule Consensus Tree). *For a profile of trees $T_1,\ldots,T_n$, z-restricted majority-rule consensus tree is built of valid splits s such that s is restrictedly equal on z to some split, from the majority of trees.*

Definition 10 (Common-restricted Majority-rule Consensus Tree). *Common-restricted majority-rule consensus tree is a z-restricted consensus tree, where z is an intersection of all corresponding leafsets $L_1 \ldots L_n$ of the whole profile.*

In the Fig. 2 and Fig. 3 there are examples on z-restricted on abcdef and common-restricted majority rule consensus trees.

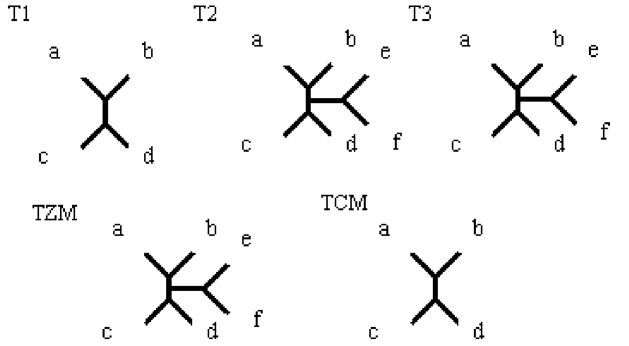


Fig. 2. Profile of trees together with their z-restricted on abcdef and common-restricted majority- rule consensus trees

The examples from Fig. 2 and Fig. 3 show that choosing a set z is not obvious. If an intersection of leaves is used, sometimes the tree may lose some interesting information, like in example from Fig. 2, however taking a larger leafset may bring totally uninformative tree like in example from Fig. 3. Finding most informative z-restricted majority-rule consensus tree is another interesting task for future considerations.

Property 2. For any given set of trees z-restricted majority-rule consensus tree is a middle tree with respect to z-distance i.e. it minimizes the sum of z-distances between itself and all the trees. (Theorem 1 is a proof of this property)

Lemma 1. *For any set of trees $T_1,\ldots,T_m$ on the same leafset if T_M is a majority-rule consensus tree then*

$$T_M : \min \sum_{i=1}^{m} d(T_i, T_M) \quad (6)$$

(this was proved by [3]).

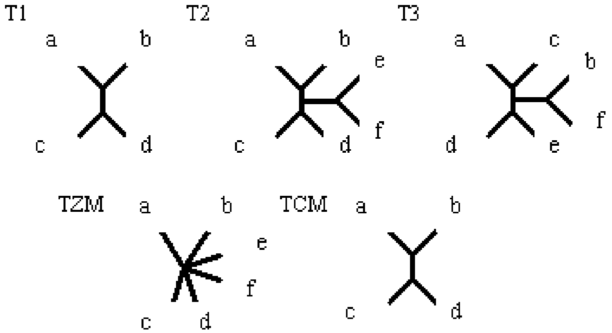


Fig. 3. Profile of trees together with their z-restricted on abcdef and common-restricted majority- rule consensus trees

Theorem 1.

$$T_{Mz} : \min \sum_{i=1}^{m} d^z(T_i, T_{Mz}). \tag{7}$$

Proof.

$$\begin{aligned} T_M : \min \textstyle\sum_{i=1}^m d(T_i, T_M) \Rightarrow T_{Mz}^z : \min \textstyle\sum_{i=1}^m d(T_i^z, T_{Mz}^z) \\ \Rightarrow T_{Mz}^z : \min \textstyle\sum_{i=1}^m d^z(T_i, T_{Mz}), \\ \text{because } d^z(T_i, T_{Mz}) = d(T_i^z, T_{Mz}^z) \\ \Rightarrow T_{Mz} : \min \textstyle\sum_{i=1}^m d^z(T_i, T_{Mz}), \\ \text{because } T_{Mz} = T_{Mz}^z. \end{aligned} \tag{8}$$

Consensus methods presented above are suitable for representing common information in leaf-labelled trees on free leafset.

3 Clustering of Leaf-Labelled Trees on Free Leaf-Set

The aim of our clustering techniques is to divide trees into k groups in such a way the clustering is possibly the best towards our quality measure.

3.1 Quality Measure

The quality measure is based on the informativity of the representative trees of each cluster. The representative may be any predefined tree that shares common knowledge of all the trees, it can possibly be strict consensus tree, majority-rule consensus tree or other. The representative tree shall only contain the knowledge present in input trees but nothing more. We can state that $S_R \subseteq \bigcup_{i \in C} S_i^z$, which is again the free-leafset extension of what was proposed in [1]. Here we focus on z-restricted (also common-restricted as a special case) strict consensus tree and majority-rule consensus tree, because these trees can be efficiently counted with simple algorithms. The quality is counted as follows:

1. select the k representative trees, one for each cluster
2. count how much information is lost when replacing the whole dataset of trees with k representative trees - this is information loss
3. count how much information is lost when replacing the whole dataset with a single representative tree - this is one-cluster information loss
4. count information gain as follows:

$$IG = \frac{\Delta I_{C_0} - \Delta I}{\Delta I_{C_0}}, \tag{9}$$

which shows how much our clustering is better from no clustering.

The informativity of a tree is simply the amount of non-trivial splits contained by tree [4], therefore information loss for a cluster is counted with formula:

$$\Delta I_{C_x} = \sum_{i=1}^{l} |S_i \div^z S_R|. \tag{10}$$

For further information on informativity and information gain please refer to [1].

3.2 Clustering of Leaf-Labelled Trees with Free Leafset with a z-Restricted Strict Consensus Tree as a Representative Tree

The aim of this clustering is to divide trees into k groups in such a way that information gain towards the z-restricted strict consensus tree is maximal. For this purpose we choose an agglomerative clustering algorithm, but we replace common merging strategies min, max and complete linkage with our own: minimum information loss linkage (agg-inf). We choose such two clusters to merge that merging minimizes the information loss of the clustering after the merging.

$$\arg\min_{C_x, C_y} \Delta I' - \Delta I. \tag{11}$$

This way it automatically maximizes the information gain in each step. Fortunately, while selecting the clusters to merge, we do not need to count complete information loss for all possible mergings. It is enough to count the components of the two candidate clusters (x, y) and one resulting cluster (z). So the merging condition:

$$\arg\min_{C_x, C_y} \Delta I_z - (\Delta I_x + \Delta I_y). \tag{12}$$

Due to this property and Property 1 we can construct the algorithm that is very efficient. In such an algorithm, the clusters in each step are represented only with their consensus trees and the amount of trees assigned. Moreover, the information loss in each step is not completely counted, because the minimum loss linkage in each step can be determined on the basis of consensus trees informativity in the previous step. It can be shown, (which we omit due to lack of space), that for a agg-inf clustering for a given z used for z-distance and z-restricted strict consensus tree

$$\Delta I' - \Delta I = l_x * (|S_{C_x}| - |S_{C_x} \cap^z S_{C_y}|) + l_y * (|S_{C_y}| - |S_{C_x} \cap^z S_{C_y}|), \qquad (13)$$

where l_x and l_y are the amount of trees in clusters candidate for merge and $|S_c|$ is the amount of splits in corresponding consensus trees.

3.3 Clustering of Leaf-Labelled Trees with Free Leafset with z-Restricted Majority Rule Consensus Tree as a Representative Tree

The aim of this clustering is to divide the trees into k groups in such a way that the information gain towards the z-restricted majority-rule consensus tree is maximal. For this purpose we choose k-mean clustering algorithm. Because of Property 2, which states that majority-rule consensus tree is a middle tree, we can use it as a centroid in k-mean algorithm whose objective function will be automatically identical to ours because its objective function is as follows:

$$\min_{C,\{T_{M_k}\}_{k=1}^{K}} \sum_{k=1}^{K} \sum_{C(i)=k} d(T_i, T_{M_k}). \qquad (14)$$

3.4 Z Parameter Selection

The main problem of this approach is the selection of set z. We may think of an application, for example from phylogenetic analysis, where particular taxa let's say a, b, c, d are of a special interest. In this case, the quite obvious thing is to choose a set z as *abcd*. On the other hand, we may also think of such a clustering where no particular taxa is preferred. For a phylogenetic or duplication trees, where all the clustered trees share most but not all leaves, we may choose z as an intersection of leaves. However, when the input data contains a weakly connected set of leaves such an approach will not bring any reasonable results. There is a need to provide a distance measure that does not require arbitrary z selection, for example based on coherence relation. Construction of a middle tree for such distance is required as well. We intend to do it in future studies.

4 Results

Below we describe the results of clustering tandem duplication trees, which are the leaf-labelled trees on free leafset taken from a bioinformatics field. Tandem duplication is a DNA sequence built of the adjacent copies of a pattern. The adjacent copies are not exactly the same as they diverged over time, due to point mutations. Tandem duplications are thought to be a result of events based on the duplication of one or more already existent copies. Tandem duplication process can be illustrated as a leaf-labelled tree where the labels on leaves correspond to

Table 1. Quality of clustering with various algorithms

k	Agg-inf	Agg-min	Agg-max	Agg-compl	K-mean
10	0.83	0.49	0.73	0.73	0.85
9	0.76	0.41	0.73	0.73	0.65
8	0.68	0.32	0.62	0.62	0.68
7	0.61	0.25	0.54	0.54	0.60
6	0.51	0.19	0.50	0.50	0.53
5	0.45	0.11	0.29	0.37	0.49

Table 2. Sample clustering results

k	Agg-inf	Agg-min	Agg-max	K-mean

5-12	48(1.0):81(2.0):
	80(2.0):78(1.0):
	93(1.0): 77(1.0):
	98(1.0):69(2.0)
		423(0.0): 39(2.0):
		46(2.0): 8(2.0):
		34(2.0): 42(2.0):
		21(2.0): 11(2.0)
			69(1.0): 189(1.0):
			134(1.0): 11(2.0):
			42(2.0): 77(1.0):
			46(2.0): 56(2.0)
				202(0.0): 15(2.0):
				61(1.0): 23(2.0):
				6(6.0): 17(3.0):
				11(6.0): 16(2.0)

9-12	40(1.0):170(0.0):
	41(1.0):32(1.0):
	15(2.0):7(4.0):
	14(2.0): 32(1.0)
		343(0.0):1(6.0):
		1(6.0):1(6.0):
		1(6.0):1(6.0):
		2(4.0): 1(6.0)
			277(0.0):17(1.0):
			15(2.0):12(2.0):
			8(1.0):7(1.0):
			5(2.0): 10(1.0)
				158(0.0):54(1.0):
				37(1.0):18(2.0):
				15(2.0):4(6.0):
				21(3.0): 44(1.0)

the position of a given copy in a sequence. There are techniques that are able to reconstruct such a tree, basing on a sequence, especially the differences between the copies [5]. In general cases such trees are unrooted due to problems with estimating time on the basis of those differences. Here we have performed experiments on tandem duplication trees which were reconstructed with DTScore algorithm [5]. The sequences were retrieved from Tandem Repeats Database [6]. We have examined trees that contained from 5 up to 12 copies due to efficiency barriers considering trees reconstruction. The selection of z was natural as an intersection of leafsets of examined trees. So when examining for example trees consisting of 9-12 copies at a time, a leafset containing $1, 2, 3, 4, 5, 6, 7, 8, 9$ is chosen. As a sample of results we present the agg-inf algorithm as opposed to standard min, max and complete linkage clustering for strict consensus and k-mean algorithm for majority-rule consensus. The input trees were pre-processed by removing duplicating trees for more reliable results. In the Table 1 we show the results of clustering the trees with 5-12 copies, for a different number of clusters (k). Because of the large possible number of different trees the clustering results is only reliable for at least 5 groups. In the Table 2 the sample clustering results for 8 groups are presented, where trees with 5-12 and 9-12 leaves were tested. The results are presented in format: 203(0.0):179(1.0): 69(2.0): 80(2.0): 93(1.0) which indicates how many trees were assigned to the following groups: 203,179,69,93 and what was the informativity (numer of non-trivial splits) of a representative tree -(value in brackets). In all cases, the agg-inf strategy was better then others, even up to 75%. For experiments with smaller range of copies, the informativity of representative trees was significantly higher.

5 Discussion

In this paper we have presented the methodology aimed at the clustering leaf-labelled trees on a free leafset. Although we perform experiments for tandem duplication data, our approach is described in general terms. In the future there will be a need to construct a better distance measure that does not require arbitrary z selection and allows more accurate clustering. A middle tree for such a distance is also required. There is also a need to test the proposed methods for trees from other disciplines.

References

1. Koperwas, J., Walczak, K.: Clustering of leaf labeled-trees. In: ICANNGA 2007. Part I, LNCS, vol. 4431, pp. 702–710. Springer, Heidelberg (2007)
2. Ganeshkumar, G., Warnow, T.: Finding a maximum compatible tree for a bounded number of trees with bounded degree is solvable in polynomial time. In: Gascuel, O., Moret, B.M.E. (eds.) Algorithms in Bioinformatics. LNCS, vol. 2149, pp. 156–163. Springer, Heidelberg (2001)
3. Barthelemy, J.P., McMorris, F.R.: The median procedure for n-trees. J. Classif. 3, 329–334 (1986)
4. Bryant, D.: Building trees, hunting for trees, and comparing trees. Theory And Method. In: Phylogenetic Analysis. Ph.D Thesis University of Canterbury (1997)
5. Elemento, O. et al.: Reconstructing the duplication history of tandemly repeated genes. Molecular Biology and Evolution 19, 278–288 (2002)
6. Tandem Repeats Database, http://tandem.bu.edu
7. Stockham, C., Wang, L.S., Warnow, T.: Statistically based postprocessing of phylogenetic analysis by clustering. Bionformatics 18, 285–293 (2002)
8. Amenta, N., Klingner, J.: Case study: Visualizing sets of evolutionary trees. 8th IEEE Symposium on Information Visualization, pp. 71–74 (2002)
9. Akutsu, T., Halldrsson, M.: On the approximation of largest common point sets and largest common subtrees. Unpublished manuscript (1997)
10. Gascuel, O. et al.: The combinatorics of tandem duplication trees. Systematic Biology 52(1), 110–118 (2003)
11. Bille, P.: Tree edit distance, alignment distance and inclusion. Technical report TR-2003-23 in IT University Technical Report Series (2003)

Checking Brain Expertise Using Rough Set Theory

Andrzej W. Przybyszewski

[1] Dept Psychology McGill University, Montreal, Canada
[2] Dept of Neurology, University of Massachusetts Medical Center, Worcester, MA US
przy@ego.psych.mcgill.ca

Abstract. Most information about the external world comes from our visual brain. However, it is not clear how this information is processed. We will analyze brain responses using machine learning methods based on rough set theory. We will test the expertise of the visual area V4, which is responsible for shape classifications. Characteristic of each stimulus are treated as a set of learning attributes. We assume that bottom-up information is related to hypotheses, while top-down information is related to predictions. Therefore, neuronal responses are divided into three categories. Category 0 occurs if cell response is below 20 spikes/s (sp/s), indicating that the hypothesis is not valid. Category 1 occurs if cell activity is higher than 20 spikes, implying the hypothesis is valid. Category 2 occurs if cell response is above 40 sp/s; in this case we conclude that the hypothesis and prediction are valid. By using experimental data we make a decision table for each cell, and generate equivalence classes. We express the brains basic concepts by means of the learners basic categories. By approximating stimulus categories with concepts of different cells we determine core properties of cells, and differences between them. On this basis we have created profiles of their receptive field properties.

Keywords: V4, machine learning, bottom-up, top-down processes, neuronal activity.

1 Introduction

Most of our knowledge about function of the brain is based on electrophysiological recordings from single neurons. In the lower visual areas like the retina, LGN or V1 (primary visual cortex) it is relatively easy to find an optimal stimulus for each neuron. The receptive fields in these areas are small and simple. On the other end, in the area designated as IT (inferotemporal cortex), receptive fields are very large and optimal stimuli are generally unknown, though they could be as complex as faces. In consequence, different laboratories propose different often contradictory hypotheses on the basis of their different testing stimuli. Another part of the confusion is related to non-uniform properties of neurons in area V4 of the brain. Therefore we do not know if different experimental results and hypotheses are related to different methods and classifications or to different classes of cells.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 746–755, 2007.

In order to clarify these confusions, we propose the use of rough set theory (Pawlak, [1]) to classify concepts of different cells as related to different stimuli attributes. We define an information system [1] as a pair $S = (U, A)$ where U denotes a nonempty set of objects, and A set of attributes. For each pair (a, u), $a \in A$, $u \in U$ the value $a(u)$ is a unique element of V (a value set). The *indiscernibility relation* of any subset B of A, or $IND(B)$, is defined [1] as follows: $(x, y) \in IND(B)$ if and only if $a(x) = a(y)$ for every $a \in B$, where $a(x) \in V$. $IND(B)$ is the equivalence relation, and $[u]_B$ is the equivalence class of u. The concept $X \subseteq U$ is $B-definable$ if for each $u \in U$ either $[u]_B \subseteq X$ or $[u]_B \subseteq U - X$. $\underline{B}\, X = \{u \in U : [u]_B \subseteq X\}$ is a lower approximation of X. The concept $X \subseteq U$ is $B-indefinable$ if is not $B-definable$ and exists such $u \in U$ that $[u]_B \cap X \neq \emptyset$. $\bar{B}X = \{u \in U : [u]_B \cap X \neq \emptyset\}$ is an upper approximation of X.

2 Methods

Most of our analysis will be related to data from Pollen et al. [2]. As mentioned above we have divided all cell responses into three ranges. Activity below 20 sp/s is defined as a category 0 cell response. Activity above 20 sp/s is defined as category 1, and activity above 40 sp/s as category 2. The reason for choosing the minimum significant cell activity of 20 sp/s is as follows. During normal activity our eyes are constantly moving. Our fixation periods are between 100 and $300ms$, which is similar to those of monkeys (averaged fixation duration was $195 \pm 168ms(SD)$, median $144ms$ [3]).

Assuming that a single neuron, in order to give reliable information about an object, must fire a minimum of 2-3 spikes during the eye fixation period, we obtained a minimum frequency of 20 sp/s. We assume that these discharges are related to bottom-up information (hypothesis testing) and that they are related to the objects form.

The brain is constantly making predictions which are verified by comparing them with sensory information. These tests are performed in a positive feedback loop ([4], [5]). If prediction is in agreement with the hypothesis, activity of the cell increases approximately twofold ([4]). This increased activity is related to category 2. (neuronal discharges of 40 sp/s). We will represent data from Pollen et al. [2] in the following table. In the first column there are different measurements of neurons. Neurons are classified by numbers related to various figures in [2]. Different measurements of the same cell are denoted by letters (a, b,). For example, 11a denotes the first measurement in neuron 1 Fig. 1, 11b - etc. Stimulus properties are as follows:

1. orientation in degrees appears in the column labeled o, and orientation bandwidth is labeled by ob.
2. spatial frequency is denoted as sf, spatial frequency bandwidth is sfb
3. x-axis position is denoted by xp and the range of x-positions is xpr

4. y-axis position is denoted by yp and the range of y-positions is ypr
5. x-axis stimulus size is denoted by xs
6. y-axis stimulus size is denoted by ys
7. stimulus shape is denoted by s: for grating $s = 1$, for vertical bar $s = 2$, for horizontal bar $s = 3$, for disc $s = 4$, for annulus $s = 5$

Stimulus attributes can be express as: $B = \{o, ob, sf, sfb, xp, xpr, yp, ypr, xs, ys, s\}$. Cell responses are denoted by r and divided into three ranges: r_0: activity below 20 sp/s; r_1: activity above 20sp/s; r_2: activity above 40sp/s .

3 Results

We have analyzed several neurons from [2]. Below we have shown modified figures from the above work, along with their decision tables. On this basis we have generated figures comparing the category of the stimulus with the concept of the brain cell. Fig. 1 shows tests performed on two neurons. Curves describe responses to long narrow bars which in Fig. 1A, C are oriented vertically and in Fig. 1B, D horizontally. They change their position along the x and y axis. The light intensity of bars is constantly changing these are so-called drifting gratings [2]. The cell in the left part of Fig. 1 (Fig. 1A, B) does not show strong responses. Only when a vertical (Fig. 1A) or horizontal bar (Fig. 1C) is near the middle of the receptive field the cells activity reaches 20 spikes/s. It means that this stimulus has category 1. More interesting is the second cell (on the right Fig. 1C, D). It shows several areas of strong activity where not only category 1 but

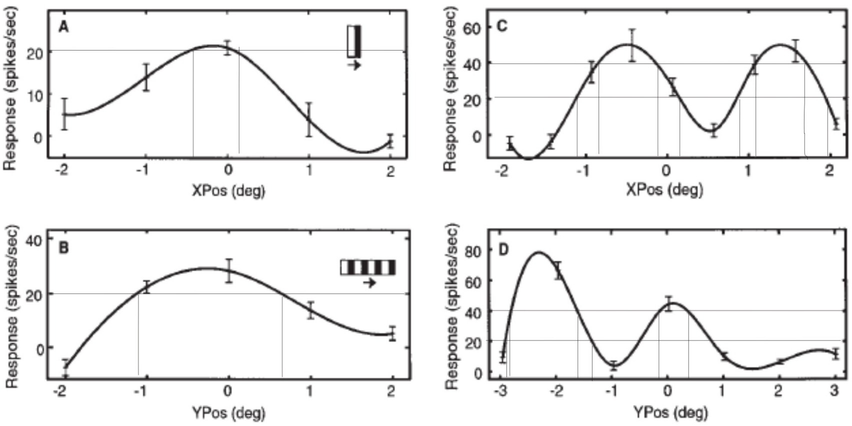


Fig. 1. Curves represent approximated responses of two cells (A,B) and (C, D) from area V4 to vertical and horizontal bars. Bars changed their position in Xpos or Ypos directions and responses of the cell was measured. Mean SE are marked in the figures. Stimulus attributes are shown in the table below. Cell responses are divided to two ranges (concepts) by horizontal lines. Plots are modified on the basis of [2].

also category 2 are realized. As one can notice, these *hot spots* are not symmetric along the middle of the receptive field, but they divide the receptive field into several smaller subfields. Such results are the basis of the idea that the receptive field of V4 neurons can be divided into several independent parts (see Fig. 3). In the next step of our analysis, we have converted these data into decision table (Table 1). In the top row of the table is a list of stimulus attributes, next two rows describe the first cell other rows describe the second cell from Fig. 1. As it was mentioned above different rows are related to different measurements. Results presented in the decision table for the second cell are shown in Fig. 2 as the preferred stimulus for this cell. Fig. 2 shows areas in the receptive field where category 1 (left side) and category 2 (right side) are fulfilled and become concept 1 and concept 2.

Table 1. Decision table for two cells shown in Fig. 1. Attributes ob, sf, sfb were constant and they are not presented in the table

cell	*o*	*xp*	*xpr*	*yp*	*ypr*	*xs*	*ys*	*s*	*r*
11a	90	0	0.6	0	0	0.5	1	2	1
11b	0	0	0	-0.4	1.5	2	0.5	3	1
12a	90	-0.6	1.3	0	0	0.4	4	2	1
12a1	90	-0.6	0.8	0	0	0.4	4	2	2
12a2	90	1.3	1.1	0	0	0.4	4	2	1
12a3	90	1.3	0.6	0	0	0.4	4	2	2
12b	0	0	0	-2.2	1.5	4	0.4	3	1
12b1	0	0	0	-2.2	1.2	4	0.4	3	2
12b2	0	0	0	0.15	1.4	4	0.4	3	1
12b3	0	0	0	0.15	0.5	4	0.4	3	2

Let us define $0 \leq xpr \leq 0.8$ will be sign as xpr_n (narrow bar x-range), $0 \leq ypr \leq 1.2$ will be sign as ypr_n (narrow bar y-range),

Decision rules related to the cell in Fig. 1C, D are following:

DR1: $o_{90} \wedge (xp_{-0.6} \vee xp_{1.3}) \wedge xpr_n \wedge xs_{0.4} \wedge ys_4 \rightarrow r_2$
DR2: $o_0 \wedge (yp_{-2.2} \vee yp_{0.15}) \wedge ypr_n \wedge xs_4 \wedge ys_{0.4} \rightarrow r_2$

Fig. 3 shows responses of a V4 cell tested with different stimuli. Fig. 3A shows cell responses to different orientation of grating of a large disc covering the receptive field (RF). Fig. 3B shows changes in cell response when the width of the stimulus was changed. Figs. 3C-F show cell responses when different subfields of the RF were stimulated with different stimulus orientation. Cell responses were also tested when the same subfields were stimulated with different spatial frequencies (Fig. 5 in [2]). These results are summarized in the Table 2.

Let us simplify $0 < ob < 50$ will be sign as ob_n (narrow orientation bandwidth), $ob > 100$ as ob_w (wide orientation bandwidth), $0 < sfb < 2$ as sfb_n , and $sfb > 2.5$ sfb_w.

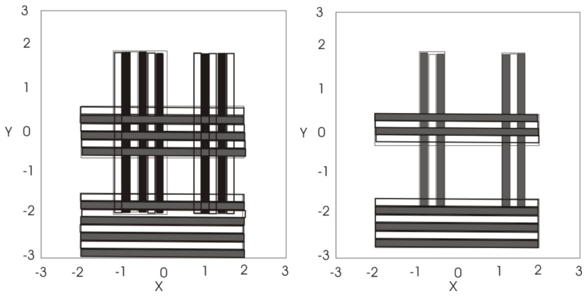


Fig. 2. Schematic representation of Table 1. Long bars have approximate concepts of the stimulus, with their positions in the receptive field related to the concept in the brain. The left schematic represents concept 1, while the right side represents concept 2.

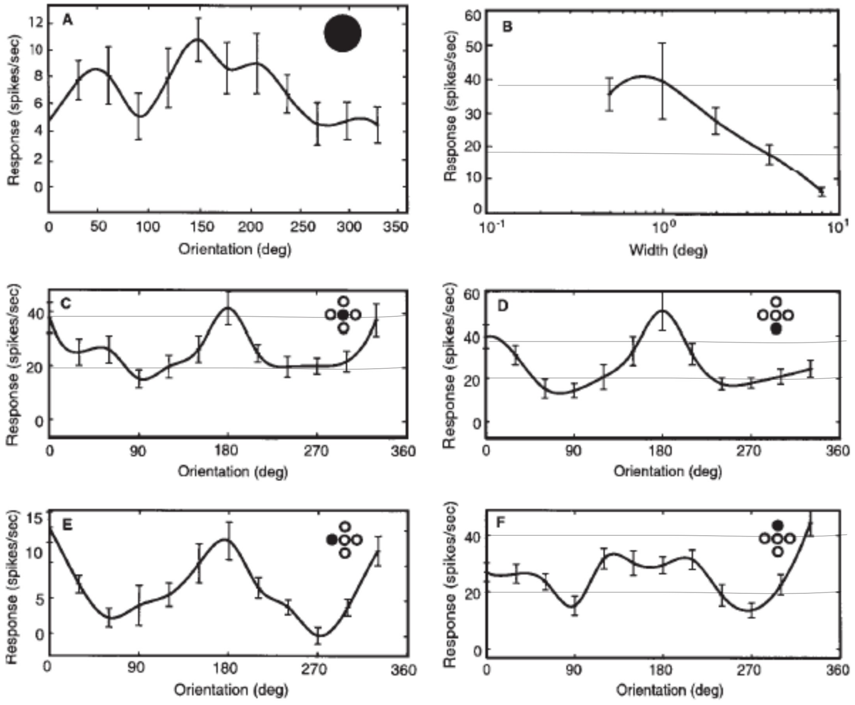


Fig. 3. Modified plots on the basis of [2]. One V4 cell tested with different stimuli. A. a large disc of grating covering the whole receptive field B. a large slit of light which changes its width. Notice the optimal width is around 1 deg. C-F Curves representing responses of the same cell when its subfields (their positions are shown in plots) are covered with a small 2 deg grating discs 2 deg apart in a 6 deg receptive field.

Table 2. Decision table for one cell responses to subfields stimulation Fig. 3C-F and Fig.5 in [2]. Attributes xpr, ypr, s are constant and they are not presented in the table.

cell	o	ob	sf	sfb	xp	yp	r
3c	172	105	2	0	0	0	1
3c1	10	140	2	0	0	0	1
3c2	180	20	2	0	0	0	2
3d	172	105	2	0	0	-2	1
3d1	5	100	2	0	0	-2	1
3d2	180	50	2	0	0	-2	2
3e	180	0	2	0	-2	0	0
3f	170	100	2	0	0	2	1
3f1	10	140	2	0	0	2	1
3f2	333	16	2	0	0	2	2
5a	180	0	2.3	2.6	0	-2	1
5b	180	0	2.5	3	0	2	1
5c	180	0	2.45	2.9	0	0	1
5c1	180	0	2.3	1.8	0	0	2

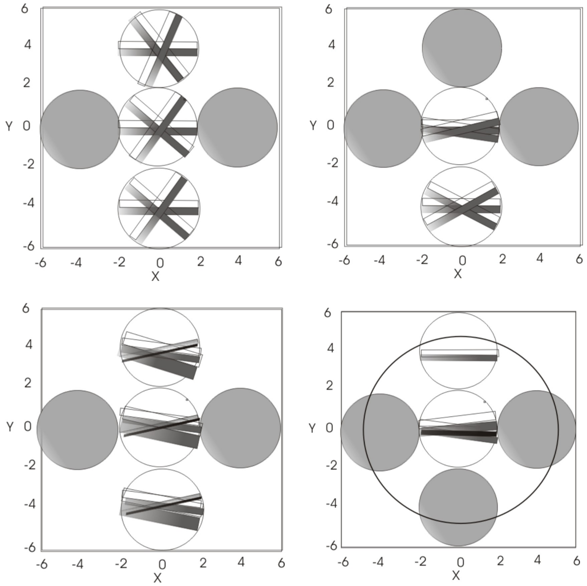


Fig. 4. Schematic representation of Table 2. Receptive field was divided into five subfields which were stimulated separately. Gray circles indicate cell response was below 20 spikes/s. The two upper plots represent subfields tuning to different orientations, whereas the two lower plots describe spatial frequency tuning. Plots on the left are related to concept 1, and plots on the right to concept 2. Notice that on the basis of the plots on the right one can imagine an optimal stimulus. It cannot be the same stimulus in all subfields because it does not give a strong response (Fig. 3A).

Table 3. Decision table for eight cells comparing the center-surround interaction. All stimuli were concentric, and therefore attributes were not xs, ys, but xo outer diameter, xi inner diameter. All stimuli were localized around middle of the receptive field so $xp = yp = xpr = ypr = 0$ and were skipped.

cell	*ob*	*sf*	*sfb*	*xo*	*xi*	*s*	*r*
101	0	0.5	0	7	0	4	0
101a	0	0.5	0	7	2	5	1
102	0	0.5	0	8	0	4	0
102a	0	0.5	0	8	3	5	0
103	0	0.5	0	6	0	4	0
103a	0	0.5	0	6	2	5	1
104	0	0.5	0	8	0	4	0
104a	0	0.5	0	8	3	5	2
105	0	0.5	0	7	0	4	0
105a	0	0.5	0	7	2	5	1
106	0	0.5	0	6	0	4	1
106a	0	0.5	0	6	3	5	2
107	0	0.5	0.25	6	0	4	2
107a	0	2.1	3.8	6	2	5	2
107b	0	2	0	4	0	4	1
108	0	0.5	0	6	0	4	1
108a	0	2	0	4	0	4	2
108b	0	5	9	6	2	5	2
20a	0.5	0.5	0	6	0	4	1
20b	0.3	0.5	0	6	0	4	2

Decision rules related to cell from Fig. 3 are following:

DR3: $ob_n \wedge (yp_0 \vee yp_2) \rightarrow r_2$
DR4: $ob_w \wedge xp_0 \rightarrow r_1$
DR5: $sfb_n \wedge yp_0 \rightarrow r_2$
DR6: $sfb_w \wedge xp_0 \rightarrow r_1$

Notice that Figs. 2 and 4 show possible configurations of the optimal stimuli. However, they do not take into account interactions between several stimuli, when more than one subfield is stimulated.

Therefore we propose following **Subfield Interaction Rules:**

SIR1: facilitation when stimulus consists of multiple bars with small distances $(0.5 - 1$ deg) between them, and inhibition when distance between bars is $1.5 - 2$ deg.
SIR1: inhibition when stimulus consists of multiple similar discs with distance between them ranging from 0 deg (touching) to 3 deg.
SIR1: Center-surround interaction, which is described below in detail.

We will concentrate on the center-surround interaction. We will make a decision table for nine different cells tested with the disc covering their receptive field and an annulus when the center of the receptive field is not stimulated (Pollen et al. [2] Fig. 10). If the center is stimulated with another stimulus attributes then the surround inhibitory mechanism is also weak (Fig. 9B in [2]). In order to compare different cells, we have normalized their optimal orientation which will be denoted as 1.

The experiments test receptive field with disc and annulus stimuli, which could be, described as following six categories:

$$Y_0 = |o_1\ ob_0\ sf_{0.5}\ sfb_0\ xo_7\ xi_0\ s_4| = \{101, 105\}$$

$$Y_1 = |o_1\ ob_0\ sf_{0.5}\ sfb_0\ xo_7\ xi_2\ s_5| = \{101a, 105a\}$$

$$Y_2 = |o_1\ ob_0\ sf_{0.5}\ sfb_0\ xo_8\ xi_0\ s_4| = \{102, 104\}$$

$$Y_3 = |o_1\ ob_0\ sf_{0.5}\ sfb_0\ xo_8\ xi_3\ s_5| = \{102a, 104a\}$$

$$Y_4 = |o_1\ ob_0\ sf_{0.5}\ sfb_0\ xo_6\ xi_0\ s_4| = \{103, 106, 107, 108, 20a\}$$

$$Y_5 = |o_1\ ob_0\ sf_{0.5}\ sfb_0\ xo_6\ xi_2\ s_5| = \{103a, 106a, 107a, 108b, 20b\}$$

$$Y_6 = |o_1\ ob_0\ sf_2\ sfb_0\ xo_4\ xi_0\ s_4| = \{107b, 108a\}$$

which are equivalence classes for stimulus attributes, which means that in each class they are indiscernible $IND(B)$. For simplicity we simplify orientation bandwidth to 0 in $\{20a, 20b\}$ and spatial frequency bandwidth to 0, in cases $\{107, 107a, 108a, 108b\}$, and put values covered by the bandwidth to the spatial frequency parameters. There are three ranges of responses denoted as r_o, r_1, r_2 therefore the experts knowledge involves the following three concepts:

$$|r_o| = \{101, 102, 102a, 103, 104, 105\}$$

$$|r_1| = \{101a, 103a, 105a, 107b, 108, 20a\}$$

$$|r_2| = \{104a, 106a, 107, 107a, 108a, 108b, 20b\}$$

which will be denoted as X_o, X_1, X_2.

We want to find out whether equivalence classes of the relation $IND\{r\}$ form the union of some equivalence relation $IND(B)$, or whether $B \Rightarrow \{r\}$. We will calculate the lower and upper approximation [1] of the brains basic concepts in term of stimulus basic categories:

$\underline{B}\ X_0 = Y_0 = \{101, 105\}$
$\bar{B}X_0 = Y_0 \cup Y_2 \cup Y_3 \cup Y_4 = \{101, 105, 102, 104, 102a, 104a, 103, 106, 107, 108, 20a\}$
$\underline{B}\ X_1 = Y_1 = \{101a, 105a\}$
$\bar{B}X_1 = Y_1 \cup Y_5 \cup Y_6 \cup Y_4 =$
$\{101a, 105a, 103a, 107a, 108b, 106a, 20b, 107b, 108a, 103, 107, 106, 108, 20a\}$
$\underline{B}\ X_2 = 0$
$\bar{B}X_2 = Y_3 \cup Y_4 \cup Y_5 \cup Y_6 =$
$\{102a, 104a, 103a, 107a, 108b, 106a, 20b, 103, 107, 106, 108, 20a, 107b, 108a\}$

Concept 0 and concept 1 are roughly $B-defined$, which means that only with some approximation can we say that stimulus Y_o does not evoke a response (concept 0) in cells 101, 105, but that other stimuli Y_2, Y_3 can evoke no response or weak (concept 1) or strong (concept 2) response. This is similar for concept 1. However, concept 2 is internally $B - undefinable$. Stimulus attributes related to this concept should give us information about cell characteristics, but data from the Table 3 cannot do it.

We can find quality [1] of our experiments by comparing properly classified stimuli $POSB(r) = \{101, 101a, 105, 105a\}$ to all stimuli and to all responses: $\gamma\{r\} = \frac{card\{101,101a,105,105a\}}{card\{101,101a,,20a,20b\}} = 0.2$. We can also ask what percentage of cells we fully classified. We obtain consistent responses from 2 of 9 cells, which means that $\gamma = 0.22$. This is related to the fact that for some cells we have tested more than two stimuli. What is also important from an electrophysiological point of view is there are negative cases. There are many negative instances for the concept 0, which means that in many cases this brain area responds to our stimuli; however it seems that our concepts are still only roughly defined. **Decision rules** related to cells listed in the Table 3 are following:

DR7: $xo_7 \wedge xi_2 \wedge s_5 \rightarrow r_1$
DR8: $xo_7 \wedge s_4 \rightarrow r_0$
DR9: $xo_8 \wedge s_4 \rightarrow r_0$

They can be interpreted that large annulus (s5) evokes weak response, but large disc (s4) evokes no response.

4 Discussion

The purpose of our study was to determine how different categories of stimuli and particular concepts, as related to the expertise of a single cell. We can test our theory on a set of data from David et al. [5], shown in Fig.5.

Assuming that the stimulus configuration in top two images on the left side is similar to that proposed in Fig. 2, we can apply DR2 and SIR1. This means that these images will be related to concept 2. Top-right and bottom-left images show significant differences between their center and surround, therefore these images would also give significant responses. However, in the top-right image only part of the surround is stimulated therefore DR4, DR6, and DR7 rules are applied. In the bottom-left image the object is localized in part of the center and part of the surround: DR5 but SIR3. In consequence responses to both images are related to the concept 1. In two bottom-right images there is no significant difference between stimulus in the center and the surround. Therefore the response will be similar to that obtained when a single disc covers the whole receptive field: DR8, DR9. In most cells such a stimulus is classified as concept 0.

In summary, we have showed that using rough set theory we can divide stimulus attributes in relationships to neuronal responses into different concepts. Even if most of our concepts were very rough, they determine rules on whose basis we can predict neural responses to new, natural images.

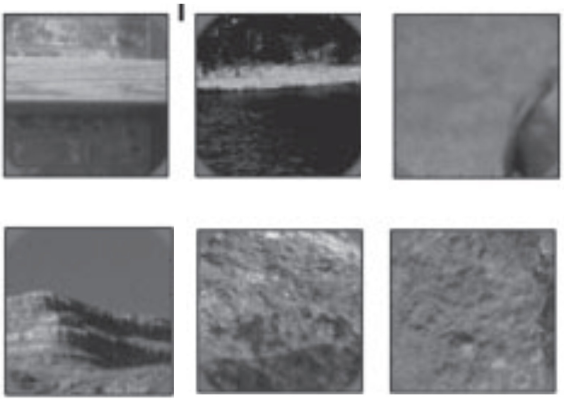


Fig. 5. In their paper David et al. [6] stimulated V4 neurons (medium size of their receptive fields was 10.2 deg) with natural images. Several examples of their images are shown above. We have divided responses of their cells into three concepts. Two left images in top gave strong responses above 40 sp/s related concept 2. Image top-right and bottom-left evoke responses above 20 sp/s related to concept 1. Two images on the right in bottom row gave very weak related to concept 0 responses.

Acknowledgement. I thank M. Kon for useful discussion and comments.

References

1. Pawlak, Z.: Rough Sets - Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
2. Pollen, D.A., Przybyszewski, A.W., Rubin, M.A., Foote, W.: Spatial receptive field organization of macaque V4 neurons. Cereb Cortex 12, 601–616 (2002)
3. Mazer, J.A., Gallant, J.L.: Goal-related activity in V4 during free viewing visual search. Evidence for a ventral stream visual salience map. Neuron. 40, 1241–1250 (2003)
4. Przybyszewski, A.W., Gaska, J.P., Foote, W., Pollen, D.A.: Striate cortex increases contrast gain of macaque LGN neurons. Vis Neurosci. 17, 485–494 (2000)
5. Przybyszewski, A.W., Kon, M.A.: Synchronization-based model of the visual system supports recognition. In: Program No. 718.11. 2003 Abstract Viewer/Itinerary Planner, Society for Neuroscience, Washington, DC (2003)
6. David, S.V., Hayden, B.Y., Gallant, J.: Spectral receptive field properties explain shape selectivity in area V4. J. Neurophysiol. 96, 3492–3505 (2006)

Analysis of a Dobutamine Stress Echocardiography Dataset Using Rough Sets

Kenneth R. Revett

University of Westminster, Harrow School of Computer Science, London, England
HA1 3TP

Abstract. Stress echocardiography is an important functional diagnosis and prognostic tool that is now routinely applied to evaluate the risk of cardiovascular artery disease (CAD). A complete dataset containing data on 558 subjects undergoing a prospective longitudinal study is employed to investigate what attributes correlate with the final outcome. The dataset was examined using rough sets, which resulted in a series of decision rules that predict which attributes influence the outcomes measured clinically and recorded in the dataset. The results indicate that the ECG attribute was very informative. In addition, prehistory information has a significant impact on the classification accuracy.

1 Introduction

Heart disease remains the number one cause of mortality in the western world. Coronary arterial disease (CAD) is a primary cause of morbidity and mortality in patients with heart disease. The early detection of CAD was in part made possible in the late 1970s by the introduction of echocardiography a technique for measuring the physical properties of the heart using a variety of imaging techniques such as ultrasound, and doppler flow measurements. The purpose of these imaging studies is to identify structural malformations such as aneurysms and valvular deformities. Although useful, structural information may not provide the full clinical picture in the way that functional imaging techniques such as stress echocardiography (SE) may. This imaging technique is a versatile tool that allows clinicians to diagnosis patients with CAD efficiently and accurately. In addition, it provides information concerning the prognosis of the patient which can be used to provide on-going clinical support to help reduce morbidity.

The underlying basis for SE is the induction of cardiovascular stress, which generates ischemia, resulting in wall motion abnormality (WMA) distal to the coronary lesion. In addition to detecting CAD, the technique is also routinely employed to measure the extent of valvular heart disease. Normally, the walls of the heart (in particular the left ventrical) change (move) in a typical fashion in response to stress (i.e. heavy exercise). A quantitative measure called the wall motion score is computed and its magnitude is directly related to the extent of the WMA score. The WMA provides a quantitative measure of how the heart responds to stress. Stress echocardiography (SE) was originally induced under conditions of strenuous exercise such as bike and treadmills. In

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 756–763, 2007.

many cases though, patients are not able to exercise to the level required and pharmacological agents such as dobutamine or dipyridamole have been used to induce approximately the same level of stress on the heart as physical exercise. Dobutamine in particular emulates physical exercise effects on the cardiovascular system by increasing the heart rate and blood pressure and impacts cardiac contractility which drives cardiac oxygen demand [1]. A number of reports have indicated that though there are subtle differences between exercise and pharmacologically induced stress, they essentially provide the same stimulus to the heart and can therefore, in general, be used interchangeably [2], [3].

In this paper, we investigate the effectiveness of dobutamine stress echocardiography (DSE) by analysing the results of a large study of 558 patients undergoing DSE. The purpose is to determine which attributes collected in this study correlate most closely with the decision outcome. After a careful investigation of this dataset, a set of rules is presented that relates condition attributes to decision outcomes. This rule set is generated through the application of rough sets, a data mining technique developed by the late Professor Pawlak [4]. In the next section, we present an overview of the dataset, followed by a description of the pre-processing stages.

1.1 The Dataset

The data employed in this study was obtained from a prospective dobutamine stress echocardiography (DSE) study at the UCLA Adult Cardiac Imaging and Hemodynamics Laboratory held between 1991 and 1996. The patients were monitored during a five year period and then observed for a further twelve months to determine if the DSE results could predict patient outcome. The outcomes were categorised into the following cardiac events: cardiac death, myocardial infarction (MI), and revascularisation by percutaneous transluminal coronary angioplasty (PTCA) or coronary artery bypass graft surgery (CABG) [2], [3]. After normal exclusionary processes, the patient cohort consisted of 558 subjects (220 women and 338 men) with a median age of 67 (range 26-93). Dobutamine was administered intraveneously using a standard delivery system yielding a maximum dose of 40 g/kg/min. There were a total of 30 attributes collected in this study which are listed in Table 1. The attributes were a mixture of categorical and continuous values. The decision class used to evaluate this dataset was the outcomes as listed as listed above and in Table 1. As a preliminary evaluation of the dataset, the data was evaluated with respect to each of the four possible measured outcomes included in the decision table individually, excluding each of the other three possible outcomes. This process was repeated for each of the outcomes in the decision table. Next, the effect of the echocardiogram (ECG) was investigated. Reports indicate that this is a very informative attribute with respect to predicting the clinical outcome of a patient [3]. To evaluate the effect of ECG on the outcomes, the base case investigation (all four possible outcomes) was investigated with (base case) and without the ECG attribute. Lastly, we investigated whether any prehistory information would provide a correlation between the DSE and the outcome. There were a total of six different history

Table 1. The decision table attributes and their data types (continuous, ordinal, or discrete) employed in this study (see [2] for details). Note the range of correlation coefficients was 0.013 to 0.2476 (specific data not shown).

Attributename	*Attributetype*
bhr basal heart rate	Integer
basebp basal blood pressure	Integer
basedp basal double product (= bhr x basebp)	Integer
pkhr peak heart rate	Integer
sbp systolic blood pressure	Integer
dp double product (= pkhr x sbp)	Integer
dose dose of dobutamine given	Integer
maxhr maximum heart rate	Integer
mphr(b) % of maximum predicted heart rate	Integer
mbp maximum blood pressure	Integer
dpmaxdo double product on maximum dobutamine dose	Integer
dobdose dobutamine dose at which maximum double product	Integer
age	Integer
gender (male = 0)	Level (2)
baseef baseline cardiac ejection fraction	Integer
dobef ejection fraction on dobutamine	Integer
chestpain (0 experienced chest pain)	Integer
posecg signs of heart attack on ecg (0 = yes)	Integer
equivecg ecg is equivocal (0 = yes)	Integer
restwma wall motion anamoly on echocardiogram (0 = yes)	Integer
posse stress echocardiogram was positive (0 = yes)	Integer
newMI new myocardial infarction, or heart attack (0 = yes)	Integer
newPTCA recent angioplasty (0 = yes)	Level (2)
newCABG recent bypass surgery (0 = yes)	Level (2)
death died (0 = yes)	Level (2)
hxofht history of hypertension (0 = yes)	Level (2)
hxofptca history of angioplasty (0 = yes)	Level (2)
hxofcabg history of bypass surgery (0 = yes)	Level (2)
hxofdm history of diabetes (0 = yes)	Level (2)
hxofMI history of heart attack (0 = yes)	Level (2)

attributes (see Table 1) that were tested to determine if each in isolation had a positive correlation with the outcomes. In the next section, we describe the experiments that were performed using rough sets (RSES 2.2.1).

2 Results

In the first experiment, each outcome was used as the sole decision attribute. The four outcomes were: new Myocardial Infarction (MI) (28 cases), death (24 cases), newPTCA (27 cases), and newCABG (33 cases). All continuous attributes were discretised using the MDL algorithm within Rosetta ([9]). Note there were no missing values in the dataset. A 10-fold cross validation was performed using

Table 2. Confusion matrices for the base cases of the four different outcomes. The label A corresponds to death, B to MI, C to new PTCA, and D to newCABG. Note the overall accuracy is placed at the lower right hand corner of each subtable (large bold).

A	0	1		*B*	0	1	
0	204	7	0.97	0	205	4	0.98
1	2	10	0.80	1	0	14	1.0
	0.95	0.22	0.92		0.94	0	0.92
C	0	1		D	0	1	
0	207	9	0.96	0	191	25	0.88
1	1	6	0.14	1	0	7	1.0
	0.97	0.10	0.93		0.96	0.0	0.93

decision rules and dynamic reducts. The results reported here are the average values from 10 executions under identical conditions. Without any filtering of the reducts or rules, Table 2 presents randomly selected confusion matrices that were generated for each of the decision outcomes for the base case. The number of rules was quite large and initially no filtering was performed to reduce either the number of reducts nor the number of rules. The number of reducts for panels A D in Table 2 were: 104, 159, 245, and 122 respectively. On average, the length of the reducts ranged from 5-9, out of a total of 27 attributes (minus the 3 other outcome decision classes). The number of rules (all of which were deterministic) was quite large, with a range of 23,356-45,330 for the cases listed in table 2. Filtering was performed on both reducts (based on support) and rule coverage in order to reduce the cardinality of the decision rules. The resulting decision rule set were reduced to a range of 314-1,197. The corresponding accuracy was reduced by approximately 4% (range 3- 6%). Filtering can be performed on a variety of conditions, such as LHS support, coverage, RHS support. For a discussion of rule filtering, please consult [5], [6], [8] for a comprehensive discussions of this topic.

The number of rules was quite large and initially no filtering was performed to reduce either the number of reducts nor the number of rules. The number of reducts for panels A D in Table 2 were: 104, 159, 245, and 122 respectively. On average, the length of the reducts ranged from 5-9, out of a total of 27 attributes (minus the 3 other outcome decision classes). The number of rules (all of which were deterministic) was quite large, with a range of 23,356-45,330 for the cases listed in table 2. Filtering was performed on both reducts (based on support) and rule coverage in order to reduce the cardinality of the decision rules. This technique has been employed successfully in similar types of biomedical datasets (see [10], [11]. The resulting decision rule set were reduced to a range of 314-1,197. The corresponding accuracy was reduced by approximately 4% (range 3-6%).

In the next experiment, the correlation between the outcome and the ECG result was examined. It has been reported that the ECG, which is a standard cardiological test to measure functional activity of the heart, should be corre-

Table 3. Confusion matrices for the base cases without the inclusion of the ECG attribute for the four different outcomes (as in Table 2). The label A corresponds to death, B to MI, C to new PTCA, and D to newCABG. Note the overall accuracy is placed at the lower right hand corner of each subtable (large bold).

A	0	1		*B*	0	1	
0	206	5	0.98	0	209	0	1.0
1	2	10	0.80	1	0	14	1.0
	0.95	0.22	0.92		0.94	0	0.92
C	0	1		D	0	1	
0	207	9	0.96	0	191	25	0.91
1	1	6	0.86	1	0	7	1.0
	0.97	0.10	0.93		0.96	0.0	0.93

lated with the outcome [2]. We therefore repeated the experiment in Table 2, with the ECG attribute excluded (masked) from the decision table. The results are reported in Table 3. Lastly, we examined the effect of historical information that was collected and incorporated into the dataset (see Table 1). These historical attributes include: history of hypertension, diabetes, smoking, myocardial infarction, angioplasty, and coronary artery bypass surgery. We repeated the base set of experiments (including ECG) and withheld each of the historical attributes one at a time and report the results as a set of classification accuracies, listed in Table 4.

Table 4. The classification accuracy obtained from the classification using the exact same protocol for the table reported in Table 2 (note the ECG attribute was included in the decision table). The results are the average over the four different outcomes.

Attributename	*Classificationaccuracy*
History of hypertension	91.1%
History of diabetes	85.3%
History of smoking	86.3%
History of angioplasty	90.3%
History of coronary artery bypass surgery	82.7%

In addition to classification accuracy, rough sets provides a collection of decision rules in conjunctive normal form. These rules contain the attributes and their values that are antecedents in a rule base. Therefore, the decision rules provide a codification of the knowledge contained within the decision table. Examples of the resulting rule set for the base case, using MI as the decision attribute is presented in table 5.

Lastly, to further validate and compare the accuracy of the classification element of this study, two standard neural networks (radial basis function and feed forward multi-layer) were applied to this dataset. With both neural network validation experiments, the inputs were the discretised version employed

Table 5. Sample set of rules from the base case (+ ECG) with death as the outcome. The right hand column lists the support (LHS) for the corresponding rule. These rules were selected randomly from the rule set.

Decisionrule	*Support*
dp([20716, *)) AND dobdose(40) AND hxofDM(0) AND anyevent(0) = death(0)	19
dp([*, 13105)) AND dobdose(40) AND hxofDM(0) AND anyevent(0) = death(0)	18
basebp([*, 159)) AND sbp([115, 161)) AND dose(40) AND dobdose(40) AND dobEF([61, 71)) AND hxofDM(0) = death(0)	24
dp([*, 13105)) AND dobdose(35) AND dobEF([53, 61)) AND hxofDM(1) = death(1)	14
dp([20633, 20716)) AND dobdose(40) AND baseEF([56, 76)) AND hxofDM(0) AND anyevent(1) = death(1)	9
dp([*, 13105)) AND dobdose(30) AND hxofCABG(0) AND anyevent(1) AND ecg([*, 2)) = death(1)	12

in the rough sets analysis and the outputs were the aforementioned four decision classes. For the feed forward multi-layered network, training was applied using a batch mode back propagation algorithm, with a momentum value - 0.1 and a learning rate parameter = 0.2. The data was divided 70/30 (training/testing) and the error was measured over 10 trials and the results averaged. The classification accuracy for this neural network was 86.8%. The radial basis function network employed the same input/outputs and produced an overall accuracy (after 10 trials, 70/30 training/testing) of 89.9%.

3 Conclusion

This dataset contained a complete set of attributes (30) that was a mixture of continuous and categorical data. The data was obtained from a prospective study of cardiovascular health obtained by professional medical personal (cardiographers). The attributes were obtained from patients undergoing stress echocardiography, a routine medical technique employed to diagnose cardiovascular artery disease. From the initial classification results, the specificity of the classification using rough sets was quite high (90+%) consistent with some literature reports [2]. The accuracy produced by rough sets was higher than that generated using neural networks such as multi-layer perceoptrons and a radial basis function. As can be seen in Table 2, the sensitivity of the test was quite low, resulting in a reduced classification accuracy. The effect of ECG, the attribute most correlated with the clinical outcome of CAD, was measured by masking this attribute. The results indicate that this attribute did not have a significant impact on the overall classification accuracy, but did manage to increase the sensitivity when it was excluded from the decision table. This is an interesting result that may require specific medical knowledge in order to interpret. The effect of patient history was examined, and the results (see Table 4) indicate that in general, relevant medical history did have a positive impact on the classification accuracy. This result was quantified by examining the classification accuracy when these 5 history factors were removed from the decision table (one at a time). The effect of their combination was not examined in this paper, which is left for future work. Lastly, the rule set that was produced yielded a consistently reduced set of attributes ranging from 4-9 attributes, greatly reducing the size of the dataset. As displayed in Table 5 - and generally across the rule set, the dp and dobdose attributes appear consistently (has a large support) within all decision outcomes (data not displayed). This type of analysis is a major product of the rough sets approach to data analysis extraction of knowledge from data.

This is a preliminary study that will be pursued in conjunction with a qualified cardiologist. The results generated so far are interesting and certainly consistent and in many cases superior to other studies [1],[3]. To this authors knowledge, this is the first report which examined the dobutamine SE literature using rough sets. Komorowski & Ohn have examined a similar dataset but the imaging technique and attributes selected were different from those used in the study investigated in this work [7]. It is hoped that close collaboration between medical experts and data mining engineers will provide the conditions necessary for a full extraction of knowledge from the data.

Acknowledgements

The author would like to acknowledge the source of the dataset used in this study:

http://www.stat.ucla.edu:16080/projects/datasets /cardiacexplanation.html

References

1. Armstrong, W.F.: Stress Echocardiography: Current Methodology and Clinical Applications. J. Am. Coll. Cardiology 45, 1739–1747 (2005)
2. Krivokapich, J., Child, J.S., Walter, D.O., Garfinkel, A.: Prognostic value of dobutamine stress echocardiography in predicting cardiac events in patients with known or suspected coronary artery disease. J. Am. Coll. Cardiology 33, 708–716 (1999)
3. Bergeron, S., Hillis, G., Haugen, E., Oh, J., Bailey, K., Pellikka, P.: Prognostic value of dobutamine stress echocardiography in patients with chronic kidney disease. American Heart Journal 153(3), 385–391 (2005)
4. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
5. Bazan, J., Szczuka, M.: The Rough Set Exploration System. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 37–56. Springer, Heidelberg (2005)
6. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: A tutorial. In: Rough Fuzzy Hybridization – A New Trend in Decision Making, pp. 3–98. Springer, Heidelberg (1999)
7. Komorowski, J., Orhn, A.: Modelling prognostic power of cardiac tests using rough sets. Artificial Intelligence in Medicine 15, 167–191 (1999)

8. Slezak, D.: Approximate Entropy Reducts. Fundamenta Informaticae (2002)
9. Rosetta: Rosetta: http://www.idi.ntnu.no/~aleks/rosetta
10. Revett, K., Gorunescu, F., Gorunescu, M., El-Darzi, E., Gorunescu, S.: A Hybrid Breast Cancer Diagnosis System: A combined Approach Using Rough Sets and Probabilistic Neural Networks. In: Eurocon 2005, IEE Conference, Belgrade, Serbia and Montenegro, November 22-24, 2005, pp. 1124–1127 (2005)
11. Revett, K., Khan, A.: Datamining a Hepatitis dataset Using Rough Sets, The 5th International Conference on Artificial Intelligence and Digital Communications, Research Notes in Artificial Intelligence and Digital Communications (AIDC) 2005, Craiova, Romania, pp. 12–20 (2005)

An Improved SVM Classifier for Medical Image Classification*

Yun Jiang[1,2], Zhanhuai Li[1], Longbo Zhang[1,3], and Peng Sun[1]

[1] College of Computer Science, Northwestern Polytechnical University, 710072, Xi'an, P.R. China

[2] College of Mathematics and Information Science, Northwest Normal University, 730070, Lanzhou, P.R. China

[3] School of Computer Science, Shandong University of Technology, Zibo 255049, China

Abstract. Support Vector Machine (SVM) has high classifying accuracy and good capabilities of fault-tolerance and generalization. The Rough Set Theory (RST) approach has the advantages on dealing with a large amount of data and eliminating redundant information. In this paper, we join SVM classifier with RST which we call the Improved Support Vector Machine (ISVM) to classify digital mammography. The experimental results show that this ISVM classifier can get 96.56% accuracy which is higher about 3.42% than 92.94% using SVM, and the error recognition rates are close to 100% averagely.

1 Introduction

Support vector machine (SVM) is a proven success and a state-of-the-art method in many areas, and a promising machine learning technique proposed by Vapnik and his group AT Bell Laboratories[1]. It is based on VC dimensional theory and statistical learning theory. For many practical problems, including pattern matching and classification[2][3], function approximation[4], data clustering and forecasting[5][6], support vector machine has drawn much attention and been applied successfully in recent years because of its greater generalization performance. An interesting property of SVM is that it is an approximate implementation of the structural risk minimization induction principle that aims at minimizing a bound on the generalization error of a model, rather than minimizing the mean square error over the data set[7]. SVM is considered as a good learning method that can overcome the internal drawbacks of neural networks[8]. But there exists a drawback which can not distinguish the importance of training sample attributes. Furthermore, it will take up more storage space when there are a large number of sample attributes. Although SVM has strong capabilities of recognizing patterns and good capabilities of fault-tolerance and generalization, SVM cannot reduce the input data and select the most important information.

* This paper is supported by National Science Foundation No. 60573096 and Gansu province Science Foundation of China No. 3ZS 051-A25-042.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 764–773, 2007.

Several techniques aim to reduce the prediction complexity of SVM by expressing the SVM solution with a smaller kernel expansion. Since one must compute the SVM solution before applying these post-processing techniques, they are not suitable for reducing the complexity of the training stage[9].

Rough Set Theory(RST), introduced by Pawlak[10] in his seminal paper of 1982, is a new mathematical approach to uncertain and vague data analysis and is also a new fundamental theory of soft computing [11]. In recent years, RST becomes an attractive and promising issue. RST can mine useful information from a large amount of data, generate decision rules without prior knowledge, and eliminate redundant information. It is used generally in many fields[12], such as knowledge discovery, machine learning, pattern recognition and data mining. In this paper, a new classification algorithm based on SVM and RST is proposed, which we call Improved Support Vector Machine (ISVM). ISVM inherits the merits of both SVM and RST. We apply ISVM to medical images classify. It is tested on real datasets MIAS[13](the Mammographic Image Analysis Society)and can get 96.56% accuracy which is higher about 3.42% than 92.94% using SVM, and the error recognition rates are close to 100%averagely.

The rest of the paper is organized as follows: Section 2 describes the theory of SVM. Section 3 presents rough set theory, the reduction algorithm and the Improved SVM algorithm–ISVM. In section 4, data pre-processing and feature extraction are introduced. In section 5, we present our experiments and results. Finally, in section 6, we show our conclusions and future work.

2 Support Vector Machine(SVM)[1]

Consider the problem of separable training vectors belonging to two separate classes,

$$G = \{(x_i, y_i)\}_{i=1}^{l}, \quad x_i \in R^n, \ y_i \in \{-1, 1\}, \ i = 1, \cdots, l \tag{1}$$

We should find a linear function,

$$y = f(x) = \omega\varphi(x) + b \tag{2}$$

That is to say, we should make the margin between the two classes points as possible as big, it is equal to minimize $\frac{1}{2}\|\omega\|^2$, so the optimal classification problem is transformed into a convex quadratic programming problem:

$$\min \quad \frac{1}{2}\|\omega\|^2 \quad s.t. \quad y_i((\omega \cdot x_i) + b) \geq 1, i = 1, \cdots, l \tag{3}$$

when the training points are non-linearly separable, (3) should be transformed into (4).

$$\min \quad \frac{1}{2}\|\omega\|^2 + c\Sigma_{i=1}^{l}\xi_i \quad s.t. \quad y_i((\omega \cdot x_i) + b) \geq 1 - \xi_i, i = 1, \cdots, l \tag{4}$$

The solution to the above optimization problem of equation (4) is transformed into the dual problem (5) by the saddle point of the Lagrange functional,

$$\begin{aligned} \max \quad & \Sigma_{i=1}^{l}\alpha_i - \frac{1}{2}\Sigma_{i=1}^{l}\Sigma_{j=1}^{l}\alpha_i\alpha_j y_i y_j k(x_i, x_j) \\ s.t. \quad & \Sigma_{i=1}^{l}\alpha_i y_i = 0, \quad 0 \le \alpha_i \le c, \; i = 1, \cdots, l \end{aligned} \tag{5}$$

We can get the decision function:

$$f(x) = \Sigma_{i=1}^{l}\alpha_i y_i K(x_i, x) + b \tag{6}$$

kernel function $K(x_i, x) = (\Phi(x_i) \cdot \Phi(x))$ is a symmetric function satisfying Mercer's condition, when given the sample sets are not separate in the primal space, we can be used to map the data with mapping Φ into a high dimensional feature space where linear classification is performed.

There are three parameters in SVM model that we should choose, they make great impact on model's generalization ability. It is well known that SVM generalization performance depends on a good setting of hyperparameters C, the kernel function and kernel parameter. Moreover, kernel function and kernel parameter's selection connects with feature selection in SVM, so feature selection is very important.

3 The Improved Support Vector Machine Algorithm (ISVM)

In this section, the theory of rough sets has been used in the first stage to reduce the original feature sets. In the second stage the SVM algorithm has been executed with the reduced feature sets. The reduction of original feature sets results in a smaller structure and quicker learning of the SVM and as a whole the hybrid algorithm provides better performance than the SVM algorithm from individual paradigm. The following are the basic concepts of the rough set theory, the algorithms of reduction and the improved SVM (ISVM).

3.1 Rough Set Theory

The original Rough Set Theory was proposed by Pawlak [10][14]. This theory is concerned with analysis of deterministic data dependencies.

Information Systems. In the Rough Set Theory, information systems are used to represent knowledge. An information system $S = (U, A, V, f)$ consists of U which is a nonempty, finite set named universe, which is a set of objects, $U = \{x_1, x_2, \cdots, x_m\}$; A is a nonempty, finite set of attributes, $A = C \cup D$, in which C is the set of condition attributes, and D is the set of decision attributes; $V = \bigcup_{a \in A} Va$ is the domain of a; $f : U \times A \to V$ is an information function. For each $a \in A$ and $x \in U$, an information function $f(x, a) \in Va$ is defined, which means that for each object x in U, f specify its attribute value.

Lower and Upper Approximation. Due to imprecision which exists in the real world data, there are always conflicting objects contained in a decision table. Here conflicting objects refers to two or more objects that are undistinguishable by employing any set of condition attributes, but they belong to different decision classes. Such objects are called inconsistent. Such a decision table is called inconsistent decision table. In the rough set theory, the approximations of sets are introduced to deal with inconsistency. If $S = (U, A, V, f)$ is a decision table, suppose $B \subseteq A$, and $X \subseteq U$, then the *B-lower* and *B-upper* approximations of X are defined as:

$$\begin{aligned} \underline{B(X)} &= \bigcup\{Y \in U/IND(B) : Y \subseteq X\}, \\ \overline{B(X)} &= \bigcup\{Y \in U/IND(B) : Y \cap X \neq \emptyset\} \end{aligned} \tag{7}$$

Here, $U/IND(B)$ denotes the family of all equivalence classes of B; $IND(B) = \{(x, y) \in U \times U | \forall\ a \in B, f(x, a) = f(y, a)\}$ is the *B-indiscernibility relation.* $\underline{B(X)}$is the set of all elements of U which can be certainty classified as elements of X, employing the set of attributes B. The *Positive Region* of X is defined as:

$$POS_B(X) = \underline{B(X)} \tag{8}$$

$\overline{B(X)}$ is the set of elements of U which can be possibly classified as elements of X using the set of attributes B. The set $Bnd_B(X) = \overline{B(X)} - \underline{B(X)}$ is called the *B-boundary* of X. If $Bnd_B(X) = \emptyset$, then we say that X is definable on B; otherwise we say that X is non-definable on B, which is also named as *rough set.*

Attribute Reduction. An important issue in the Rough Set Theory is about attributes reduction. The process of finding a smaller set of attributes than original one with same classify ability as original set is called attribute reduction. *Core* is the intersection of all reductions. Given an information system S, for a given set of condition attributes $P \subseteq C$, we can define a positive region $POS_P(D) = \bigcup_{x \in U/D} \underline{P}X$, which contains all objects in U, which can be classified without error into distinct classes defined by $IND(D)$ based only on information in the $IND(P)$. Another important issue in data analysis is discovering dependencies between attributes. Let D and C be subsets of $A. D$ depends on C in degree as denoted in the following:

$$\gamma(C, D) = card(POS_C(D))/card(U),\ \gamma(C, D) \in [0, 1] \tag{9}$$

The set of attributes reduction is described as:

$$R = \{R : R \subseteq C, \gamma(R, D) = \gamma(C, D)\} \tag{10}$$

Thereby, the equality of the attributes dependency can be used as the end condition of iterative operation.

3.2 Attribute Reduction Algorithm

For a given decision information table $S' = (U, C \cup D, V, f)$, the subset $C' \subseteq C$ is the smallest reduction of C. If C' satisfy two conditions as below: (1).$POS_c(\gamma) = POS_{C'}(\gamma)$, (2).Not exist $C'' \subset C'$, so as to $POS_{c''}(\gamma) = POS_{C'}(\gamma)$. Based on the definition of attributes dependency, the importance of an attribute $a \in C - R$ can be defined as:

$$\theta(a, R, D) = \gamma(R \cup \{a\}, D) - \gamma(R, D) \tag{11}$$

where $R = \emptyset, \theta(a, D) = \gamma(\{a\}, D)$. Based on the hereinabove definition, we design the attributes reduction algorithm-Algorithm1.

```
**Algorithm1:** Reduce $(S', R)$-Attributes Reduction Algorithm.
**Input:** Decision information table $S' = (U, C \cup D, V, f)$
**Output:** An attribute reduction set $R$ of $S'$
1). $R = \emptyset$ ;
2).For every attribute $a_i \in C - R$ calculating its attribute importance $\theta(a_i, R, D)$;
3).Choosing the attribute $a_i$ which the value of $\theta(a_i, R, D)$ is the largest, and
   $R \Leftarrow R \cup \{a_i\}$;
4). If $\gamma(R, D) = \gamma(C, D)$ then goto 5) else goto 2);
5).Return $(R)$; // Return the attribute set $R$ which has been reduced.
```

Obviously,the complexity of the above algorithm is $O(m^2)$, m is the number of condition attributes in decision table S'.

3.3 The Algorithm of Improved Support Vector Machine(ISVM)

The ISVM algorithm is composed with two stages. Firstly, the condition attributes of the information set is reduced by running the reduction algorithm. Then, the reduced information set will be classified by the SVM classifier. The ISVM algorithm-Algorithm2 is as following:

```
**Algorithm2:** Improved Support Vector Machine-ISVM$(S, Y)$
**Input:** A decision information table $S = (U, C \cup D, V, f)$
**Output:** The classify result $Y$
1).Discrete$(S)$;// Discrete the decision information table $S$
2).Reduce $(S, R)$; // Running the reduction algorithm1, $R$ is the reduced
                     //condition attributes set.
3).$S = R \cup D$; //$S$ is a new information table which condition attributes has
                   //been reduced
4).SVM $(S, Y)$;// Executing SVM classifier. Its input is the new information
                //table $S$, and $Y$ is the classified result.
5).Return $(Y)$;
```

4 Data Pre-processing and Feature Extraction

This section summarizes the mammography collection and the techniques used to enhance the mammograms as well as the features that were extracted from images.

4.1 Mammography Collection

The data collection used in our experiments was taken from the Mammographic Image Anlysis Society (MIAS)[13]. We selected this dataset because it is freely available, and to be able to compare our method with other published work like [15], since it is a commonly used database for mammography categorization.

MIAS consists of 322 images, which belong to three big categories: normal, benign and malign. There are 208 normal images, 63 benign and 51 malign, which are considered abnormal. In addition, the abnormal cases are further divided in six categories: microcalcification, circumscribed masses, speculated masses, ill-defined masses, architectural distortion and asymmetry. All the images also include the locations of any abnormalities that may be present. The existing data in the collection consists of the location of the abnormality (like the center of a circle surrounding the tumor), its radius, breast position (right or left), type of breast tissues (fatty, fatty-glandular and dense) and tumor type if it exists (benign or malign). All the mammograms are medio-lateral oblique view.

4.2 Data Pre-processing

Pre-processing is always a necessity whenever the data to be mined is noisy, inconsistent or incomplete. Pre-processing significantly improves the effectiveness of data mining techniques [16].The type size of the images in MIAS is 1024x1024 and almost 50% of the whole image comprised the background with a lot of noise. In addition, these images are scanned at different illumination conditions, and therefore some images appeared too bright and some were too dark. The first step toward noise removal was pruning the images with a cropping operation.The second step was an image enhancement. Thus, we eliminated almost all the background information and most of the noise. An example of cropping that eliminates the artefacts and the black background is given in Figure 1 (a-b). Since the resulting images had different sizes, the x and the y coordinates were normalized to a value between 0 and 255. The cropping operation was done

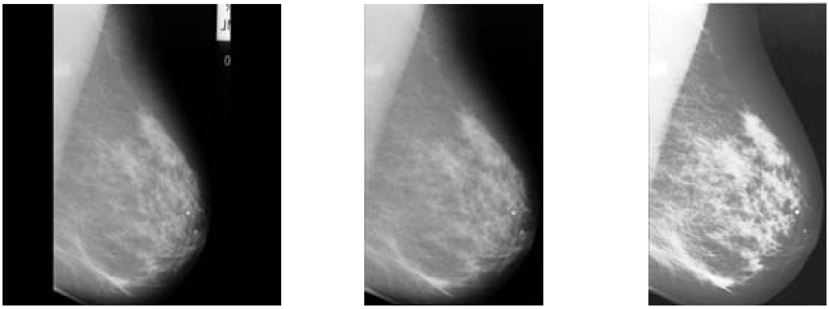

(a) original image
(b) cropped image
(c) enhanced image

Fig. 1. Pre-processing phase on an example image

automatically by sweeping horizontally through the image. Then we applied the Histogram Equalization method to enhance the image in order to diminish the effect of over-brightness or over-darkness in images. Histogram Equalization increases the contrast range in an image by increasing the dynamic range of grey levels [16]. Figure 1 (c) shows an example of histogram equalization result after cropping.

4.3 Feature Extraction

After pre-processing the images, features relevant to the classification are extracted from the cleaned images. The extracted features are organized in a database, which is the input for the mining phase of the classifier. This database is also constructed by merging some already existing features like the type of the tissue (dense, fatty and fatty-glandular) and the location of the abnormality (like the center of a circle surrounding the tumor). The extracted features are four statistical parameters: mean, variance, skewness and kurtosis. The formula for the statistical parameters computed is the following: $Mean$ is $\mu = \Sigma_{k=1}^{N} f_k p_f(f_k)$; $Variance$ is $\sigma^2 = \Sigma_{k=1}^{N} (f_k - \mu)^2 p_f(f_k)$; $Skewness$ is $\mu_3 = \frac{1}{\sigma^3} \Sigma_{k=1}^{N} (f_k - \mu)^3 p_f(f_k)$; $Kurtosis$ is $\mu_4 = \frac{1}{\sigma^4} \Sigma_{k=1}^{N} (f_k - \mu)^4 p_f(f_k)$. Where N denotes the number of gray levels in the mammogram, f_k is the kth gray level and $p_f(f_k) = \frac{n_k}{n}$, where n_k is the number of pixels with f_k gray level and n is the total number of pixels in the region.

All these extracted features are computed over smaller windows of the original image. The original image is first split in four parts. For a more accurate extraction of the features we split each of these four regions in other four parts. The statistical parameters were computed for each of the sixteen sub-parts of the original image [15]. After that, we get sixty-four statistical features.

5 Experimental Results

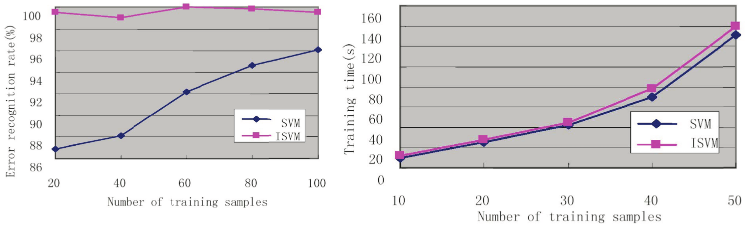
We used the 10 fold cross validation techniques to evaluate the algorithm performance. We divided the features database in ten splits. For each split we selected about 90% of the dataset for training and the rest for testing. That is 288 images in the training set and 34 images in the testing set. The features database is composed with the extracted features and the existing data of 322 images in MIAS. All the numeric attributes are discrete using algorithm DBChi2[17]. In the training phase, the ISVM was applied on the training data. Then, for an image in the testing set, the classification process searches in this ISVM for finding the class that is the closest to be attached with the object presented for categorization. At the same time, the number of choosing attributes is recorded. The SVM program is from LIBSVM[18].The experimental results is in table1.

Table 1. The comparison of SVM and ISVM algorithms on MIAS

Ten	SVM	ISVM	ISVM
splits	Accuracy(%)	Attributes number	Accuracy(%)
1	93.56	21	96.42
2	90.21	16	97.12
3	92.19	18	97.56
4	93.88	15	96.87
5	93.47	23	96.06
6	94.66	20	96.44
7	92.25	13	95.15
8	90.83	26	94.96
9	93.64	19	97.34
10	94.75	15	97.69
Average	92.94	18.6	96.56

Table 1 represents the comparison in terms of the choosing attributes number and classifying accuracy of the present algorithm ISVM and the algorithm SVM. The first column is the ten splits of MIAS. The second and the fourth columns are the classified accuracy of SVM and ISVM based on ten splits. The third volume is the number of choosing attributes of ISVM. At the bottom of the table, each column's average is shown. The table shows that the ISVM performs better than only SVM algorithm in terms of the classifying accuracy. At the same time, because the data set is reduced firstly, there are only 18.6 condition attributes averagely inputted to SVM classifier, which makes it easy for the SVM classifying.

To compare the capabilities of classification and the training time of SVM with ISVM on MIAS, we give the experiment results about error recognition rate and training time on small samples simultaneously. Figure 2(a) is the comparison of error recognition rate on different samples including 20,40,60,80 and 100 applied to train SVM and ISVM respectively. Figure 2(b) is the comparison of training time on training samples varies from 10 to 50. Figure 2(a) shows that error



(a) Error recognition rate comparison

(b) Training time comparison

Fig. 2. The experimental results comparison of SVM and ISVM

recognition rate of ISVM classifier is close to 100%, which is higher than SVM classifier obviously. When the training samples are smaller than 50, the error recognition rate of SVM can not reach to 90%. From Figure 2(b), the ISVM classifier needs more time than that of SVM classifier because of the attributes reduction stage of ISVM. But the training time of them is closed. Because of calculating with second, the distinction of training time is very little and can be ignored in practice.

6 Conclusions

In this paper, we have presented a hybrid classifier based on Rough Set Theory(RST) and Support Vector Machine(SVM) which is called Improved Support Vector Machine-ISVM. ISVM makes great use of the advantages of SVM's greater generalization performance and RST in effectively dealing with vagueness and uncertainty information. By data-analyzed method of RST, it can remove large amount of redundancy, and decrease volume of SVM training data. The preprocessing step enhances the efficiency of SVM in training and testing phases and strengthens classification and generation capabilities of SVM. Finally, ISVM was applied to medical image classification, and the evaluation of the ISVM was carried out on MIAS dataset. The experimental results show that the accuracy of the ISVM classifier can reach 96.56% than 92.94% which execute SVM classifier, and the error recognition rate values of ISVM tend to 100% in more than half the splits. There are some future research directions to be studied. To cooperate with medical staff would get more interesting results. In addition, the extraction of different features or a different database organization could lead to improved results.

References

1. Vapnik, V.N.: The nature of statistical learning theory. Springer, Heidelberg (1995)
2. Osareh, A., Mirmehdil, M., Thomas, B., Markham, R.: Comparative Exudate Classification Using Support Vector Machines and Neural Networks. In: Dohi, T., Kikinis, R. (eds.) MICCAI 2002. LNCS, vol. 2489, pp. 413–420. Springer, Heidelberg (2002)
3. Foody, G.M., Mathur, A.: A Relative Evaluation of Multiclass Image Classification by Support Vector Machines. IEEE Transactions on Geoscience and Remote Sensing 42(6), 1335–1343 (2004)
4. Ma, J.S., Theiler, J., Perkins, S.: Accurate On-line Support Vector Regression. Neural Computation 15(11), 2683–2703 (2003)
5. Ben-Hur, A., Horn, D., Siegelmann, H.T., Vapnik, V.: Support Vector Clustering. Journal of Machine Learning Research 2(2), 125–137 (2001)
6. Kim, K.J.: Financial Time Series Forecasting Using Support Vector Machines. Neuro-computing 55(1), 307–319 (2003)
7. Dibike, Y.B., Velickov, S., Solomatine, D.: Support Vector Machines: Review and Applications in Civil Engineering. In: Proc. of the 2nd Joint Workshop on Application of AI in Civil Engineering, pp. 215–218 (2000)

8. Wang, L.P. (ed.): Support Vector Machines: Theory and Application. Springer, Heidelberg (2005)
9. Schlkopf, B., Smola, A J.: Learning with Kernel-Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2002)
10. Pawlak, Z.W.: Rough Sets. International Journal of Information and Computer Science 11(5), 341–356 (1982)
11. Lin, T.Y.: Introduction to the Special Issue on Rough Sets. International Journal of Approximate Reasoning 15(4), 287–289 (1996)
12. Wang, G.Y.: Rough Set Theory and Knowledge Acquisition. Xi'an Jiaotong University Press, Xi'an (2001)
13. (2006-9) http://www.wiau.man.ac.uk/services/MIAS/MIASweb.html
14. Pawlak, Z.W.: Rough Sets and Intelligent Data Analysis. Information sciences (147), 1–12 (2002)
15. Antonie, M.-L., Zaiane, O.R., Coman, A.: Application of data mining techniques for medical image classification. In: Proc. of Second Intl. Workshop on Multimedia Data Mining (MDM/KDD'2001) in conjunction with Seventh ACM SIGKDD, San Francisco, pp. 94–101 (2001)
16. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Addison-Wesley, Reading (1993)
17. Hu, X., Cercone, N.: Data Mining Via Generalization, Discretization and Rough Set Feature Selection [J]. Knowledge and Information System: An International Journal, vol. 1(1) (1999)
18. Chang.C., Lin, C. (2001) LIBSVM http://www.csie.ntu.edu.tw/~cjlin/libsvm,2006-9

Searching for Metric Structure of Musical Files

Bozena Kostek, Jaroslaw Wojcik, and Piotr Szczuko

Gdansk University of Technology, Narutowicza 11/12,
80-952 Gdansk, Poland
jaroslaw.wojcik@intesk.pl, {bozenka, szczuko}@sound.eti.pg.gda.pl

Abstract. The aim of this paper is to compare the effectiveness of various computational intelligence approaches applied to the task of retrieving musical rhythm from musical symbolic files. The study presented in this paper describes how Artificial Neural Networks and Rough Sets can be used for searching the metric structure of musical files. The described approaches are based on examining physical attributes of sound that are most significant in determining the placement of a particular sound in the accented location of a musical piece. The results of the experiments show that the approach based solely on duration is sufficient enough to retrieve the metric structure of rhythm from musical files.

Keywords: Music Information Retrieval, Rhythm Retrieval, Metric Rhythm, Artificial Neural Networks, Rough Sets.

1 Introduction

Content-based methods of music retrieval are nowadays developed by researchers from the multimedia retrieval domain. Rhythm, which is an informative element of a piece, determines musical style and might be valuable in retrieving music on the basis of a musical genre. The most common classes of rhythm retrieval models are: rule-based, multiple-agents, multiple-oscillators and probabilistic. The rhythm retrieval methods can be classified within the context of what type of actions they take, i.e. whether they quantize musical data, or find the tempo of a piece (e.g. van Belle [2]), time signatures, positions of barlines, a metric structure or an entire hypermetric hierarchy. Rhythm finding systems very often rank the hypotheses of rhythm, basing on the sound salience function. Since scientists differ in opinions on the aspect of salience, the Authors carried out special experiments to solve the salience problem. A number of research studies are based on the theory published by Lerdahl & Jackendoff [8], who claim that such physical attributes of sounds as pitch (frequency), duration and velocity (amplitude) influence the rhythmical salience of sounds. Another approach, proposed by Rosenthal [12], ranks higher the hypotheses in which long sounds are placed in accented positions. In Dixon's [4] multiple-agent approach, two salience functions are proposed, combining duration, pitch and velocity. The first, is a linear combination of physical attributes, Dixon calls it an *additive function.* The other one is a *multiplicative function.* Dahl [3] notices that drummers play accented

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 774–783, 2007.

strokes with higher amplitude than unaccented ones. Parncutt, in his book [10], claims that lower sounds fall on the beat. In the review of Parncutt's book, Huron [5] notices that the high salience of low sounds is "neither an experimentally determined fact nor an established principle in musical practice". A duration-based hypothesis appears to predominate in rhythm-related works, but this approach seems to be based on intuition only. The experimental confirmation of this thesis – based on the Data Mining (DM) association rules and Artificial Neural Networks (ANNs) – can be found in former works by the Authors of this paper [6, 7] and also in the doctoral thesis of one of them [15]. The experiments employing rough sets, which are a subject of this paper, were performed in order to confirm results obtained from the DM and ANN approaches. Another reason was to verify if all three computational intelligence models applied to the salience problem, return similar findings, which may prove the correctness of these approaches.

2 Computational Intelligence Models in the Emulation of Human Perception

Computational Intelligence (CI) is a branch of *Artificial Intelligence*, which deals with the AI soft facets, i.e. programs behaving intelligently. The CI is understood in a number of ways, e.g. as a study of the design of intelligent agents or as a subbranch of AI, which aims "to use learning, adaptive, or evolutionary computation to create programs that are, in some sense, intelligent" [14]. Researchers are trying to classify the branches of CI to designate the ways in which CI methods help humans to discover how their perception works. However, this is a multi-facet task with numerous overlapping definitions, thus the map of this discipline is ambiguous. The domain of CI groups several approaches, the most common are: the Artificial Neural Networks (ANNs), Fuzzy Systems, Evolutionary Computation, Machine Learning including Data Mining, Soft Computing, Rough Sets, Bayesian Networks, Expert Systems and Intelligent Agents. Currently, in the age of CI people are trying to build machines emulating human behaviors, and one of such applications concerns rhythm perception. This paper presents an example of how to design and build an algorithm which is able to emulate human perception of rhythm. Two CI approaches, namely the ANNs and Rough Sets (RS), are used in the experiments aiming at the estimation of musical salience. The first of them, the ANN model, concerns processes, which are not entirely known, e.g. human perception of rhythm. The latter is the RS approach, introduced by Pawlak [9] and used by many researches in data discovery and intelligent management [11].

Since the applicability of ANNs in recognition was experimentally confirmed in a number of areas, neural networks are also used to estimate rhythmic salience of sounds. There exists a vast literature on ANNs, and for this reason only a brief introduction to this area is presented in this paper. A structure of an ANN usually employs the McCulloch-Pitts model, involving the modification of the neuron activation function, which is usually sigmoidal. All neurons are interconnected. Within the context of the neural network topology, ANNs can be classified as *feedforward* or *recurrent* networks, which are also called *feedback* networks. In the case of recurrent ANNs the connections between units form cycles, while in feedforward ANNs the information moves in only one direction, i.e. forward. The elements of a vector of object features constitute the values, which are fed to the input of an ANN. The type of data accepted at the input and/or returned at the output of an ANN is also a differentiating factor. The *quantitative* variable values are continuous by nature, and the *categorical* variables belong to a finite set (small, medium, big, large). The ANNs with continuous values at input are able to determine the degree of the membership to a certain class. The output of networks based on categorical variables may be Boolean, in which case the network decides whether an object belongs to a class or not. In the case of the salience problem the number of categorical output variables equals to two, and it is determined whether the sound is accented or not.

In their experiments the Authors examined whether a supervised categorical network such as Learning Vector Quantization (LVQ) is sufficient to resolve the salience problem. The classification task of the network was to recognize the sound as accented or not. LVQs are self-organizing networks with the ability to learn and detect the regularities and correlations at their input, and then to adapt their responses to that input. An LVQ network is trained in a supervised manner, it consists of the competitive and a linear layers. The first one classifies the input vectors into subclasses, and the latter transforms input vectors into target classes.

3 Organizing Experimental Database

The experiments proposed in this paper are conducted on a database of national anthems retrieved from the Internet. The format of files in the experimental database is symbolic (MIDI files). Sounds constituting melodies of anthems were included in the training and testing sets. Storing information about meter in the files is necessary to indicate accented sounds in a musical piece. This information, however, is optional in MIDI files, thus in the training stage of either ANN- or RS-based experiments, the information whether the sound is accented or not is always available. In a number of musical files retrieved from the Internet, the assigned meter is incorrect or there is no information about meter at all. This is why the correctness of meter was checked by inserting an additional simple drum track into the melody. The hits of the snare drum were inserted in the locations of the piece calculated with **Formula (1)**, where T is a period computed with the autocorrelation function, and i indicates subsequent hits of a snare drum.

$$i \cdot T, i = 0, 1, 2, \ldots \tag{1}$$

The Authors listened to the musical files with snare drum hits inserted, and rejected all the files where accented locations were indicated incorrectly. Also some anthems with changes in time signature could not be included in the training and testing sets, because this metric rhythm retrieval method deals with hypotheses based on rhythmic levels of a constant period. Usually the change in time signature results in changes in the period of a rhythmic level corresponding to the meter, and an example of such change might be from 3/4 into 4/4. Conversely, an example of a change in time signature which does not influence the correct indication of accented sounds could be from 2/4 into 4/4. Salience experiments presented in this paper are conducted on polyphonic MIDI tracks containing melodies of eighty national anthems, only sounds coming from the tracks constituting the melodies of anthems were included in the training and testing sets, overlapping sounds coming from the tracks other than melodic ones, were not included in the experimental sets.

For the purpose of the experiments the values of physical sound attributes were normalized and discretized with equal subrange method. Minimum and maximum values within the domain of each attribute are found. The whole range is then divided into $m$subranges with thresholds between the subranges**,** placed in the locations counted with aid of the **Formula (2)**.

$$MinValue + (MaxValue - MinValue) \cdot j/mforj = 0, 1, 2, \ldots m \qquad (2)$$

4 ANN Experiment

For the training phase, accented locations in each melody were found with methods described in the previous Section. One of the tested networks had three separate inputs – one for each physical attribute of sound (duration, frequency and amplitude - *DPV*). Three remaining networks had one input each. Each input took a different physical attribute of a given sound, namely *D* – duration, *P* – pitch (frequency) or *V* – velocity (amplitude). All attributes were from the range of 0 to 127. The network output was binary: 1 if the sound was accented, or 0 if it was not. Musical data were provided to the networks to train them to recognize accented sounds on the basis of physical attributes.

In this study LVQ network recognized a sound as 'accented' or 'not accented'. Since physical attributes are not the only features determining whether a sound is accented, some network answers may be incorrect. The network accuracy *NA* was formulated as the ratio of the number of accented sounds, which were correctly detected by the network, to the total number of accented sounds in a melody, as stated in **Formula (3)**.

NA = number of accented sounds correctly detected by the network / number of all accented sounds

(3)

Hazard accuracy *HA* is the ratio of the number of accents given by the network to the number of all sounds in a set, as stated in **Formula (4)**.

HA = number of accented sounds detected by the network / number of all sounds

(4)

The melodies of anthems were used to create 10 training/testing sets. Each set included 8 entire pieces. Each sound with an index divisible by 3 was assumed to be a training sound. The remaining sounds were treated as testing sounds. As a consequence, the testing set was twice as large as the training set. Accuracies in the datasets were averaged for each network separately. Evaluating a separate accuracy for each ANN allowed to compare their preciseness. Standard deviations were also calculated. Fractions equal to standard deviations were divided by average values. Such fractions help compare the stability of results. The lower the value of the fraction is, the more stable the results are. All results are shown on the right side of **Table 1**. A single accuracy value was assigned to each ANN. Standard deviations were also calculated and the resultant stability fraction equal to standard deviations divided by average values was presented

Table 1. Parameters of training and testing data and performance of ANNs

Set No.	Number of sounds			Acc/all [%]	NA/HA			
	All	Accented	Not accented		*D*	*P*	*V*	*DPV*
1	937	387	550	41	1.90	1.01	0.95	1.96
2	1173	386	787	33	2.28	0.89	1.23	2.19
3	1054	385	669	37	2.14	0.96	0.11	2.13
4	937	315	622	34	2.25	1.13	0.79	2.49
5	801	293	508	37	1.98	1.02	1.04	1.95
6	603	245	358	41	1.67	1.02	0.93	1.24
7	781	332	449	43	1.93	0.98	1.16	1.89
8	880	344	536	39	2.06	0.97	1.13	2.14
9	867	335	532	39	1.91	0.87	0.83	1.73
10	1767	509	1258	29	2.14	0.72	1.62	2.66
Avg.	**980**	**353**	**626**	**37**	**2.03**	**0.96**	**0.98**	**2.03**
StdDev	317	71	251	4	0.19	0.11	0.39	0.39
StdDev/Avg					0.09	0.12	0.40	0.19

The accuracy of finding accented sounds estimated for four networks can be seen in **Fig. 1**, the plots are drawn on the basis of the data from **Table 1**. There are three plots presenting the results of networks fed with one attribute only, and one plot for the network presented with all three physical attributes at its single input (line *DPV*). The consequent pairs of training and testing sets are on the horizontal axis, the fraction *NA/HA*, signifying how many times an approach is more accurate than a blind choice, is on the vertical axis.

5 Rough Sets Experiment

The aim of this experiment was to obtain the results analogical to the ones coming from the ANN and to confront them with each other. In particular, it was expected to confirm whether physical attributes influence a tendency of sounds

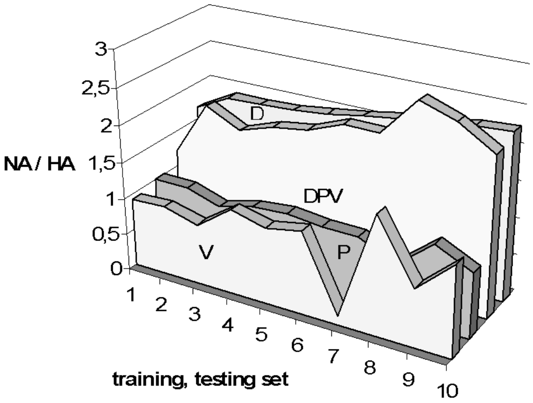


Fig. 1. Accuracy of four networks for melodies of anthems

to be located in accented positions. Further, it was to answer how complex is the way the rhythmic salience of sound depends on its physical attributes, and to observe the stability of the accuracies obtained in the RS-based experiment.

In the rough set-based experiment, the dataset named RSESdata1 was split into a training and testing sets in the 3:1 ratio. Then the rules were generated, utilizing a genetic algorithm available in the Rough Set Exploration System [1][13]. For dataset RSESdata1, 7859 rules were obtained resulting in the classification accuracy of 0.75 with the coverage equal to 1. Rules with support less than 10 were then removed. The set of rules was thus reduced to 427 and the accuracy dropped to 0.736 with the coverage still remaining 1. Then the next attempt to further decrease the number of rules was made, and rules with support less than 30 were excluded. In this case, 156 rules were still valid but the accuracy dropped significantly, i.e. to 0.707, and at the same time the coverage fall to 0.99. It was decided that for a practical implementation of a rough set-based classifier, a set of 427 rules is suitable. Reducts used in rule generation are presented in **Table 2**.

The same approach was used to dataset RSESdata2, and resulted in 11121 rules with the accuracy of 0.742 and the coverage of 1. After removing rules with support less than 10, only 384 rules remained, and the accuracy dropped

Table 2. Reduct for RSESdata1 dataset

Reducts	Positive Region	Stability Coefficient
{ duration, pitch }	0.460	1
{ duration, velocity }	0.565	1
{ pitch, velocity }	0.369	1
{ duration }	0.039	1
{ pitch }	0.002	1
{ velocity }	0.001	1

Table 3. Reduct for RSESdata2 dataset

Reducts	Positive Region	Stability Coefficient
{ duration, velocity }	0.6956	1
{ duration, pitch }	0.6671	1
{ pitch, velocity }	0.4758	1
{ duration }	0.0878	1
{ pitch }	0.0034	1
{ velocity }	0.0028	1

to 0.735. Again, such number of rules is practically applicable. Reducts used in rule generation are presented in **Table 3**.

The approach taken to LVQ network was also implemented for rough sets. Ten different training\test sets were acquired by randomly splitting data into five pairs, and than each set in a pair was further divided into two sets – a training and a testing one – with the 2:1 ratio. Therefore testing sets contained 1679 objects each. The experiments, however, were based on RSESdata1 set because of its higher generalization ability (see **Table 4**).

Table 4. Parameters of training and testing data and performance of RSES (RSA is a Rough Set factor, analogical to *NA* in ANNs)

Set No.	Number of sounds			Acc/all	RSA/HA			
	All testing sounds	Accented	Not accented		*D*	*P*	*V*	*DPV*
1	1679	610	1069	36.33	1.81	1.06	1.21	1.75
2	1679	608	1071	36.21	1.90	1.08	1.09	1.74
3	1679	594	1085	35.37	1.84	1.12	1.19	1.74
4	1679	638	1041	37.99	1.68	1.08	1.12	1.62
5	1679	632	1047	37.64	1.67	1.07	1.12	1.64
6	1679	605	1074	36.03	1.87	1.16	1.13	1.88
7	1679	573	1106	34.12	1.77	1.09	1.18	1.68
8	1679	618	1061	36.80	1.90	1.06	1.17	1.73
9	1679	603	1076	35.91	1.77	1.08	1.11	1.70
10	1679	627	1052	37.34	1.77	1.08	1.15	1.66
Avg.	**1679**	**610**	**1068**	**36.37**	**1.80**	**1.09**	**1.15**	**1.72**
StdDev	0	19.2	19.2	1.14	0.08	0.02	0.039	0.07
StdDev/Avg					0.04	0.02	0.033	0.04

It should be remembered that reduct is a set of attributes that discerns objects with different decisions. Positive region shows what part of indiscernibility classes for a reduct is inside the rough set. The larger boundary regions are, the more rules are nondeterministic, and the smaller positive region is. Stability coefficient reveals if the reduct appears also for subsets of original dataset, which are calculated during the reduct search. For reduct {duration} positive region is very small, but during classification a voting method is used to infer correct outcome from many nondeterministic rules, and, finally, high accuracy is obtained. Adding another dimension, e.g. {duration, velocity}, results in higher number of deterministic rules, larger positive region, but it does not guarantee the accuracy increase (**Table 4**).

Rules were generated utilizing different reduct sets (compare with **Table 1**): D - {duration} only; P - {pitch} only; V - {velocity} only; DPV - all 6 reducts {duration, velocity}, {duration, pitch}, {pitch, velocity}, {duration}, {pitch}, {velocity} have been employed.

6 Concluding Remarks

On the basis of the results (see **Tables 1**, **4**) obtained for both: RS and ANN experiments, it may be observed that the average accuracy of all approaches taking duration D into account – solely or in the combination of all three attributes DPV – is about twice as good as hazard accuracy (values of 1.72 for Rough Set DPV, 1.80 for Rough Set D, and a value of 2.03 both for Network D and for Network DPV were achieved). The performance of approaches considering pitch P and velocity V separately are very close to random accuracy, the values are equal to 1.09 and 1.15 for Rough Sets. For the ANN, the values are 0.96 and 0.98, respectively. Thus, it can be concluded that the location of a sound depends only on its duration.

The algorithms with the combination of DPV attributes performed as well as the one based only on duration, however this is especially valid for ANNs, rough sets did a little bit worse. Additional attributes do not increase the performance of the ANN approach. It can be thus concluded that the rhythmic salience depends on physical attributes in a simple way, namely it depends on a single physical attribute – duration.

Network D is the ANN that returns the most stable results. The value of fraction in the third row of **Table 1** is low for this network and it is equal to 0.09. Network DPV, which takes all attributes into account, is much less reliable because the stability fraction is about twice worse than the stability of Network D and it is equal to 0.19. The stability of Network P, considering the pitch, is quite high (it equals 0.12), but its performance is close to the random choice. For learning and testing data used in this experiment, velocity appeared to be the most data-sensitive attribute (see results of Network V). Additionally, this network appeared to be unable to find accented sounds.

In the case of Rough Sets, the duration-based approaches D and DPV returned less stable results than P and V approaches. Values of 0.045, 0.043, 0.026, 0.033 were obtained for D, DPV , P, and V respectively.

The ANN salience-based experiments described in the earlier work by the Authors [7], were conducted on a database of musical files containing various musical genres. It consisted of monophonic (non-polyphonic), and the polyphonic files. Also, a verification of the association rules model of the Data Mining domain for musical salience estimation was presented in that paper. The conclusions derived from the experiments conducted on national anthems for the purpose of this paper, are consistent with the ones described in the work by Kostek et al. [7]. Thus, the ANNs can be used in systems of musical rhythm retrieval in a wide range of genres and regardless of the fact whether the music is monophonic of polyphonic. The average relative accuracy for duration-based approaches where Rough Sets are used is lower than this obtained by LVQ ANNs. However, the same tendency is noticeable – utilization of the duration parameter leads to successful classification. The P (pitch) and V (velocity) parameters appeared not to be important in making decision about rhythmical structure of a melody. The Authors believe that using some other discretization schemes instead of the equal subranges technique could improve the accuracy of rough sets-based rhythm classification.

Music Information Retrieval (MIR) methods can be applied in practice. A composer creating a new melody can very often be interested, whether similar melody was not composed before, in music libraries and shops with music, customers seeking for music can sing or hum the song not knowing its author and a title. Computer programs employing MIR algorithms can also be helpful for musical professionals in the process of music composing and performing. The method presented in this paper is a step towards content-based retrieval of musical phrases, the advantage of this type of methods is that no human assistance is necessary to create text descriptors, indexing the musical piece. This method is helpful in retrieving the time signature and the locations of barlines from a piece on the basis of its content only. Rhythmic salience approach worked out and described in this paper is also valuable in ranking rhythmic hypotheses and music transcription. On the other hand, fully transcribed music can also be a subject of automatic analysis or retrieval. Transcription is also useful in applications of automatic harmonization – beginnings of musical phrases are locations where a harmonic chord changes, the endings indicate locations where a percussive fill-in by a drummer is usually performed. Other possible applications of metric rhythm retrieval method are systems for music recommendation, plagiarism detection, synchronization music with other elements of multimedia applications, support in creating musical scores basing on melody played on a MIDI instrument, retrieval on the basis of musical genre. Also the system, creating drum accompaniment to a given melody, automatically, on the basis of highly ranked rhythmic hypothesis is an useful practical application of rhythmic salience method. A prototype of such a system, using salience approach was developed on the basis of findings of authors of this paper.

Acknowledgments. This work was supported by the Ministry of Science and Education within the Grant No. 3 T11F02729.

References

1. Bazan, J.G., Szczuka, M.S.: The Rough Set Exploration System, Transactions on Rough Sets III. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 37–56. Springer, Heidelberg (2005)

2. van Belle, W.: BPM Measurement of Digital Audio by means of Beat Graphs & Ray Shooting. Department Computer Science, University Tromsø (2004) Retrieved from http://bio6.itek.norut.no/werner/Papers/bpm04/
3. Dahl, S.: On the beat - Human movement and timing in the production and perception of music. Ph. D. Thesis, KTH Royal Institute of Technology, Stockholm, Sweden (2005)
4. Dixon, S.: Automatic Extraction of Tempo and Beat from Expressive Performances. J. of New Music Research, Swets & Zeitlinger, 30(1), 39–58 (2001)
5. Huron, D.: Review of Harmony: A Psychoacoustical Approach (Parncutt, 1989). Psychology of Music 19(2), 219–222 (1991)
6. Kostek, B., Wójcik, J.: Machine Learning System for Estimation Rhythmic Salience of Sounds. Int. J. of Knowledge-Based and Intelligent Engineering Systems (2005)
7. Kostek, B., Wójcik, J., Holonowicz, P.: Estimation the Rhythmic Salience of Sound with Association Rules and Neural Networks. In: Proc. of the Intern. IIS: IIPWM'05 Conference, Intelligent Information Processing and Web Mining, Advances in Soft Computing, pp. 531–540. Springer, Heidelberg, 13.6.2005- 16.6.2005, Gdansk-Sobieszewo (2005)
8. Lerdahl, F., Jackendoff, R.: A Generative Theory of Tonal Music. MIT Press, Cambridge (1983)
9. Pawlak, Z.: Rough Sets. International J Computer and Information Sciences, 11 (1982)
10. Parncutt, R.: Harmony: A Psychoacoustical Approach. Springer, Heidelberg (1989)
11. Peters, J.F., Skowron, A.: Transactions on Rough Sets V. LNCS, vol. 4100. Springer, Heidelberg (2004-2006)
12. Rosenthal, D.F.: Emulation of human rhythm perception. Comp. Music J. 16(1), 64–76 (1992)
13. RSES Homepage, http://logic.mimuw.edu.pl/~rses
14. Wikipedia homepage
15. Wójcik, J.: Methods of Forming and Ranking Rhythmic Hypotheses in Musical Pieces. Ph.D. Thesis, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Gdansk (2007)

Parameter-Based Categorization for Musical Instrument Retrieval

Rory Lewis[1] and Alicja Wieczorkowska[2]

[1] University of North Carolina, 9201 University City Blvd. Charlotte, NC 28223, USA
[2] Polish-Japanese Institute of Information Technology, Koszykowa 86, 02-008 Warsaw, Poland

Abstract. In the continuing goal of codifying the classification of musical sounds and extracting rules for data mining, we present the following methodology of categorization, based on numerical parameters. The motivation for this paper is based upon the fallibility of Hornbostel and Sachs generic classification scheme, used in Music Information Retrieval for instruments. In eliminating the redundancy and discrepancies of Hornbostel and Sachs' classification of musical sounds we present a procedure that draws categorization from numerical attributes, describing both time domain and spectrum of sound. Rather than using classification based directly on Hornbostel and Sachs scheme, we rely on the empirical data describing the log attack, sustainability and harmonicity. We propose a categorization system based upon the empirical musical parameters and then incorporating the resultant structure for classification rules.

1 Instrument Classification

Information retrieval of musical instruments and their sounds has invoked a need to constructive cataloguing conventions with specialized vocabularies and other encoding schemes. For example the Library of Congress subject headings [1] and the German Schlagwortnormdatei Decimal Classification both use the Dewey classification system [3,11] In 1914 Hornbostel-Sachs devised a classification system, based on the Dewey decimal classification which essentially classified all instruments into strings, wind and percussion. Later it went further and broke instruments into four categories:

1.1 Idiophones, where sound is produced by vibration of the body of the instrument
2.2 Membranophones, where sound produced by the vibration of a membrane
3.3 Chordophones, where sound is produced by the vibration of strings
4.4 Aerophones, where sound is produced by vibrating air.

For purposes of music information retrieval, the Hornbostel-Sachs cataloguing convention is problematic, since it contains exceptions, i.e. instruments that could fall into a few categories. This convention is based on what element vibrates to produce sound (air, string, membrane, or elastic solid body), and playing method, shape, relationship of parts of the instrument and so on. Since

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 784–792, 2007.

this classification follows a humanistic conventions, it makes it incompatible for a knowledge discovery discourse. For example, a piano emits sound when the hammer strikes strings. For many musicians, especially playing jazz, the piano is considered percussive, yet its the string that emits the sound vibrations, so it is classifies as a chordophone, according to Sachs and Hornbostel scheme. Also, the tamborine comprises a membrane and bells making it both an membranophone and an idiophone. Considering this, our paper presents a basis for an empirical music instrument classification system conducive for music information retrieval, specifically for automatic indexing of music instruments.

2 A Three-Level Empirical Tree

We focus on three properties of sound waves that can be calculated for any sound and can differentiate. They are: log-attack, harmonicity and sustainability. The first two properties are part of the set of descriptors for audio content description provided in the MPEG-7 standard and have aided us in musical instrument timbre description, audio signature and sound description [16]. The third one is based on observations of sound envelopes for singular sound of various instruments and for various playing method, i.e. articulation.

2.1 LogAttackTime (LAT)

The motivation for using the MPEG-7 temporal descriptor, LogAttackTime (LAT), is because segments containing short LAT periods cut generic percussive (and also sounds of plucked or hammered string) and harmonic (sustained) signals into two separate groups [6,7]. The *attack* of a sound is the first part of a sound, before a real note develops where the LAT is the logarithm of the time duration between the point where the signal starts to the point it reaches its stable part.[12] The range of the LAT is defined as $log_{10}(\frac{1}{samplingrate})$ and is determined by the length of the signal. Struck instruments, such a most percussive instruments have a short LAT whereas blown or vibrated instruments contain LATs of a longer duration.

$$LAT = log_{10}(T1 - T0), \tag{1}$$

where $T0$ is the time the signal starts; and $T1$ is reaches its sustained part (harmonic space) or maximum part (percussive space).

2.2 AudioHarmonicityType (HRM)

The motivation for using the MPEG-7 descriptor, AudioHarmonicityType is that it describes the degree of harmonicity of an audio signal.[7] Most ”percussive” instruments contain a latent indefinite pitch that confuses and causes exceptions to parameters set forth in Hornbostel-Sachs. Furthermore, some percussive instruments such as a cuica or guido contain a weak LogAttackTime and therefore fall

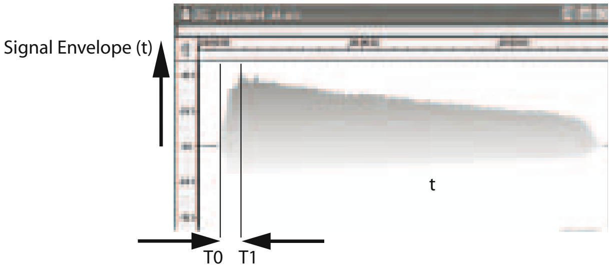


Fig. 1. Illustration of log-attack time. *T0* can be estimated as the time the signal envelope exceeds .02 of its maximum value. *T1* can be estimated, simply, as the time the signal envelope reaches its maximum value.

into non-percussive cluster while still maintaining an indefinite pitch (although, we can perceive differences in contents of low and high frequencies in percussive sounds as well). The use of the descriptor AudioHarmonicityType theoretically should solve this issue. It includes the weighted confidence measure, SeriesOfScalarType that handles portions of signal that lack clear periodicity. AudioHarmonicity combines the ratio of harmonic power to total power: HarmonicRatio, and the frequency of the inharmonic spectrum: UpperLimitOfHarmonicity.

First: We make the Harmonic Ratio *H(i)* the maximum *r(i,k)* in each frame, i where a definitive periodic signal for *H(i)* =1 and conversely white noise = 0.

$$H(i) = max\ r(i,k) \tag{2}$$

where *r(i,k)* is the normalised cross correlation of frame i with lag k:

$$r(i,k) = \sum_{j=m}^{m+n-1} s(j)\, s(j-k) \Bigg/ \left(\sum_{j=m}^{m+n-1} s(j)^2 * \sum_{j=m}^{m+n-1} s(j-k)^2 \right)^{\frac{1}{2}} \tag{3}$$

where s is the audio signal, m=i*n, where i=0, $M-1$=frame index and M = the number of frames, n=t*sr, where t = window size (10ms) and sr = sampling rate, k=*1*, K=*lag*, where K=ω*sr, ω = maximum fundamental period expected (40ms)

Second: Upon obtaining the i) DFTs of $s(j)$ and comb-filtered signals $c(j)$ in the AudioSpectrumEnvelope and ii) the power spectra $p(f)$ and $p'(f)$ in the AudioSpectrumCentroid we take the ratio f_{lim} and calculate the sum of power beyond the frequency for both $s(j)$ and $c(j)$:

$$a(f_{lim}) = \sum_{f=f_{lim}}^{f_{max}} p'(f) \Bigg/ \sum_{f=f_{lim}}^{f_{max}} p(f) \tag{4}$$

where f_{max} is the maximum frequency of the DFT.

Third: Starting where $f_{lim} = f_{max}$ we move down in frequency and stop where the greatest frequency, f_{ulim}'s ratio is smaller than 0.5 and convert it to an octave scale based on 1 kHz:

$$UpperLimitOfHarmonicity = log2(f_{ulim}/1000) \tag{5}$$

2.3 Sustainability (S)

We define sustainability into 5 categories based on the degree of dampening or sustainability the instrument can maintain over a maximum period of 7 seconds. For example, a flutist, horn player and violinist can maintain a singular note for more than 7 seconds therefore they receive a 1. Conversely a plucked guitar or single drum note typically cannot sustain that one sound for more than 7 seconds. It is true that a piano with pedal could maintain a sound after ten seconds but the sustainability factor would be present.

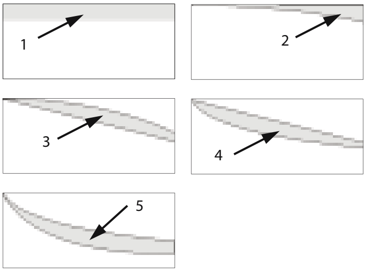


Fig. 2. Five levels of sustainability to severe dampening

3 Experiments

The sound data consists of a sample set of 156 signals extracted from our online database at http://www.mir.uncc.edu which contains 6,300 segmented sounds mostly from MUMS audio CD's that contain samples of broad range of musical instruments, including orchestral ones, piano, jazz instruments, organ, etc. [10] These CD's are widely used in musical instrument sound research [2,9,15,5,8,4], so they can be considered as a standard. The database consists of 188 samples each representing just one sample from group that make up the 6,300 files in the database. Mums divides the database into the following 18 classes: violin vibrato, violin pizzicato, viola vibrato, viola pizzicato, cello vibrato, cello pizzicato, double bass vibrato, double bass vibrato, double bass pizzicato, flute, oboe, b-flat clarinet, trumpet, trumpet muted, trombone, trombone muted, French horn, French horn muted, and tuba. Preprocessing these groups is not a part of rough set theory because rough sets require that input data process the rough sets. Rough set are objective with respect to its data. Here we discretize, using MPEG-7 classifiers as the experts. This is the point of the paper, we show a novel, empirical methodology of dividing sounds conducive to automatic retrieval of music.

4 Testing

The principle objective of our testing is to prove how parameter-based classification differs, and when used on Sachs-Hornbostel - improves Sachs-Hornbostel. Our parameters are machine-based, based on MPEG-7 and the temporal signal dampening. It is not based upon humanistic intuitiveness. We first prove that our attributes divide instruments into groups. Next we prove that our objects, which are in leaves for a given class, actually represent another class. This will show how parameter-based classification differs from and improves Sachs-Hornbostel. To induce the classification rules in the form of decision trees from a set of given examples we used Quinlan's C4.5 algorithm. [13] The algorithm constructs a decision tree to form production rules from an unpruned tree. Next a decision tree interpreter classifies items which produces the rules. We used Bratko's Orange software [14] and implement C4.5 with scripting in Python.

4.1 HRM, LAT, S, with HS01

The first test comprised the testing of the decision attribute Sachs-Hornbostel-level-1 against our two MPEG-7 descriptors, Harmonicity (HRM), Log Attack (LAT) and our temporal feature Sustainability (S). The Sachs-Hornbostel-level-1 attribute consists of four classes based upon human intuitiveness: aerophones, idiophones, chordophones and membranophones. See Appendix Figure 3

4.2 HRM, LAT, S, with HS02

The second test comprised the testing of the decision attribute Sachs-Hornbostel-level-2 against the HRM, LAT and S descriptors. The Sachs-Hornbostel-level-2 attribute consists of four classes: aerophones, idiophones, chordophones and membranophones. See Appendix Figure 4

4.3 HRM, LAT, S, with Instruments

The third test comprised the testing of the decision attribute instruments against the HRM, LAT and S descriptors. The Instrument attribute consists of four classes that describe instruments in the manner machines look at their signals: percussion, blown, string and struck Harmonics. See Appendix Figure 5

4.4 Resulting Tree

The resulting tree shows how the sound objects are grouped, and we can compare how this classification differs from Sachs-Hornbostel system. The misclassified objects show discrepancies between the Sachs-Hornbostel system, and sound properties described by physical attributes. The novelty of this methodology is that adding the temporal feature and grouping the instruments from the machines point of view have lead to 83% correctness. We have 26 more MPEG-7 descriptors to use with this methodology to breakdown the 17% misclassified

5 Summary and Conclusion

The idea and experiments presented in this paper show how musical instrument sounds can be classified according to physical properties of sounds, described by numerical parameters. The differences between obtained classification and Sachs-Hornbostel classification system show how ambiguous sounds, representing instruments played with various articulation, can be unambiguously classified.

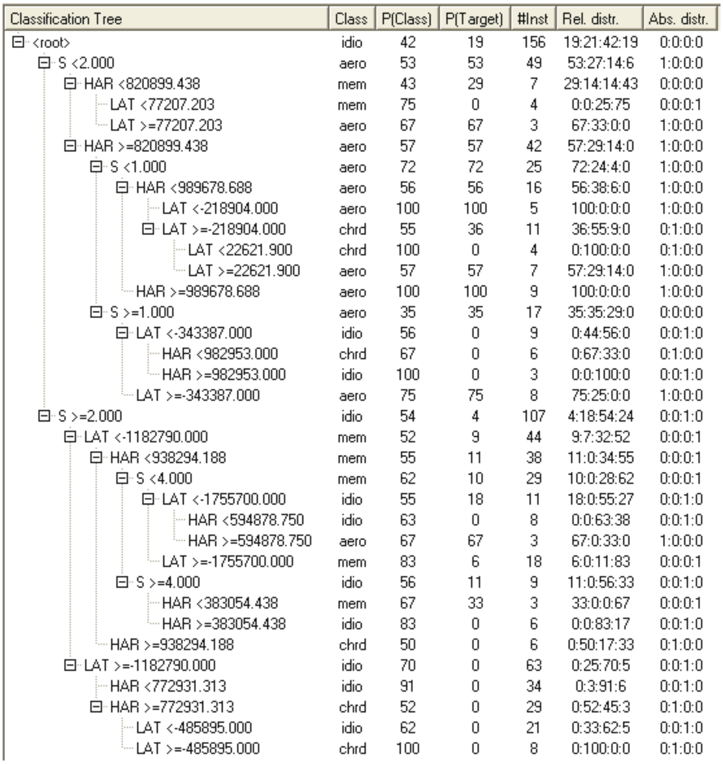
We plan to continue our experiments, using more of our MPEG-7 features and applying clustering algorithms in order to find probably better classification scheme for musical instrument sounds.

References

1. Brenne, M.: Storage and retrieval of musical documents in a FRBR-based library catalogue: Thesis, Oslo University College Faculty of journalism, library and information science (2004)
2. Cosi, P., De Poli, G., Lauzzana, G.: Auditory Modelling and Self-Organizing Neural Networks for Timbre Classification. Journal of New Music Research 23, 71–98 (1994)
3. Doerr, M.: Semantic Problems of Thesaurus Mapping. Journal of Digital Information 1(8), Article No. 52, 2001-03-26, 2001–03 (2001)
4. Eronen, A., Klapuri, A.: Musical Instrument Recognition Using Cepstral Coefficients and Temporal Features. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2000, Plymouth, MA, pp. 753–756 (2000)
5. Fujinaga, I., McMillan, K.: Realtime recognition of orchestral instruments. In: Proceedings of the International Computer Music Conference, pp. 141–143 (2000)
6. Gomez, E., Gouyon, F., Herrera, P., Amatriain, X.: Using and enhancing the current MPEG-7 standard for a music content processing tool. In: Proceedings of the 114th Audio Engineering Society Convention, March 2003, The Netherlands, Amsterdam (2003)
7. Information Technology — Multimedia Content Description Interface — Part 4: Audio. ISO/IEC JTC 1/SC 29, Date: 2001-06-9. ISO/IEC FDIS 15938-4:2001(E) ISO/IEC J/TC 1/SC 29/WG 11 Secretariat: ANSI (2001)
8. Kaminskyj, I.: Multi-feature Musical Instrument Classifier. MikroPolyphonie, 6 (2000) (online journal at http://farben.latrobe.edu.au/)
9. Martin, K.D., Kim, Y.E.: 2pMU9. Musical instrument identification: A pattern-recognition approach. 136-th meeting of the Acoustical Soc. of America, Norfolk, VA (1998)
10. Opolko, F., Wapnick, J.: MUMS – McGill University Master Samples. CD's (1987)
11. Patel, M., Koch, T., Doerr, M., Tsinaraki, C.: Semantic Interoperability in Digital Library Systems. IST-2002-2.3.1.12 Technology-enhanced Learning and Access to Cultural Heritage. UKOLN, University of Bath (2005)
12. Peeters, G., McAdams, S., Herrera, P.: Instrument sound description in the context of MPEG-7: In: Proceedings of the International Computer Music Conference (ICMC'00), Berlin, Germany (2000)
13. Quinlan, J.R.: 2pMU9. Bagging, boosting, and C4. 5. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence, vol. 725, p. 730 (1996)

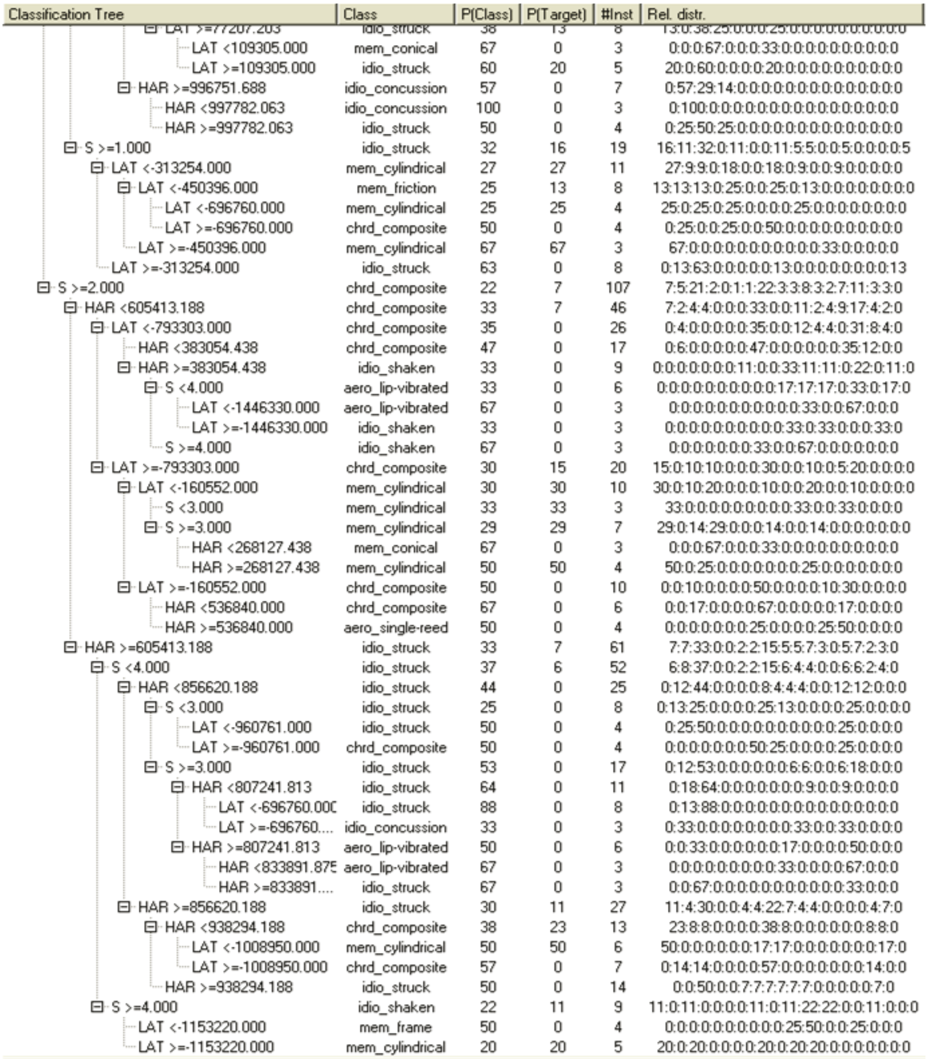
14. Demsar, J., Zupan, B., Leban, G.: http://www.ailab.si/orange
15. Wieczorkowska, A.: Rough Sets as a Tool for Audio Signal Classification. In: Raś, Z.W., Skowron, A. (eds.) Foundations of Intelligent Systems. LNCS, vol. 1609, pp. 367–375. Springer, Heidelberg (1999)
16. Wieczorkowska, A., Wróblewski, J., Synak, P., Ślęzak, D.: Application of temporal descriptors to musical instrument sound recognition. In: Proceedings of the International Computer Music Conference (ICMC'04), Berlin, Germany (2004)

Appendix



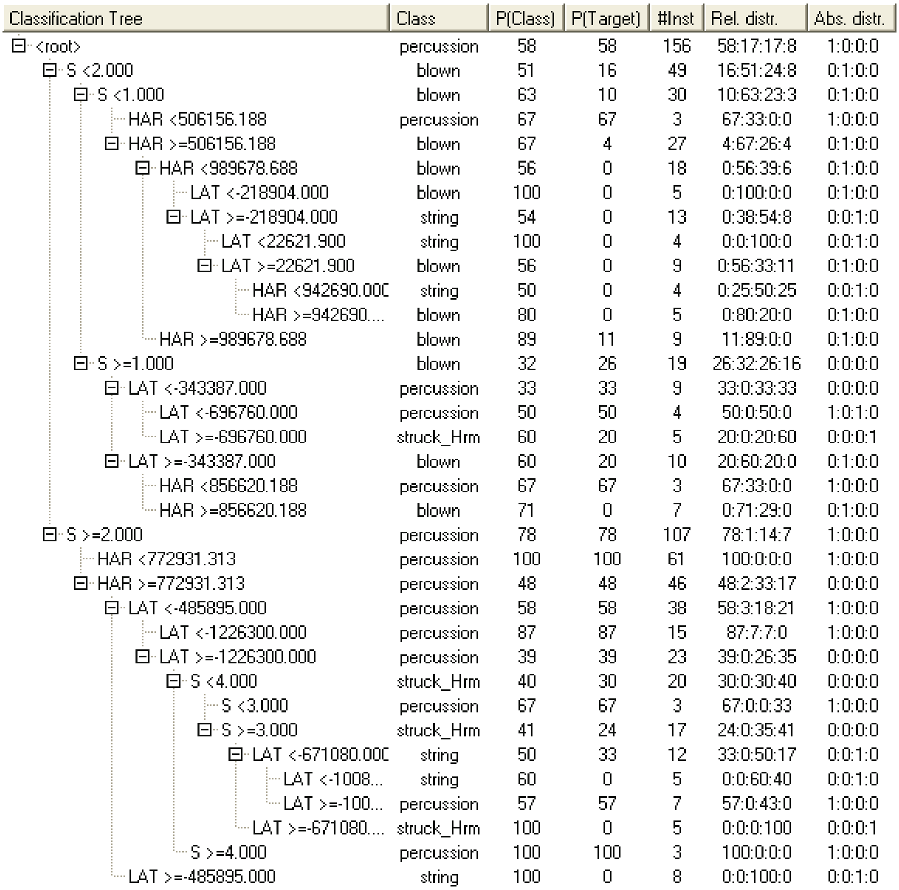
Classification Tree	Class	P(Class)	P(Target)	#Inst	Rel. distr.	Abs. distr.
⊟ <root>	idio	42	19	156	19:21:42:19	0:0:0:0
⊟ S <2.000	aero	53	53	49	53:27:14:6	1:0:0:0
⊟ HAR <820899.438	mem	43	29	7	29:14:14:43	0:0:0:0
LAT <77207.203	mem	75	0	4	0:0:25:75	0:0:0:1
LAT >=77207.203	aero	67	67	3	67:33:0:0	1:0:0:0
⊟ HAR >=820899.438	aero	57	57	42	57:29:14:0	1:0:0:0
⊟ S <1.000	aero	72	72	25	72:24:4:0	1:0:0:0
⊟ HAR <989678.688	aero	56	56	16	56:38:6:0	1:0:0:0
LAT <-218904.000	aero	100	100	5	100:0:0:0	1:0:0:0
⊟ LAT >=-218904.000	chrd	55	36	11	36:55:9:0	0:1:0:0
LAT <22621.900	chrd	100	0	4	0:100:0:0	0:1:0:0
LAT >=22621.900	aero	57	57	7	57:29:14:0	1:0:0:0
HAR >=989678.688	aero	100	100	9	100:0:0:0	1:0:0:0
⊟ S >=1.000	aero	35	35	17	35:35:29:0	0:0:0:0
⊟ LAT <-343387.000	idio	56	0	9	0:44:56:0	0:0:1:0
HAR <982953.000	chrd	67	0	6	0:67:33:0	0:1:0:0
HAR >=982953.000	idio	100	0	3	0:0:100:0	0:0:1:0
LAT >=-343387.000	aero	75	75	8	75:25:0:0	1:0:0:0
⊟ S >=2.000	idio	54	4	107	4:18:54:24	0:0:1:0
⊟ LAT <-1182790.000	mem	52	9	44	9:7:32:52	0:0:0:1
⊟ HAR <938294.188	mem	55	11	38	11:0:34:55	0:0:0:1
⊟ S <4.000	mem	62	10	29	10:0:28:62	0:0:0:1
⊟ LAT <-1755700.000	idio	55	18	11	18:0:55:27	0:0:1:0
HAR <594878.750	idio	63	0	8	0:0:63:38	0:0:1:0
HAR >=594878.750	aero	67	67	3	67:0:33:0	1:0:0:0
LAT >=-1755700.000	mem	83	6	18	6:0:11:83	0:0:0:1
⊟ S >=4.000	idio	56	11	9	11:0:56:33	0:0:1:0
HAR <383054.438	mem	67	33	3	33:0:0:67	0:0:0:1
HAR >=383054.438	idio	83	0	6	0:0:83:17	0:0:1:0
HAR >=938294.188	chrd	50	0	6	0:50:17:33	0:1:0:0
⊟ LAT >=-1182790.000	idio	70	0	63	0:25:70:5	0:0:1:0
HAR <772931.313	idio	91	0	34	0:3:91:6	0:0:1:0
⊟ HAR >=772931.313	chrd	52	0	29	0:52:45:3	0:1:0:0
LAT <-485895.000	idio	62	0	21	0:33:62:5	0:0:1:0
LAT >=-485895.000	chrd	100	0	8	0:100:0:0	0:1:0:0

Fig. 3. C4.5 results testing the decision attribute Sachs-Hornbostel-level-1 against our two MPEG-7 descriptors, Harmonicity (HRM), Log Attack (LAT) and our temporal feature Sustainability (S). S is divided at the ¡2.000 and ¿=2.000 node, Harmonicity is divided at ¡820889 and ¿=820889 for S ¡2.000 whereas, at ¿=2.000 LAT cuts the tree at LAT ¡-1182790 and ¿=1182790.



```
Classification Tree                Class               P(Class)  P(Target)  #Inst  Rel. distr.
        ⊟ LAT >=77207.203          idio_struck            38        13        8    13:0:38:25:0:0:0:25:0:0:0:0:0:0:0:0:0
            LAT <109305.000        mem_conical            67         0        3    0:0:67:0:0:0:33:0:0:0:0:0:0:0:0:0:0
            LAT >=109305.000       idio_struck            60        20        5    20:0:60:0:0:0:0:20:0:0:0:0:0:0:0:0:0
      ⊟ HAR >=996751.688           idio_concussion        57         0        7    0:57:29:14:0:0:0:0:0:0:0:0:0:0:0:0:0
          HAR <997782.063          idio_concussion       100         0        3    0:100:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0
          HAR >=997782.063         idio_struck            50         0        4    0:25:50:25:0:0:0:0:0:0:0:0:0:0:0:0:0
  ⊟ S >=1.000                      idio_struck            32        16       19    16:11:32:0:11:0:0:11:5:5:0:0:5:0:0:0:0:5
    ⊟ LAT <-313254.000             mem_cylindrical        27        27       11    27:9:9:0:18:0:0:18:0:9:0:0:9:0:0:0:0:0
      ⊟ LAT <-450396.000           mem_friction           25        13        8    13:13:13:0:25:0:0:25:0:13:0:0:0:0:0:0:0
          LAT <-696760.000         mem_cylindrical        25        25        4    25:0:25:0:25:0:0:0:0:25:0:0:0:0:0:0:0
          LAT >=-696760.000        chrd_composite         50         0        4    0:25:0:0:25:0:0:50:0:0:0:0:0:0:0:0:0
        LAT >=-450396.000          mem_cylindrical        67        67        3    67:0:0:0:0:0:0:0:0:0:0:0:33:0:0:0:0
      LAT >=-313254.000            idio_struck            63         0        8    0:13:63:0:0:0:0:0:13:0:0:0:0:0:0:0:13
⊟ S >=2.000                        chrd_composite         22         7      107    7:5:21:2:0:1:1:22:3:3:8:3:2:7:11:3:3:0
  ⊟ HAR <605413.188                chrd_composite         33         7       46    7:2:4:4:0:0:0:33:0:0:11:2:4:9:17:4:2:0
    ⊟ LAT <-793303.000             chrd_composite         35         0       26    0:4:0:0:0:0:0:35:0:0:12:4:4:0:31:8:4:0
        HAR <383054.438            chrd_composite         47         0       17    0:6:0:0:0:0:0:47:0:0:0:0:0:0:35:12:0:0
      ⊟ HAR >=383054.438           idio_shaken            33         0        9    0:0:0:0:0:0:0:11:0:0:33:11:11:0:22:0:11:0
        ⊟ S <4.000                 aero_lip-vibrated      33         0        6    0:0:0:0:0:0:0:0:0:0:17:17:17:0:33:0:17:0
            LAT <-1446330.000      aero_lip-vibrated      67         0        3    0:0:0:0:0:0:0:0:0:0:0:0:33:0:0:67:0:0:0
            LAT >=-1446330.000     idio_shaken            33         0        3    0:0:0:0:0:0:0:0:0:0:0:33:0:33:0:0:0:33:0
          S >=4.000                idio_shaken            67         0        3    0:0:0:0:0:0:0:33:0:0:67:0:0:0:0:0:0:0
    ⊟ LAT >=-793303.000            chrd_composite         30        15       20    15:0:10:10:0:0:0:30:0:0:10:0:5:20:0:0:0:0
      ⊟ LAT <-160552.000           mem_cylindrical        30        30       10    30:0:10:20:0:0:0:10:0:0:20:0:0:10:0:0:0:0
          S <3.000                 mem_cylindrical        33        33        3    33:0:0:0:0:0:0:0:0:0:33:0:0:33:0:0:0:0
        ⊟ S >=3.000                mem_cylindrical        29        29        7    29:0:14:29:0:0:0:14:0:0:14:0:0:0:0:0:0:0
            HAR <268127.438        mem_conical            67         0        3    0:0:0:67:0:0:0:33:0:0:0:0:0:0:0:0:0:0
            HAR >=268127.438       mem_cylindrical        50        50        4    50:0:25:0:0:0:0:0:0:0:25:0:0:0:0:0:0:0
      ⊟ LAT >=-160552.000          chrd_composite         50         0       10    0:0:10:0:0:0:0:50:0:0:0:0:10:30:0:0:0:0
          HAR <536840.000          chrd_composite         67         0        6    0:0:17:0:0:0:0:67:0:0:0:0:0:17:0:0:0:0
          HAR >=536840.000         aero_single-reed       50         0        4    0:0:0:0:0:0:0:25:0:0:0:0:25:50:0:0:0:0
  ⊟ HAR >=605413.188               idio_struck            33         7       61    7:7:33:0:0:2:2:15:5:5:7:3:0:5:7:2:3:0
    ⊟ S <4.000                     idio_struck            37         6       52    6:8:37:0:0:2:2:15:6:4:4:0:0:6:6:2:4:0
      ⊟ HAR <856620.188            idio_struck            44         0       25    0:12:44:0:0:0:0:8:4:4:4:0:0:12:12:0:0:0
        ⊟ S <3.000                 idio_struck            25         0        8    0:13:25:0:0:0:0:25:13:0:0:0:0:25:0:0:0:0
            LAT <-960761.000       idio_struck            50         0        4    0:25:50:0:0:0:0:0:0:0:0:0:0:25:0:0:0:0
            LAT >=-960761.000      chrd_composite         50         0        4    0:0:0:0:0:0:0:50:25:0:0:0:0:25:0:0:0:0
        ⊟ S >=3.000                idio_struck            53         0       17    0:12:53:0:0:0:0:0:0:6:6:0:0:6:18:0:0:0
          ⊟ HAR <807241.813        idio_struck            64         0       11    0:18:64:0:0:0:0:0:0:0:9:0:0:9:0:0:0:0
              LAT <-696760.000     idio_struck            88         0        8    0:13:88:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0
              LAT >=-696760....    idio_concussion        33         0        3    0:33:0:0:0:0:0:0:0:0:33:0:0:33:0:0:0:0
          ⊟ HAR >=807241.813       aero_lip-vibrated      50         0        6    0:0:33:0:0:0:0:0:0:17:0:0:0:0:50:0:0:0
              HAR <833891.875      aero_lip-vibrated      67         0        3    0:0:0:0:0:0:0:0:0:33:0:0:0:0:67:0:0:0
              HAR >=833891....     idio_struck            67         0        3    0:0:67:0:0:0:0:0:0:0:0:0:0:0:33:0:0:0
      ⊟ HAR >=856620.188           idio_struck            30        11       27    11:4:30:0:0:4:4:22:7:4:4:0:0:0:0:4:7:0
        ⊟ HAR <938294.188          chrd_composite         38        23       13    23:8:8:0:0:0:0:38:8:0:0:0:0:0:0:8:8:0
            LAT <-1008950.000      mem_cylindrical        50        50        6    50:0:0:0:0:0:0:17:17:0:0:0:0:0:0:0:17:0
            LAT >=-1008950.000     chrd_composite         57         0        7    0:14:14:0:0:0:0:57:0:0:0:0:0:0:0:14:0:0
          HAR >=938294.188         idio_struck            50         0       14    0:0:50:0:0:7:7:7:7:7:7:0:0:0:0:0:7:0
    ⊟ S >=4.000                    idio_shaken            22        11        9    11:0:11:0:0:0:0:11:0:11:22:22:0:0:11:0:0:0
        LAT <-1153220.000          mem_frame              50         0        4    0:0:0:0:0:0:0:0:0:0:0:25:50:0:0:25:0:0:0
        LAT >=-1153220.000         mem_cylindrical        20        20        5    20:0:20:0:0:0:0:20:0:20:20:0:0:0:0:0:0:0
```

Fig. 4. C4.5 results testing the decision attribute Sachs-Hornbostel-level-2 against our two MPEG-7 descriptors, Harmonicity (HRM), Log Attack (LAT) and our temporal feature Sustainability (S)



Classification Tree	Class	P(Class)	P(Target)	#Inst	Rel. distr.	Abs. distr.
⊟ <root>	percussion	58	58	156	58:17:17:8	1:0:0:0
⊟ S <2.000	blown	51	16	49	16:51:24:8	0:1:0:0
⊟ S <1.000	blown	63	10	30	10:63:23:3	0:1:0:0
HAR <506156.188	percussion	67	67	3	67:33:0:0	1:0:0:0
⊟ HAR >=506156.188	blown	67	4	27	4:67:26:4	0:1:0:0
⊟ HAR <989678.688	blown	56	0	18	0:56:39:6	0:1:0:0
LAT <-218904.000	blown	100	0	5	0:100:0:0	0:1:0:0
⊟ LAT >=-218904.000	string	54	0	13	0:38:54:8	0:0:1:0
LAT <22621.900	string	100	0	4	0:0:100:0	0:0:1:0
⊟ LAT >=22621.900	blown	56	0	9	0:56:33:11	0:1:0:0
HAR <942690.000	string	50	0	4	0:25:50:25	0:0:1:0
HAR >=942690....	blown	80	0	5	0:80:20:0	0:1:0:0
HAR >=989678.688	blown	89	11	9	11:89:0:0	0:1:0:0
⊟ S >=1.000	blown	32	26	19	26:32:26:16	0:0:0:0
⊟ LAT <-343387.000	percussion	33	33	9	33:0:33:33	0:0:0:0
LAT <-696760.000	percussion	50	50	4	50:0:50:0	1:0:1:0
LAT >=-696760.000	struck_Hrm	60	20	5	20:0:20:60	0:0:0:1
⊟ LAT >=-343387.000	blown	60	20	10	20:60:20:0	0:1:0:0
HAR <856620.188	percussion	67	67	3	67:33:0:0	1:0:0:0
HAR >=856620.188	blown	71	0	7	0:71:29:0	0:1:0:0
⊟ S >=2.000	percussion	78	78	107	78:1:14:7	1:0:0:0
HAR <772931.313	percussion	100	100	61	100:0:0:0	1:0:0:0
⊟ HAR >=772931.313	percussion	48	48	46	48:2:33:17	0:0:0:0
⊟ LAT <-485895.000	percussion	58	58	38	58:3:18:21	1:0:0:0
LAT <-1226300.000	percussion	87	87	15	87:7:7:0	1:0:0:0
⊟ LAT >=-1226300.000	percussion	39	39	23	39:0:26:35	0:0:0:0
⊟ S <4.000	struck_Hrm	40	30	20	30:0:30:40	0:0:0:0
S <3.000	percussion	67	67	3	67:0:0:33	1:0:0:0
⊟ S >=3.000	struck_Hrm	41	24	17	24:0:35:41	0:0:0:0
⊟ LAT <-671080.000	string	50	33	12	33:0:50:17	0:0:1:0
LAT <-1008...	string	60	0	5	0:0:60:40	0:0:1:0
LAT >=-100...	percussion	57	57	7	57:0:43:0	1:0:0:0
LAT >=-671080....	struck_Hrm	100	0	5	0:0:0:100	0:0:0:1
S >=4.000	percussion	100	100	3	100:0:0:0	1:0:0:0
LAT >=-485895.000	string	100	0	8	0:0:100:0	0:0:1:0

Fig. 5. C4.5 results testing of the decision attribute instruments against the HRM, LAT and S descriptors. The Class files indicate whether the instruments are percussive, blown, string or struck harmonics.

Automatic Singing Voice Recognition Employing Neural Networks and Rough Sets

Paweł Żwan, Piotr Szczuko, Bożena Kostek, and Andrzej Czyżewski

Gdańsk University of Technology, Multimedia Systems Department
Narutowicza 11/12, 80-952 Gdańsk, Poland
{zwan,bozenka}@sound.eti.pg.gda.pl

Abstract. The aim of the research study presented in this paper is the automatic singing voice recognition. For this purpose a database containing singers' sample recordings has been constructed and parameters are extracted from recorded voices of trained and untrained singers of various voice types. Parameters, which are especially designed for the analysis of the singing voice are described and their physical interpretation is given. Decision systems based on artificial neutral networks and rough sets are used for automatic voice type/voice quality classification. Results obtained in the automatic classification performed by both decision systems are then compared and conclusions are derived.

Keywords: Singing voice, Feature extraction, Artificial Neural Networks, Rough Sets, Automatic Classification, Music Information Retrieval.

1 Introduction

A parametric description is necessary in many applications related to the automatic sound recognition. Such systems are well developed in speech, as well as in the musical instrument sound domains due to many existing applications both in speech content/speaker automatic recognition and music information retrieval (MIR). Singing and speech have a common voice production organ but singing is a form of an artistic expression therefore new parameters are required to be defined and extracted. Those parameters are shortly described in the first part of this paper. A very complicated biomechanics of the singing voice [6, 13] requires numerous features to describe its operation. Such a parametric representation needs intelligent decision systems in the classification process. In the paper artificial neural network (ANN) and rough set (RS) decision systems are employed for the purpose of the singing voice type/voice quality recognition. For every singing voice sample a feature vector (FV) containing 331 parameters was extracted. Resulting parameters were divided into two groups: a so-called "dedicated" designed by the authors for singing voice and more general which may be found in the literature on MIR and on speech recognition. The decision system ability of automatic singing voice recognition is discussed by comparing

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 793–802, 2007.

the efficiency of ANN and RS in two categories: 'voice type' (classes: bas, baritone, tenor, alto, mezzo-soprano, soprano) and 'voice quality' (classes: amateur, semi-professional, professional).

2 Singing Voice Parametrization

Singing is produced by vibration of human vocal cords and resonances in the throat and head cavities. As a result of resonances formants in the spectrum of produced sounds appear. Formants are not only related to articulation allowing for production of different vowels but also characterize timbre and voice type qualities. The formant of the middle frequency band (3.5 kHz) is described in literature as 'singer's formant' and its relation to voice quality has been proved [2, 13]. The interaction between two factors, namely glottal source and resonance characteristics shapes, and the resulting timbre and power of an outgoing vocal sound is equally important. The relation between them is not simple but can be simplified by assuming the linearity of the vocal tract filter. Since in the proposed model there exists an analogy between FIR filtering and singing sound which can be represented as a convolution of glottal source and impulse response of the vocal tract, singing voice parameters can be divided into two groups related to those two factors. In literature some inverse filtration methods for deriving glottis parameters are presented, however they are inefficient due to phase problems [6]. In this aspect only parameters of vocal tract formants are possible to calculate directly from the inverse filtering analysis since they are defined in the frequency domain. Glottal parameters must be parameterized by other methods which will be shown later. On the other hand, vocal tract parameters can be derived from the warped-LPC method (further called WLPC analysis) resulting in frequencies and levels of formants. The WLPC analysis allows for controlling low frequency resolution, which is crucial for precise extraction of formants independently from sound pitch [5, 14]. Since WLPC analysis is applied to small signal frames it can be performed for several parts of the analyzed sounds. Therefore, any formant parameter Fforms a vector which describes its values in consecutive frames. Median values of this vector represent a so-called static parameter F_{med}, variances of vector values are dynamic representation and are denoted as F_{var}.

Some of the singing voice parameters must be calculated for a whole sound and not for single frames. Those parameters are defined on the basis of the fundamental frequency contour analysis and they are related to vibrato and intonation. Vibrato is described as a modulation of the fundamental frequency of sounds performed by singers in order to change timbre of sounds, intonation is their ability to produce sounds perceived as stable and precise in tune.

Another way of determining singing voice parameters is the use of a more general signal description such as descriptors contained in the MPEG-7 standard. Although those parameters are not related to the singing voice biomechanics, they may be useful in the singing voice recognition process. The MPEG-7 parameters are not to be presented in detail here, since they were reviewed in previous work by one of the authors [9]. The MPEG-7 audio parameters can be divided into the following groups:

- *ASE* (Audio Spectrum Envelope). Mean values and variances of each spectral coefficient over time are denoted as $ASE_1 \ldots ASE_{34}$ and $ASE_{1var} \ldots ASE_{34var}$ respectively.
- *ASC* (Audio Spectrum Centroid). The mean value and the variance are denoted as ASC and ASC_{var} respectively.
- *ASS* (Audio Spectrum Spread). The mean value and the variance over time are denoted as ASS and ASS_{var} respectively.
- *SFM* (Spectral Flatness Measure) calculated for each frequency band. The mean values and the variances are denoted as $SFM_1 \ldots SFM_{24}$ and $SFM_{1var} \ldots SFM_{24var}$.
- Parameters related to discrete harmonic values: *HSD* (Harmonic Spectral Deviation), *HSS* (Harmonic Spectral Spread), *HSV* (Harmonic Spectral Variation).

The level of the first harmonics changes for different voice type qualities [13]. Parameters employed in the analysis were defined for harmonic decomposition of sounds: mean value of differences between amplitudes of a harmonic in adjacent time frames (s_n, where n is the number of a harmonic); mean value of amplitudes Ah of a harmonic over time (m_n; standard deviation of amplitudes Ah of a harmonic over time (md_n.

Other parameters used in experiments were: brightness (br) (center of spectrum gravity) [8] and mel-cepstrum coefficients mcc_n, where n is the number of a coefficient.

2.1 Vocal Tract Parameters

As described in previous Section estimation of formants requires methods of analysis with a good frequency resolution which is additionally dependent on pitch of the sounds. When resolution is not properly set single harmonics can be erroneously recognized as formants. For those purposes the WLPC analysis seems to be the most appropriate because λ parameter set in this analysis can be changed in function of pitch of analyzed sounds [5]. However parameters related to the 'singing formant' can be also extracted on the basis of the FFT power spectrum parametrization. Correlation between WLPC and FFT parameters is not a problematic issue. Various methods, among them statistical analysis and rough set method enable to reduce redundancy in FVs and to compare significance of parameters. In Fig. 1 WLPC and FFT analyses results are presented along. Maxima/minima of WLPC curves are determined automatically by an algorithm elaborated by one of the authors [14].

Extracted WLPC maxima are related to one of the three formants respectively: articulation, singer's (singing) and high singing formants. Since in literature a formal prescription how to define mathematically these formants does

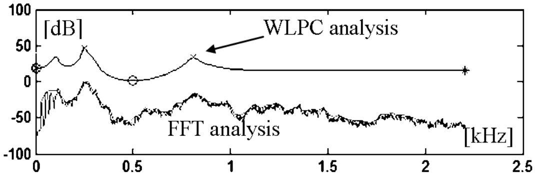


Fig. 1. WLPC analysis shown along with the FFT power spectrum analysis of sound

not exist, three definitions of each of those formants can be proposed basing on three WLPC minima.

$$F_{nm} = WLPCmx_n - WLPCmn_m \tag{1}$$

where $WLPCmx_n$ is a value of nth WLPC maximum and $WLPCmn_m$ is a value of mth WLPC minimum.

Some additional parameters related to the WLPC analysis have also been defined.

2.2 Glottal Source Parameters

Interaction between vocal tract filter and glottal shape are, along with phase problems, obstacles for an accurate automatic glottal source shape extraction [6, 7, 13]. Glottal source parameters, which are defined in the time domain, are not easy to compute from the inverse filtration but within the context of singing voice quality their stability rather that their objective values seem to be important. The analysis must be done for single periods of sound and a sonogram analysis with small analyzing frames and big overlapping should be employed.

For each frequency band a sonogram consists of a set of n sequences $S_n(k)$, where n is the number of the frequency band and k denotes the sample number. Since the aim of parametrization is a description of stability of energy changes in sub-bands, the autocorrelation function in time of sequences $S_n(k)$ is employed. The more frequent and stable are energy changes in a sub-band, the higher are the values of the maximum of the autocorrelation function (for index not equal to 0). The analysis in experiments is performed for 16 and 32 sample frames. In the first case energy band of 0-10 kHz is related to four first indexes n and the maximum of the autocorrelation function of a sub-band n is denoted as KX_n (2), in the second case n=1...8 and the resulting parameter is defined as LX_n (3). Analyzing signal for two different analyzing frames is performed for comparison purposes, only.

$$KX_n = \max(\underset{k}{Corr}(S_n^{16}(k))), \quad n = 1...4 \tag{2}$$

$$LX_n = \max(\underset{k}{Corr}(S_n^{32}(k))), \quad n = 1...8 \tag{3}$$

where $\underset{k}{Corr}(.)$ is the autocorrelation function in time domain, k – sample number, n - number of the frequency sub-band, S_n^{16}- sonogram samples sequence for the analyzed frame of 16 samples and frequency sub-band n, and S_n^{32}denotes sonogram sample sequence for the analyzed frame of 32 samples and the frequency sub-band n. Conversely, minimum of the correlation $Corr(S_n(k))$ function is related to the symmetry or anti-symmetry of energy changes in sub-bands, which is related to open quotient of glottis source [14]. Therefore in each of the analyzed sub-band KY_n and LY_n parameters are defined as (4) and (5), respectively:

$$KY_n = \min(\underset{k}{Corr}(S_n^{16}(k))), \quad n = 1...4 \tag{4}$$

$$LY_n = \min(\underset{k}{Corr}(S_n^{32}(k))), \quad n = 1...8 \tag{5}$$

where $\underset{k}{Corr}(.)$, k, n, S_n^{16},S_n^{32} are defined as in formulas (2) and (3).

Another parameter defined for each analyzed sub-band is a threshold parameter KP_n defined as the number of samples exceeding the average energy level of the sub-band n divided by the total number of samples in the sub-band. For the frame of 32 samples a similar parameter is defined and denoted as LP_n. Parameters KP_n and LP_n are also related to the open quotient of the glottal signal [14].

2.3 Vibrato and Intonation Parameters

In order to calculate vibrato parameters pitch contour needs to be extracted. There are several methods of automatic sound pitch extraction, of which autocorrelation method seems to be appropriate [4]. Autocorrelation pitch extraction method consist in determination of the maximum of the autocorrelation function of the overlapped segments of the audio signal. The fundamental frequency (f_0) within each analyzed frame is determined, and at the same time the frequency resolution of the analysis is achieved by interpolating three samples around the maximum autocorrelation function value. In experiments the length of the frame has been set to 512 samples. The pitch of analyzed sounds is not always stable in time, especially when sounds of untrained singers are concerned. In order to parameterize accurately vibrato and intonation of the analyzed sound there should be determined an equivalent pitch contour of the sound but without vibrato. The result of such an analysis is a so-called 'base contour' which is calculated by smoothing the pitch contour (using a moving average method) with frame length equal to reciprocal of half of the vibrato frequency.

Parametrization of vibrato depth and frequency (f_{VIB}) may be not sufficient in the category of singing quality. Since the quality of the sound reflects the stability of singing parameters in time [3, 13] additional three vibrato parameters are defined: '*perdiodicity'* of vibrato pitch contour, defined as the maximum value of the autocorrelation of pitch contour function (for index not equal to 0); '*harmonicity*' of vibrato by calculating Spectrum Flatness Measure for spectrum of the pitch contour; '*sinusoidality*' of vibrato VIB_S defined as similarity of the parameterized pitch contour to the sine waveform [3].

To parameterize intonation of singing a base contour is utilized. To calculate intonation parameters two methods have been proposed. The first method calculates medium value of a differential sequence of a base contour (IR). The second method does not analyze all base contour samples but the first and the last one and returns *IT* parameter. Parameters *IR* and *IT* are also defined for first and last $N/2$ samples of pitch contour separately (N is the number of samples of pitch contour) and are denoted as IR_{att}, IT_{att}, IR_{rel}, IT_{rel}, where *att* means attack, and *rel* – release of the sound.

3 Experiments

Singing voice database was formed and contains over 2900 sound samples. 1700 of them were recorded from 42 singers in a studio environment. Each vocalist recorded 5 vowels: 'a', 'e', 'i', 'o', 'u' at several sound pitches belonging to natural voice scale. Vocalists consisted of three groups: amateurs (Gdansk University of Technology Choir vocalists), semi-professionals (Gdansk Academy of Music, Vocal Faculty students) and professionals (qualified vocalists, graduated from the Vocal Faculty of the Gdansk Academy of Music). The second group of samples (1200) have been edited from professional CD recordings of famous singers. The database of professionals needed to be extended due to the fact that voice type recognition is possible to perform only for professional voices. Amateur voices do not show much differences within groups of male and female voices [2, 14].

3.1 Neural Network Results

Since Artificial Neural Networks are widely used in automatic sound recognition [8, 9, 14] the ANN classifier was tested first. The ANN employed was a simple feed-forward, three layer network with 100 neurons in hidden layer and 3 or 6 neurons in output layer respectively (dependent on the number of classes being recognized). The input layer consisted of 331 neurons. Sounds from the database were divided into three groups, namely: training (70%), validation (10%) and testing (20%). Samples in training, validation and testing sets consisted of sounds of different vowels and pitches. The network has been trained smoothly and the validation error started to increase after approx. 3000 cycles of training. To train the network optimally a minimum of global validation error function had to be observed. If the validation error was increasing for 50 successive cycles, the last validation error function minimum was assumed to be global and the learning was stopped. In Table 1 recognition results are presented for the voice type category. Rows in table describe recognized voice type classes, and columns correspond to the ANN-based classification.

A total number of sounds on which the classifier was tested was 254 in the voice quality and 181 in the voice type categories. The average recognition result amounts to 94.1% (amateur – 96.3%, semi-professional – 94.3%, and professional – 89.5%) and 90% respectively. What is important, in most cases errors of recognition occurred for 'neighboring' classes. For example only 0.9% professional voice samples were recognized as belonging to amateur class and any tenor samples were recognized as basses, mezzo-sopranos or baritones.

Table 1. ANN singing voice recognition results

Voice type category recognition [%]	bass	baritone	tenor	alto	mezzo	soprano
bass	90.6	6.3	3.1	0	0	0
baritone	3.3	90	6.7	0	0	0
tenor	0	3.6	89.3	7.1	0	0
alto	0	0	4	80	12	4
mezzo	0	0	0	0	93.8	6.3
soprano	0	0	2.9	0	2.9	94.1

3.2 Rough Sets-Based Results

Rough sets introduced by Pawlak [10] are often employed in the analysis of data which aims at discovery of significant data and eliminating redundant ones. The vast literature in the domain of rough sets cover many applications [11], RS are also used in music information retrieval [9]. Within the context of this paper, the rough set method was also used for the analysis of descriptors defined for the purpose of this study. In experiments the rough set decision system RSES was employed [12]. Since this system is widely used by many researches, thus the details concerning its algorithmic implementation and performance will not be provided here. FVs were divided into training and testing sets. Parameters were quantized according to the RSES system principles. The local discretization was used. Local discretization method uses Maximal Discernibility (MD) Heuristics [1] and allows for reducing the number of parameters before the rough sets system is trained.

In the category of voice quality the vector of parameters was reduced to 20 parameters listed below:
$FV= [F_{11}, F_{21}, F_{31}, F_{33}, F_{12var}, F_{13min}, F_{13var}, KX_1, KX_2, KP_1, LP_3, f_{VIB}, VIB_p, LAT, TC, ASE_6, ASE_7, ASE_8, ASE_{21}, s_2]$

Parameters which do not have impact on decision are removed. Those left were utilized in reducts calculation. For this purpose a genetic algorithm available in the RSES system was used, resulting in 11 reducts, each with positive region equal 1.0. Reducts were used in rules generation, then new objects were classified with the rules. If more that one decision was matched to an object, voting method was performed to choose one most frequent.
$\{\mathrm{F}_{11}, \mathrm{F}_{31}, \mathrm{F}_{12var}, \mathrm{KX}_2, \mathrm{f}_{VIB}, \mathrm{VIB}_p, \mathrm{LAT}, \mathrm{TC}, \mathrm{ASE}_6, \mathrm{ASE}_7, \mathrm{ASE}_8\}$
$\{\mathrm{F}_{11}, \mathrm{F}_{31}, \mathrm{F}_{12var}, \mathrm{F}_{13min}, \mathrm{KX}_2, \mathrm{f}_{VIB}, \mathrm{VIB}_p, \mathrm{LAT}, \mathrm{ASE}_6, \mathrm{ASE}_7, \mathrm{ASE}_8, \mathrm{ASE}_{21}\}$
$\{\mathrm{F}_{11}, \mathrm{F}_{31}, \mathrm{F}_{13var}, \mathrm{KX}_2, \mathrm{f}_{VIB}, \mathrm{VIB}_p, \mathrm{TC}, \mathrm{ASE}_6, \mathrm{ASE}_7, \mathrm{ASE}_8, \mathrm{ASE}_{21}\}$
$\{\mathrm{F}_{11}, \mathrm{F}_{31}, \mathrm{KX}_2, \mathrm{f}_{VIB}, \mathrm{VIB}_p, \mathrm{LAT}, \mathrm{TC}, \mathrm{ASE}_6, \mathrm{ASE}_7, \mathrm{ASE}_8, \mathrm{s}_2\}$
$\{\mathrm{F}_{11}, \mathrm{F}_{31}, \mathrm{KX}_2, \mathrm{f}_{VIB}, \mathrm{VIB}_p, \mathrm{LAT}, \mathrm{TC}, \mathrm{ASE}_6, \mathrm{ASE}_7, \mathrm{ASE}_8, \mathrm{ASE}_{21}\}$
$\{\mathrm{F}_{11}, \mathrm{F}_{13min}, \mathrm{KX}_2, \mathrm{KP}_1, \mathrm{f}_{VIB}, \mathrm{VIB}_p, \mathrm{TC}, \mathrm{ASE6}, \mathrm{ASE7}, \mathrm{ASE8}, \mathrm{ASE18}, \mathrm{s}_2\}$
$\{\mathrm{F}_{31}, \mathrm{F}_{12var}, \mathrm{KX}_2, \mathrm{KP}_1, \mathrm{f}_{VIB}, \mathrm{VIB}_p, \mathrm{LAT}, \mathrm{TC}, \mathrm{ASE}_6, \mathrm{ASE}_7, \mathrm{ASE}_8\}$
$\{\mathrm{F}_{31}, \mathrm{F}_{13var}, \mathrm{KX}_2, \mathrm{f}_{VIB}, \mathrm{VIB}_p, \mathrm{TC}, \mathrm{ASE}_6, \mathrm{ASE}_7, \mathrm{ASE}_8, \mathrm{ASE}_{21}, \mathrm{s}_2\}$
$\{\mathrm{F}_{13var}, \mathrm{KX}_2, \mathrm{KP}_1, \mathrm{f}_{VIB}, \mathrm{VIB}_p, \mathrm{TC}, \mathrm{ASE}_6, \mathrm{ASE}_7, \mathrm{ASE}_8, \mathrm{ASE}_{21}, \mathrm{s}_2\}$
$\{\mathrm{F}_{12var}, \mathrm{KX}_2, \mathrm{KP}_1, \mathrm{f}_{VIB}, \mathrm{VIB}_p, \mathrm{LAT}, \mathrm{TC}, \mathrm{ASE}_6, \mathrm{ASE}_7, \mathrm{ASE}_8, \mathrm{ASE}_{21}\}$
$\{\mathrm{F}_{13min}, \mathrm{KX}_2, \mathrm{KP}_1, \mathrm{f}_{VIB}, \mathrm{VIB}_p, \mathrm{LAT}, \mathrm{TC}, \mathrm{ASE}_6, \mathrm{ASE}_7, \mathrm{ASE}_8, \mathrm{s}_2\}$

Among the reduced set of parameters, descriptors related to the WLPC analysis of formants can be found, thus qualified as significant for the classification purposes. They are related to all three formants, thus this proves that in the category of voice quality all formants are required to be parameterized and contained in FVs. It is interesting that among those parameters F_{31} and F_{33}which are related to 'high formant' (middle frequency higher than 5kHz) appeared. The significance of this formant is not described in literature concerning automatic singing voice parametrization. Among glottal source parameters descriptors such as: KX_1, KX_2, KP_1, LP_3 have been selected. On the other hand, frequency (f_{VIB}) and periodicity (VIB_p) related to vibrato modulation found their place among other important descriptors. Among remaining parameters a few MPEG-7 parameters, namely: LAT, TC, ASE_6, ASE_7, ASE_8, ASE_{21} have been qualified, in addition only one parameter related to the analysis of spectrum may be found, which is represented by s_2 related to the variation of the second harmonic.

In the category of voice type over 200 from the total number of 331 parameters have been left during the analysis. It was not possible to obtain such a reduced parameter representation as in the case of the voice quality category. Within this context automatic voice type recognition seems to be more complex. One of the reasons can be a diversity of registers among different voice types and individual voice qualities which change for the same voice type. Additionally, some singers' voices were not easy to qualify to the voice type category, e.g. low registers of soprano voices were similar in timbre to mezzo-soprano and even alto voices.

Parameters derived from the rough set-based analysis for both categories were used for training and testing of the RSES system. 75% of voice samples were used in training and the remaining 25% were used for testing. For voice quality classification 548 rules were generated, with mean support 43, and maximum support 498, mean length 4.2, and maximum length equals 7. For voice type classification 15198 rules with mean support 2.8, and maximal support 28, mean length 5.6, and maximal length 12 have been generated. In Table 2 recognition results are presented for voice type category. Below, examples of rules having the highest support are given.

(KX1="(-Inf, 0.8643") & (fVIB="(0.1067, Inf") & (VIBP="(-Inf, 0.0185") & (ASE6="(-Inf, 1.9116") => (quality=pro.[475])

(KX2="(-Inf, 0.34735") & (fVIB="(0.1067, Inf") & (VIBP="(-Inf, 0.0185") & (ASE6="(-Inf, 1.9116") => (quality=pro.[431])

(F12_var="(0.08080, Inf") & (fVIB="(0.1067, Inf") & (VIBP="(-Inf, 0.0185") & (ASE6="(-Inf, 1.9116") => (quality=pro.[425])

....

Parameters contained in the rules induced by the RSES system correspond to the ones discussed above. Decision 'pro' corresponds to quality of a professional singer, 'semi-pro' denotes a semi-professional singer. Rules that regard the decision 'amateur' contained also a few descriptors, however with a lower support.

Table 2. RSES-based singing voice classification results

Voice type category recognition [%]	bass	baritone	tenor	alto	mezzo	soprano
bass	84.0	10.0	4.0	2.0	0	0
baritone	13.0	64.8	13.0	0	1.9	7.3
tenor	6.0	18.0	54.0	10.0	6.0	6.0
alto	0	4.7	16.3	51.2	16.3	11.6
mezzo	3.8	0	2.6	1.3	73.1	19.2
soprano	2.9	2.9	2.9	1.4	11.4	78.6

The automatic recognition results in the case of the category of quality are better comparing to the ANN. The rough set system achieved very good results employing a reduced FV of 20 parameters in classification of the voice quality category (total accuracy 0.976). They are respectively as follows: amateur – 94.7%, semi-professional – 95.4%, and professional – 96.7%. In the category of voice type the results are much lower. Moreover, in the case of singing voice type category erroneous classification is not always related to neighboring classes. Thus, the RSES system was not able to perform the classification as good as ANN while trained and tested on FVs of more than 200 parameters in the category of voice type where further vector size reduction was not possible (total accuracy obtained equals 0.664). It is quite obvious that types of voices being at the extreme of the voice type category have been recognized with better efficiency than those close to each other.

4 Conclusions

By comparing automatic recognition results of neural network and rough set system two main conclusions may be reached. The recognition performed by the rough set system was better for the quality category and worse in the category of the voice type in comparison to the ANN. In the case of the voice quality category it was possible by the RS system to reduce a large number of parameters to 20 descriptors to be contained in the FVs and the extraction of rules went very smoothly. Descriptors describing the level of formants, stability of glottal parameters along with those related to vibrato, and in addition MPEG-7 descriptors allowed for deriving linear IF-THEN rules. This proves that automatic recognition of quality category is possible on the basis of a significantly reduced number of descriptors contained in the FVs. In the case of the voice type it was not possible to achieve as good recognition results. Neural networks which are capable to perform non-linear class separation enabled to classify particular types of singing voices effectively while the rough-set system achieved lower efficiency. The reason for the lower recognition results may be that the database of singing voices was represented by too low number of different singers.

Acknowledgments

This work was supported by the Ministry of Science and Education (Grant No. 3 T11F 023 30).

References

1. Bazan, J.G., Szczuka, M.S.: The Rough Set Exploration System. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 37–56. Springer, Heidelberg (2005)
2. Bloothoof, G.: The sound level of the singers formant in professional singing. J. Acoust. Soc. Am. 79(6), 2028–2032 (1986)
3. Diaz, J.A., Rothman, H.B.: Acoustic parameters for determining the differences between good and poor vibrato in singing. In: Proc. 17th International Congress on Acoustics, Rome, vol. VIII, pp. 110–116 (2001)
4. Dziubiński, M., Kostek, B.: Octave Error Immune and Instantaneous Pitch Detection Algorithm. J. of New Music Research 34, 273–292 (2005)
5. Harma, A.: A comparison of warped and conventional linear predictive coding. IEEE Transactions on Speech and Audio Processing 5, 579–588 (2001)
6. Herzel, H., Titze, I., Steinecke, I.: Nonlinear dynamics of the voice: signal analysis and biomechanical modeling. CHAOS 5, 30–34 (1995)
7. Joliveau, E., Smith, J., Wolfe, J.: Vocal tract resonances in singing: the soprano voice. J. Acoust. Soc. America 116, 2434–2439 (2004)
8. Kostek, B., Czyżewski, A.: Representing Musical Instrument Sounds for Their Automatic Classification. J. Audio Eng. Soc. 49, 768–785 (2001)
9. Kostek, B.: Perception-Based Data Processing in Acoustics. Applications to Music Information Retrieval and Psychophysiology of Hearing. Series on Cognitive Technologies. Springer, Heidelberg (2005)
10. Pawlak, Z.: Rough Sets. International J. Computer and Information Sciences 11 (1982)
11. Peters, J.F., Skowron, A. (eds.): Transactions on Rough Sets V. LNCS, vol. 4100. Springer, Heidelberg (2006)
12. Rough-set Exploration System: logic.mimuw.edu.pl/~rses/
13. Sundberg, J.: The science of the singing voice, Northern Illinois University Press (1987)
14. Żwan, P.: Expert System for Automatic Classification and Quality Assessment of Singing Voices. In: Proc. 121 Audio Eng. Soc. Convention, San Francisco, USA (2006)

A Learning-Based Model for Semantic Mapping from Natural Language Questions to OWL

Mingxia Gao[1], Jiming Liu[1,2], Ning Zhong[1,3], Chunnian Liu[1], and Furong Chen[4]

[1] The International WIC Institute
Beijing University of Technology, Beijing, China
gaomx@emails.bjut.edu.cn
[2] Department of Computer Science
Hong Kong Baptist University, Hong Kong SAR
[3] Department of Life Science and Informatics
Maebashi Institute of Technology, Maebashi, Japan
[4] R&D Center TravelSky Technology Limited

Abstract. One of key problems in implementing a dynamic interface between human and agents is how to do semantic mapping from natural language questions to OWL. The paper views the task as a two-class classification problem. A pair of question variable and OWL element is a sample. Two classes of "Matched" and "Unmatched" explain two relations between the question variable and the OWL element in a given sample. Building appropriate semantic mapping is the same as classifying the sample to a "Matched" class by an effective machine learning method and a trained model. Two types of features of samples are selected. Syntactical features denote the syntactical structure of a given sample. Semantic features present multiple relations between the question variable and the OWL element in one sample. Preliminary experimental results show that the sum precision of the learning-based model is better than that of the constraints-based method.

1 Introduction

Natural language is a common communication means for human. Natural language questions are the most convenient representation to acquire knowledge. OWL [8] is a new knowledge representation language on the Semantic Web, which can be understood by agents. One of key problems in implementing a dynamic interface between human and agents is how to build semantic mapping from natural language questions to OWL.

Natural language and OWL are in essence two heterogeneous representations for the same semantic knowledge. Building semantic mapping for a natural language question and an OWL knowledge base is to map different syntactical constructs between them based on semantic equivalence. For example, a clause can be matched with an axiom. The basic mapping is from words (or named entities) to elements. Because elements in OWL are the basic units and words

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 803–812, 2007.

in natural language are the indivisible units. The paper focuses on semantic mapping from a set of words to a set of elements.

Currently, researches on semantic mapping between natural language questions and ontology knowledge encounter a few urgent problems. Firstly, most of methods are semiautomatic and require users to manually solve the ambiguity problems in semantic mapping. Secondly, lingual common sense, domain dictionaries and the behaviors of users include a great deal of accessorial knowledge, which is not completely used by existing systems, to reduce the ambiguity in semantic mapping. In order to avoid the problems, we proposed a constraints-based method for semantic mapping from natural language questions to OWL [7]. The method translated the words of a question as well as their syntactical and semantic properties into constrained question variables and functions, and thereafter, utilized an optimization-based assigning mechanism to substitute the question variables with the corresponding constructs in OWL knowledge bases. However, the method is lack in learning ability and its performance cannot be improved with increase of instances.

The paper proposes a learning-based model. The task of semantic mapping can be viewed as a two-class classification problem in the model. A pair of question variable and OWL element is a sample. A given sample can be classified to a "Matched" class or a "Unmatched" class by an effective machine learning method and a trained model. We extract two types of features for samples. Syntactical features deal with different levels of structures of a given question variable, such POS or chunk. Semantic features consider different relations between the question variable and the OWL element in the sample. We have performed some preliminary evaluations using the questions and knowledge base available from the International WIC Institute (WICI)[1].

2 A Workflow of Semantic Mapping Based on Learning

The task of semantic mapping can be viewed as a two-class classification problem. A pair of question variable and OWL element is a sample. Two classes of "Matched" and "Unmatched" explain two matching results between the question variable and the OWL element in a given sample. Building an appropriate semantic mapping is the same as classifying the sample to a "Matched" class by an effective machine learning method.

Figure 1 describes a workflow of semantic mapping based on learning. As shown in Fig. 1(a,b), firstly, we decompose a given question and a given OWL knowledge base. Secondly, we build a set of samples using the set of question variables and the set of elements acquired from the above step as shown in Fig. 1(c). Thirdly, we compute feature vectors according to the defined features as shown in Fig. 1(d). Finally, the samples will be classified based on a certain classifier and the model learned from training data as shown in Fig. 1(e).

[1] http://www.iwici.org/

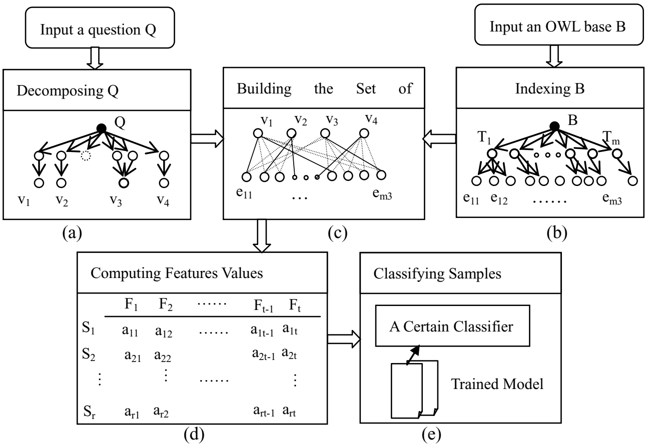


Fig. 1. A workflow of semantic mapping based on learning. v_i in Fig. 1(a) denotes the i^{th} question variables in Q. T_m and e_{mi} in Fig. 1(b) denote the m^{th} triple in B and the i^{th} element in T_m. S_i, F_j and a_{ij} in Fig. 1(d) denote the i^{th} sample, the j^{th} feature, and the value of the j^{th} feature of the i^{th} sample.

2.1 Decomposing Questions

A question is defined as a set of question variables as follows.

Definition 1. *Let $QV := \{v_i\}_{i=1}^{n}$ be a set of question variables and $v_i :=< Type, Term, Attribute >$ be a question variable in QV, where $Type := word|$ named entity denotes the real unit of the question variable, $Term := \{Token_j\}_{j=1}^{r}$ denotes the set of words that compose the question variable, and Attribute is the set of the properties of the question variable. Different types of question variables have different attributes, for example: attributes of a word include {Token, Stem, Lemma, SYN, POS, Chunk, Length, Order}.*

The ultimate goal of decomposing questions is to identify all question variables in a question. In doing so, various basic techniques of natural language processing [5] will be involved, including tokenization, identification of stop words, stemmer, POS, identification of named entities (NE), synonymies. In this work, we use Gate [4], WordNet [6] in practical application.

Figure 2 provides a schematic illustration of question decomposition, along with a sketch of question variables. As shown in the figure, an original question is firstly decomposed through tokenization into candidate question variables containing attributes and terms. Next, part of the variables are replaced with named entities, since they can be learned from related texts. Thirdly, potential stop words are marked. Finally, we supplement attributes of the variables by synonyms or abbreviation.

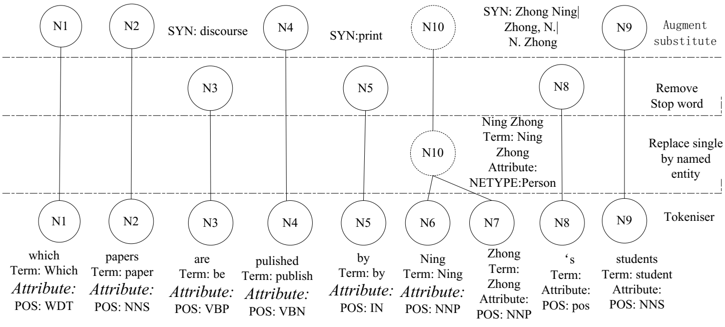

N1
N2
SYN: discourse
N4
SYN:print
N10
SYN: Zhong Ning| Zhong, N.| N. Zhong
N9
Augment substitute
N3
N5
N8
Remove Stop word
Ning Zhong Term: Ning Zhong Attribute: NETYPE:Person
N10
Replace single by named entity
N1
N2
N3
N4
N5
N6
N7
N8
N9
Tokeniser
which Term: Which Attribute: POS: WDT
papers Term: paper Attribute: POS: NNS
are Term: be Attribute: POS: VBP
pulished Term: publish Attribute: POS: VBN
by Term: by Attribute: POS: IN
Ning Term: Ning Attribute: POS: NNP
Zhong Term: Zhong Attribute: POS: NNP
‘s Term: Attribute: POS: pos
students Term: student Attribute: POS: NNS

Fig. 2. A schematic illustration of question decomposition

2.2 Indexing OWL Knowledge Bases

In order to formalize an OWL knowledge base into a set of elements, existing OWL parsers, such as Jena or OWL API, are used to parse an OWL knowledge base into a set of elements. An OWL element in the set is composed of single word or multiple words. The element with multiple words is decomposed into a set of words. Three elements are combined to form a RDF triple to present a terminological or assertion axiom.

An OWL knowledge base can be defined as follows.

Definition 2. *Let Onto* $:= (E, T)$ *be an OWL knowledge base, where* $E = \{e_j\}_{j=1}^{m}$ *is the set of elements,* $T = \{< e_i, e_j, e_k > |1 \leq i, j, k \leq m\}$ *is the set of RDF triples. Let* $e_j :=< Type, Name, Relation >$ *be an element in E, where Type is one of the set* {*class, individual, DatatypeProperty, ObjectProperty, value*}, $Name = \{Token_i\}_{i=1}^{r}$ *is the set of words that compose the element, and* $Relation = \{\{t_j\}_{j=1}^{s} | e_i \in t_j\}$ *is the set of RDF triples including the element.*

2.3 Building Sets of Samples

Given a question *QV* and an OWL knowledge base *Onto*, the most simple criterion to build a set of samples is the Cartesian product $QV \times Onto$. Every question variable in *QV* can be combined with all elements in *Onto* to form $|Onto|$ samples. However, most of elements are not completely related to a given question variable, so that they can be combined with the question variable to form some negative samples. For the task of semantic mapping, we are more interested in positive samples. It is essential to select candidate elements to build samples for a given question variable.

From the viewpoint of surface text matching, we propose a strategy of selecting candidate elements. The strategy compares the set of words of a question variable and an element, and then decides whether or not to build a sample for them. Thus, a sample can be defined as follows.

Definition 3. *Let* $S := \{< v_i, e_j > | v_i \in QV$ *and* $e_j \in E$ *and* $e_j.Name \cap v_i.Term \neq \phi\}$ *be a sample, where* QV *is the set of question variables and* E *is the set of OWL elements.*

3 Feature Extraction

The performance of the learning-based model is tightly coupled with feature extraction. Syntactical and semantic properties of question variables and OWL elements are formalized into two types of features: syntactical and semantic features. Syntactical features denote syntactical characteristics of the question variable in a given question and the OWL element in a given OWL knowledge base. Semantic features present different relations between the question variable and the OWL element in a sample.

Syntactical features of a sample include different levels of xstructures of the question variable, such as POS or chunk, and the OWL element. Every question variable is a lingual member. Some syntactical characteristics of a question variable, such as POS and Chunk, can be regarded as its features. For a given question variable, syntactical characteristics of adjacent variables have latent influence on classifying besides its characteristics. We consider not only POS and Chunk of the question variable in a given sample, but also those of its neighbors. Values of POS and Chunk use Penn Treebank II Tags[2]. The tags are too detailed to use for our task. For example, a noun has four tags such as NN (singular or mass), NNS (plural), NNP (proper noun, singular) and NNPS (proper noun, plural). In fact, the difference can be ignored when we only concern higher-level structures of a question variable. The tags compressed are displayed in Table 1. Syntactical characteristics of an OWL element in a given sample are relatively simple and only include a certain type of the OWL element. Its feature value is an element of the set {C, DP, OP, I, V}, where the elements in the set denote class, DatatypeProperty, ObjectProperty, individual, and value, respectively.

Table 1. Tags for POS and compressed POS

Compressed	NN	WW	VB	JJ	RB	ER
POS	NN,NNS, NNP, NNPS	WDT,WP, WP$, WRB	VB, VBD, VBG, VBN, VBP, VBZ	JJ, JJR, JJS	RB, RBR, RBS	Other tags

Using various information, such as surface text and counterparts, related to the question variable and the OWL element in a sample, the paper defines different types of semantic features.

The common profile is to compare the similarity of surface text of the question variable and the corresponding element. Besides the original token in questions, a word can use lemma, stem, and synonym to represent its meaning.

[2] http://bulba.sdsu.edu/jeanette/thesis/PennTags.html

Hence, *token-match*, *lemma-match*, *stem-match*, and *syn-match* are defined as four direct semantic features. They denote the degree of similarity of surface text among original token, lemma, stem, and synonym of the question variable and the OWL element in a sample, respectively. Eq. (1) is used to calculate the value of *token-match*, where $< v_i, e_j >$ is a given sample. *lemma-match*, *stem-match* and *syn-match* can use similar formulas.

$$\textit{token-match} = \begin{cases} 1 & : \quad v_i.token \cap e_j.token = v_i.token \\ 0.5 & : \quad v_i.token \cap e_j.token \neq null \\ 0 & : \quad v_i.token \cap e_j.token = null \end{cases} \tag{1}$$

An OWL element is often composed of multiple words. It may match other words in the same question besides the question variable. The case indirectly improves the degree of similarity between the question variable and the OWL element. We define a semantic feature named *other-term* to reflect the case by Eq. (2).

$$\textit{other-term} = \begin{cases} 1 & : \quad \exists w \; w \in e_j.Token \; and \; w \neq v_i \; and \; w \in q \\ 0 & : \quad otherwise \end{cases} \tag{2}$$

where $< v_i, e_j >$ is a given sample, v_i presents a question variable of q, and w denotes a word in q.

Except for the similarity between the question variable and the OWL element in a given sample, the relations between their respective counterparts have influence on semantic mapping. A counterpart of an OWL element is an OWL element that is combined with the OWL element to form a RDF triple. Counterparts of a question variable are other question variables in the same question and lingual members through Wordnet, such as synonym, hyponyms, and hypernyms. In order to present the relations between counterparts of the question variable and counterparts of the OWL element in a given sample, we define features, such as *other-variable*, *synonym 2*, *hypernyms*, and *hyponyms* by Eq. (3).

$$\textit{other-variable} = \begin{cases} 1 & : \quad \exists e_k \; v_s \in e_k.Token \\ 0 & : \quad otherwise \end{cases} \tag{3}$$

where $< v_i, e_j >$ is a sample, e_k is a counterpart of e_j, v_s is a counterpart of v_i. For *other-variable*, *synonym 2*, *hyponyms*, and *hypernyms*, v_s denotes another question variable in the same question, synonym, hyponyms and hypernyms of v_i.

For two samples $< e_1, v_1 >$ and $< e_2, v_2 >$ of a question and an OWL knowledge base, e_1 and e_2 may have a shared counterpart e_k. The shared OWL element can be viewed as an intermedius to build an indirect relation of counterpart between e_1 and e_2. Because v_1 and v_2 are from the same question, v_1 is a counterpart of v_2. In order to consider the relation of two samples, we define a feature named *sharing-element* for the samples by Eq. (4).

$$\textit{sharing-element} = \begin{cases} 1 & : \quad \exists e_k \; e_k \in e_1.Counterpart \; and \; e_k \in e_2.Counterpart \\ 0 & : \quad otherwise \end{cases} \tag{4}$$

There must be other features that can influence semantic mapping. In this work, we only consider the features based on current properties of question variables and OWL elements.

4 Experiments

4.1 Experimental Data

The OWL knowledge base of our experiments is built on the basis of the WICI portal, which includes 83 classes, 90 individuals, 37 object properties, and 20 datatype properties. We have used two types of natural language questions in our experiments: real questions (RQuestions) selected from students in the WICI; and simulated questions (SQuestions) manually produced regarding the instances of the WICI based on the questions set from Webclopedia.[3]. Table 2 shows the number of questions and the number of samples (*sample 1*). The samples are built with sets of questions variables decomposed from the questions and the set of elements parsed from the OWL knowledge base related to the WICI.

Though questions decomposition has marked the potential stop words, such as "on" ,"of", some verbs like "do", "has" are not confirmed as stop words completely. They are hold in the set of question variables and formed into the set of samples, because they have practice meaning in some questions. Hence, these samples can become noises when the question variables in the samples have not any meaning in questions. As shown in Table 2, we removed the samples with stop words to form another set of samples (*sample 2*).

Table 2. Questions and samples

	Questions	*Samples 1*			*Samples 2*		
		Sum	positive	negative	sum	positive	negative
RQuestions	69	578	242	336	524	242	282
SQuestions	40	271	124	147	234	124	110

4.2 Experimental Design and Results

In order to select an effective classifier for the task of semantic mapping, we used a special machine learning tool named Weka [14] in practical experiments. Weka is implemented in Java and has a good GUI and convenient APIs. It includes various series of Classifiers, such as bayes series, functions series, tree series and so on. We carried out several experiments by different Classifiers with default parameters in Weka, and selected results of three typical Classifiers (J48, SMO, and KStar) by comparing their precisions as shown in Table 3. From the results, we obtain three conclusions. The first one is that RQuestions and SQuestions present the prominent difference for KStar. For two sets of samples, the results of RQuestions is better than that of SQuestions. KStar is an instance-based classifier (i.e., the class of a test instance is based upon the class of those training instances similar to it), as determined by some similarity functions. Through analyzing two sets of questions, we can see that there are more similar questions in the set of real questions than in the set of simulated questions. The second one is that the precision

[3] http://www.isi.edu/natural-language/projects/webclopedia/

of J48 rarely declined from samples 1 to samples 2. J48 is a trees classifier, which is not sensitive about noises in samples. The samples with stop words can be viewed as noises when the question variables in the samples have not any meaning in questions. The third one is that the precision of sum and negative is declined and the precision of positive is improved from samples 1 to samples 2. The results show that removing stop words is useful for the task of semantic mapping.

Table 3. The precision of simulated questions and real questions. *SS1* and *SS2* denote *sample 1* and *sample 2* of SQuestions; *PS1* and *PS2* denote *sample 1* and *sample 2* of RQuestions; P denotes precision

		SS1			*SS2*			*PS1*			*PS2*		
Classifier		J48	SMO	KStar	J48	SMO	KStar	J48	SMO	KStar	J48	SMO	KStar
P %	Positive	80.1	85.0	72.6	82.7	81.3	79.0	78.4	76.1	82.0	77.7	75.2	84.1
	negative	88.9	88.9	71.5	86.1	81.1	66.4	84.3	83.7	83.1	81.9	80.3	80.0
	sum	84.50	87.08	71.96	84.19	81.20	71.79	81.83	80.45	82.70	79.95	77.86	81.68

In order to evaluate the precision of the learning-based model, we compare the precision of the learning-based method (called LBM) with the constraint-based method (called CBM) with same weight for different constraints, which is proposed in [7]. However, the measure of CBM introduced in [7] does not accord with LBM in the paper. A new evaluating method is defined by Eq. (5), which is based on the precision of question variables for CBM. For the clarity of description, we will make the following assumption: $n = n_1 + n_2 + n_3$ denotes the number of question variables, n_1 presents the number of the question variables obtained right mapping, n_2 is the number of the question variables obtained wrong mapping, and n_3 is the number of the question variables that cannot obtain sound mapping because of limitation of constraints.

$$precision = \frac{n_1 + (a_1/b_1 + \ldots + a_{n3}/b_{b3})}{n} \qquad (5)$$

where a_i and b_i denote the number of suited elements and candidate elements of the i^{th} question variable that cannot obtain sound mapping because of limitation of constraints.

Figure 3 gives the precision comparisons between LBM and CBM methods. From this figure, we can see that LBM is better than CBM. However, for SS1 and PS1 the precisions of LBM have been improved by over 15% and present a significant improvement over that of CBM. The results show that removing stop words is more useful for the CBM method.

5 Related Work

Natural language interfaces to database [1,2,11,10,13] are the same problems as semantic mapping between natural language questions and OWL knowledge.

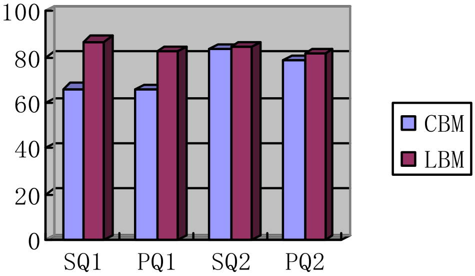


Fig. 3. The precision comparisons between LBM and CBM

Previous methods [1,2] have used predicate logic as the representation language to manually construct a concept map, which captures the concepts and roles involved in a question. PRECISE [11,10] parsed questions to the corresponding SQL queries using a statistical parser, a lexicon, and a maxflow algorithm. Others [13] have explored a learning-based approach that combines different learning methods in inductive logic programming (ILP) to allow learners to produce more expressive hypotheses than that of an individual learner and to build a predicate lexicon with different learning methods.

There are some researches [3,12,9] about semantic mapping between natural language questions and ontology knowledge. MOSES [12] can deal only with questions in Denish and Italian. AquaLog [9] dealt specifically with English questions by using customized triples as the intermediate representation language. It required users to manually solve the ambiguity problems in semantic understanding. Strictly speaking, the work presented in [3] was not a real semantic mapping from natural language questions to ontology. Solvable questions of the system was a subset of natural English (controlled English).

Compared with the related works, the main features of our work include that the proposed method without demanding additional information or intervention from users is automatic, uses many syntactical and semantic characteristics from analysis of questions and OWL knowledge bases, and can be improved with increase of instances.

6 Conclusions

In order to map natural language questions to OWL, the paper proposed a learning-based model, which views the semantic mapping as a two-class classification problem. Building an appropriate semantic mapping between a question variable and an OWL element is the same as classifying the samples to a "Matched" class by an effective machine learning method. Our preliminary experiments using the questions and knowledge base available from the WICI have indicated that the proposed model is promising for further development.

Acknowledgements

The work is partially supported by the NSFC major research program: "Basic Theory and Core Techniques of Non-Canonical Knowledge" (60496322), the National Natural Science Foundation of China (60673015, 60642003).

References

1. Androutsopoulos, I., Ritchie, G., Thanisch, P.: MASQUE/SQL-An Efficient and Portable Natural Language Query Interface for Relational Database. In: Proc. 6th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert System, Edinburgh (1993)
2. Androutsopoulos, I., Ritchie, G., Thanisch, P.: Natural Language Interfaces to Databases - An Introduction. Journal of Natural Language Engineering 1(1), 29–81 (1995)
3. Bernstein, A., Kaufmann, E., Fuchs, N. et al.: Talking to the Semantic Web - a Controlled English Query Interface for Ontologies. AIS SIGSEMIS Bulletin 2(1), 42–47 (2005)
4. Cunningham, H.: Software Architecture for Language Engineering. Doctor of Philosophy, Department of Computer Science, University of Sheffield (June 2000)
5. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing. In: Computational Linguistics, and Speech Recognition, Prentice-Hall, Englewood Cliffs (2000)
6. Fellbaum, C. (ed.): WordNet An Electronic Lexical Database. MIT Press, Cambridge (1998)
7. Gao, M., Liu, J., Zhong, N., Chen, F.: A Constraints-based Semantic Mapping Method from Natural Language Questions to OWL. In: Proc. IEEE Symposium on Computational Intelligence and Data Mining (CIDM'07), Honolulu, Hawaii, USA, April 1-5, 2007 (in press) (2007)
8. Horrocks, I., Patel-Schneider, P., Harmelen, F.: From SHIQ and RDF to OWL: the Making of a Web Ontology language. Journal of Web Semantics 1(1), 7–26 (2003)
9. Lopez, V., Pasin, M., Motta, E.: AquaLog: An Ontology-Portable Question-Answering System for the Semantic Web. In: Gómez-Pérez, A., Euzenat, J. (eds.) The Semantic Web: Research and Applications. LNCS, vol. 3532, pp. 546–562. Springer, Heidelberg (2005)
10. Popescu, A., Armanasu, A., Etzioni, O. et al.: Modern Natural Language Interfaces to Databases: Composing Statistical Parsing with Semantic Tractability. In: Proc. 20th International Conference on Computational Linguistics, Geneva (August 2004)
11. Popescu, A., Etzioni, O., Kautz, H.: Towards a Theory of Natural Language Interfaces to Databases. In: Proc. 8th International Conference on Intelligent User Interfaces, Miami, Florida, USA (2003)
12. Paggio, P., Hansen, D.: Ontology-based Question Analysis in a Multilingual Environment: The MOSES Case Study. In: Proc. OntoLex2004: Ontologies and Lexical Resources in Distributed Environments (May 2004)
13. Tang, L., Mooney, R.: Using Multiple Clause Constructors in Inductive Logic Programming for Semantic Parsing. In: Proc. 12th European Conference on Machine Learning, pp. 466-477 (2001)
14. Witten, I., Frank, E.: WEKA Machine Learning Algorithms in Jav. In: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann Publishers, San Francisco (2000)

Filtering and Sophisticated Data Processing for Web Information Gathering

Yuefeng Li[1], Ning Zhong[2], Xujuan Zhou[1], and Sheng-Tang Wu[1]

[1] School of Software Engineering and Data Communications, Queensland University of Technology, Brisbane, QLD 4001, Australia
y2.li@qut.edu.au,x.zhou@student.qut.edu.au,st.wu@student.qut.edu.au

[2] Department of Information Engineering, Maebashi Institute of Technology, 460-1 Kamisadori-Cho, Maebashi-City 371-0816, Japan
zhong@maebashi-it.ac.jp

Abstract. Mismatch and overload are two fundamental issues regarding the efficiency of Web information gathering. To provide a satisfactory solution, this paper presents a Web information gathering system that encapsulates two phases: the filtering and sophisticated data processing. The objective of the filtering is to quickly filter out most irrelevant data in order to avoid mismatch. The phase of the sophisticated data processing can use more sophisticated techniques without carefully considering time complexities. The second phase is for solving the problem of the information overload.

1 Introduction

There are two fundamental issues regarding the efficiency of Web information gathering (WIG) [9]: mismatch and overload. The mismatch means some data that meets user needs has not been found (or missed out), whereas, the overload means some gathered data is not what users need.

Traditional techniques related in information retrieval (IR) have touched the fundamental issues [2]. However, IR-based systems neither explicitly describe how the systems act like users nor discover enough knowledge from very large data sets to answer want users really want. This issue has challenged the artificial intelligence (AI) community to address "what has information gathering to do with AI" [7]. In a one time, many agent-based approaches have been presented for this challenge [8]. Unfortunately, agent-based approaches can only show us the architectures of information gathering systems. They cannot provide more contributions for finding useful knowledge from data to overcome the fundamental issues.

Currently, the application of data mining techniques to Web data, called Web mining, is used to discover knowledge from Web data to help users for information gathering [17]. Web mining can be classified into four categories: Web usage mining, Web structure mining, Web content mining and Web user profiles mining [4,15,20,11].

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 813–823, 2007.

It is obvious that Web information gathering systems must be quickly response their users' requests. However, the procedure of mining useful patterns from large numbers of Web data takes a lot of time. It is also a big challenge to guarantee the quality of discovered knowledge in Web data since duplications and ambiguities of data values (e.g., terms). On the other hand, there actually only a small amount of Web data is relevant to a certain user or a group of users. Therefore, it is desirable to quickly filter out most irrelevant data first and then conduct sophisticated data processing.

To provide a satisfactory solution for the fundamental issues, in this paper we design a Web information gathering system that encapsulates two phases: the filtering and sophisticated data processing. The objective of the filtering is to quickly filter out most irrelevant data. The expected result of filtering is only a small amount of inputs left. So the phase of the sophisticated data processing can use more sophisticated techniques without carefully considering time complexities.

2 Definitions

It is easier for users to answer which documents are relevant rather than describe what they really wants [3]. In this research we assume that the users can provide feedback to indicate which documents are positive (relevant) or negative (irrelevant), that is, we can obtain a training set of relevant documents and irrelevant documents.

Let $T = \{t_1, t_2, \ldots, t_k\}$ be a set of keywords (or terms), and D be a training set of documents, which consists of a set of positive documents, D^+; and a set of negative documents, D^-, where each document is a set of terms (may include duplicate terms).

In the phase of filtering, we let $D^- = \emptyset$ because we attempt to use a smallest training set for quickly filtering out the irrelevant information.

A set of terms is referred to as a *termset*. Given a document d (or a paragraph) and a term t, we define $tf(d,t)$ the number of occurrences of t in d. A set of term frequency pairs, $P = \{(t,f) | t \in T, f = tf(t,d) > 0\}$, is referred to as a *pattern* in this paper.

Let $termset(P) = \{t | (t,f) \in P\}$ be the termset of P. In this paper, pattern P_1 equals to pattern P_2 if and only if $termset(P_1) = termset(P_2)$. A pattern is uniquely determined by its termset. Two patterns should be composed if they have the same termset (or they are in a same category). In this paper, we use the composition operation, $\oplus$, that defined in [11] to generate new patterns.

Let P_1 and P_2 be two patterns. We call $P_1 \oplus P_2$ the composition of P_1 and P_2 which satisfies:

$$\begin{aligned} p_1 \oplus p_2 = &\{(t, f_1 + f_2) | (t, f_1) \in p_1, (t, f_2) \in p_2\} \cup \\ &\{(t,f) | t \in (termset(p_1) \cup termset(p_2)) - \\ &(termset(p_1) \cap termset(p_2)), (t,f) \in p_1 \cup p_2\} \end{aligned}$$

In the phase of filtering, we use rough association rules (see [12,13]). A rough association rule has the form of

$$< termset, wd > positive,$$

where *termset* is set of selected terms, and *wd* is a weight distribution of these terms in the rule. We can get rough association rules by using the composition operation on a set of patterns and then normalizing the corresponding weights (see [12] or [13]).

3 Filtering

The simplistic approach of filtering is using term vector spaces (e.g., a set of key words) [6,18]. The main disadvantage of the simplistic approach is the poor interpretation of negative information. In this research, we will present an ontology-based filtering model.

3.1 Ontology Extraction

Syntactically we assume that user interests (classes) can be constructed from some primary ones. Fig. 1 shows such an ontology, which organizes a set of discovered patterns. The set of **primitive objects** $\Theta = \{pet, shop, city, accommodation\}$. Because *dog* and *cat* are *pets* (also *hotel* is-an *accommodation*), we use "is-a" links to show the relation between them and the corresponding primitive objects. There are three relevant documents in the training set, which are represented as a set of keyword-frequency pairs:

$$d_1 = \{(dog, 4), (shop, 6)\}, \quad d_2 = \{(cat, 5), (shop, 15)\}, \quad \text{and}$$
$$d_3 = \{(pet, 3), (shop, 7), (city, 10)\}.$$

Using the inheritance (or is-a relation), we can obtain two compound objects: p_1 and p_2 (see Fig. 1) from d_1, d_2 and d_3, where, $d_1 \rightarrow p_1$, $d_2 \rightarrow p_1$ and $d_3 \rightarrow p_2$. The "part-of" relation is used to show the relation between compound and primitive objects. A document is *irrelevant* if its any "part-of" section does not include any pattern.

In order to measure the relationship between classes, we can define an identity for each class X, that is, $id(X) = \{Z|Z$ is a primitive class, and there is a path from X to $Z\}$, e.g., $id(p_1) = \{pet, shop\}$. We say class X = class Y if and only if $id(X) = id(Y)$. The following is the procedure for an ontology extraction:

Step 1. Lexical entry extraction

- Use $tf * idf$ to get a set of keywords (e.g., we use 150 keywords for each topic) from the training set;
- Select primitive objects (terms) from the set of keywords using the existing background knowledge, where each term is a group of keywords, e.g., term "pet" may include $\{pet, dog, cat\}$;

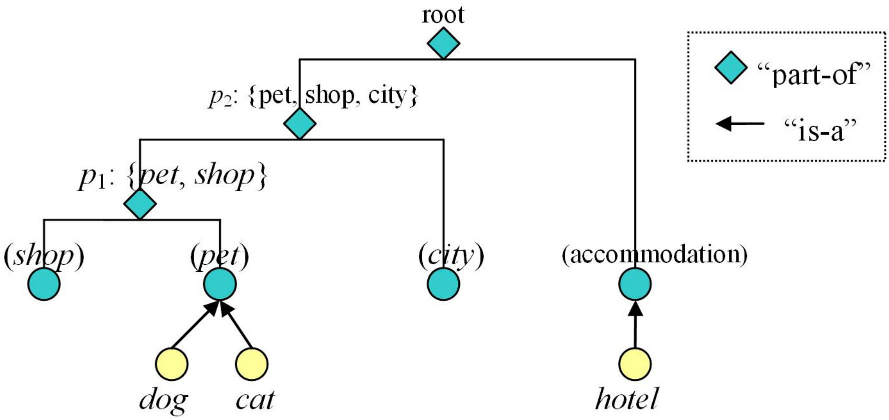


Fig. 1. Backbone of the Ontology

Step 2. Determine patterns

- Decide the *id* (a set of terms) for all relevant documents in the training sets;

Step 3. Generate a graph representation as one shown in Fig. 1.

Let $PL = \{(p_1, N_1), (p_2, N_2), \ldots, (p_n, N_n)\}$ be a set of compound objects which comes from the discovered patterns, Ω be the set of its classes: primitive or compound, where pi is a pattern $(1 \leq i \leq n)$ and N_i denotes the number of appearance of the similar objects. For example, in Fig. 1 we have $PL = \{(p_1, 2), (p_2, 1)\}$ because of $d_1 \rightarrow p_1$, $d_2 \rightarrow p_1$ and $d_3 \rightarrow p_2$. From PL we can get a support function, which satisfies:

$$support : P \rightarrow [0, 1]$$

such that

$$support(p_i) = \frac{N_i}{\sum_{(p_j, N_j) \in PL} N_j}, \text{ where } P = \{p | (p, N) \in PL\}. \tag{1}$$

We can obtain the following set-valued mapping to describe the knowledge implied in PL:

$$\Gamma : P \rightarrow \Omega; \text{ such that } \Gamma(p_i) = \begin{cases} X & \text{if } p_i \text{ is related to class } X \\ \Omega_{root} & \text{otherwise,} \end{cases} \tag{2}$$

where, Ω_{root} is the root class in Ω. We call Γ a *deploying mapping* of P on Ω. If the users could not get a corresponding class for a pattern, the pattern is indexed by default in the root class in the ontology (Note: in "is-a" taxonomy we often use empty sets). This convention makes sense if we assume that the root class represents the entire collection we discuss.

Let Θ be the set of primitive objects. We can get an *id* mapping from the deploying mapping:

$$\xi : P \to 2^{\Theta} - \{\emptyset\}, \text{ such that } \xi(p_i) = id(\Gamma(p_i)). \tag{3}$$

At last, we can obtain a probability functions pr_ξ to represent the discovered knowledge on the ontology, which satisfies:

$$pr_\xi(\theta) = \sum_{\emptyset \neq A \subseteq \Theta, \theta \in A} \frac{support(\{(p,N)|(p,N) \in PL, \xi(p) = A\})}{|A|} \tag{4}$$

for all $\theta \in \Theta$.

3.2 Decision Rules

Similar to *type rules* used for programming language processing, in this research we use decision rules on ontology. The main objective here is to make decisions for new incoming objects. Let p be a pattern and o be a new incoming object. Our basic assumption is that o should be relevant if $id(p) \subseteq id(o)$. The set of all objects o in the set of new incoming objects such that $id(p) \subseteq id(o)$ is called the *covering set* for p and denoted as $[p]$. The **positive region** (POS) is the union of all covering sets for all $p \in P$.

Except POS there are many boundary objects o such that $\exists p \in P \Rightarrow id(p) \cap id(o) \neq \emptyset$. The set of all objects o in the set of new incoming objects such that $\exists p \in P \Rightarrow id(p) \cap id(o) \neq \emptyset$ is called the **boundary region** (BND). Also, the set of all objects o in the set of new incoming objects such that $\forall p \in P \Rightarrow id(p) \cap id(o) = \emptyset$ is called the **negative region** (NEG). Given an object o, the decision rules can be determined naturally as follows:

$$\frac{\exists p \in P \Rightarrow id(p) \subseteq id(o) \neq \emptyset}{o \in POS}, \quad \frac{\exists p \in P \Rightarrow id(p) \cap id(o) \neq \emptyset}{o \in BND}, \text{ and}$$

$$\frac{\forall p \in P \Rightarrow id(p) \cap id(o) = \emptyset}{o \in NEG}.$$

3.3 Filtering Algorithm

It must be more interesting to determine thresholds theoretically rather than using experiments. The following is the idea for determining the thresholds.

The probability function pr_ξ on (see Equation 4) has the following property:

$$\sum_{t \in id(o)} pr_\xi(t) \geq \min_{p \in P}\{\sum_{t \in \xi(p)} pr_\xi(t)\} \quad \text{for all } o \in pos. \tag{5}$$

From the above analysis, we can use $\min_{p \in P}\{\sum_{t \in \xi(p)} pr_\xi(t)\}$ as the threshold. A very important conclusion we can draw from the above analysis is that our method can guarantee the processing of filtering can retrieve all positive documents (i.e., POS).

Term frequency is a very useful source in information filtering. In order to use term frequency, the *id* mapping ξ in Equation 3 can be extended to the following mapping β, which satisfies:

$$\beta : P \rightarrow 2^{\Theta\times[0,1]} - \{\emptyset\} \quad \text{such that}$$

$$\sum\nolimits_{(fst,snd)\in\beta(p_i)} snd = 1 \quad \text{and} \quad \xi(p_i) = \{fst|(fst,snd) \in \beta(p_i)\} \qquad (6)$$

for all pattern $p_i \in P$ (please see the example below). We call β a frequency distribution of ξ.

Using the frequency distribution, we can refine probability functions pr_ξ by obtaining another probability functions pr_β on Θ, which satisfies:

$$pr_\beta(\theta) = \sum\nolimits_{(p_i,N_i)\in PL,(\theta,snd)\in\beta(p_i)} support((p_i, N_i)) \times snd \qquad (7)$$

for all $\theta \in \Theta$.

Using the example in Fig. 1 we have $PL = \{(p_1, 2), (p_2, 1)\}$, and $\xi(p_1) = \{pet, shop\}$ and $\xi(p_2) = \{pet, shop, city\}$. Assume $support(p_1) = 2/3$ and $support(p_2) = 1/3$, then we have the corresponding frequency distribution β, which satisfies (notice: $d_1 \rightarrow p_1$, $d_2 \rightarrow p_1$ and $d_3 \rightarrow p_2$):

$$\begin{aligned}\beta(p_1) &= \{(pet, (4 + 5) \div (4 + 5 + 6 + 15)), (shop, (6 + 15) \div (4 + 5 + 6 + 15))\}\\ &= \{(pet, 0.3), (shop, 0.7)\}, \text{ and}\\ \beta(p_2) &= \{(pet, 3 \div (3 + 7 + 10)), (shop, 7 \div (3 + 7 + 10)),\\ &\quad (city, 10 \div (3 + 7 + 10))\} = \{(pet, 0.15), (shop, 0.35), (city, 0.5)\}.\end{aligned}$$

The new filtering algorithm first updates the ontology in Fig. 1 by replacing p_1 and p_2 with "$p_1 : \{(pet, 0.3), (shop, 0.7)\}$" and "$p_2 : \{(pet, 0.15), (shop, 0.35), (city, 0.5)\}$", respectively. It then determines a threshold (see Equation 5). It also extracts a set of terms from each new incoming document. At last it calculates the probability of the document (see Equation 7) and makes a decision according to the threshold.

4 Sophisticated Data Processing

Different from the term based filtering phase, in the phase of Sophisticated Data Processing we use phrases, a sort of sequential patterns.

It is not very difficult for the discovery of phrases from documents if we view each paragraph as a transaction. The problem is how to represent the relations between phrases. One method is to use a document index graph (DIG) [5], where each node is a unique word, and each edge is a two adjacent nodes which appear successive in a document. The drawback of this method is that a DIG may index some nonsense phrases. In this research we use sequential patterns [1] to represent phrases. We also use a new concept, which is similar to the notation of closed sequential patterns [16,21], to create a phrase taxonomy model (PTM) for Web mining.

4.1 Phrase Taxonomy Model

A sequence $s =< x_1, \ldots, x_m >$ ($x_i \subseteq T$ is a termset) is an ordered list. A sequence $\alpha =< a_1, \ldots, a_m >$ is a sub-sequence of another sequence $\beta =< b_1, \ldots, b_n >$, denoted by $\alpha \subseteq \beta$, if and only if $\exists i_1, \ldots, i_m$ such that $1 \leq i_1 < i_2 \ldots < i_m \leq n$ and $\alpha_1 \subseteq \beta_{i_1}, \alpha_2 \subseteq \beta_{i_2}, \ldots, \alpha_m \subseteq \beta_{i_m}$. A sequential pattern s is a *very closed sequential pattern* of s' if $s \subseteq s'$ and $support(s) - support(s') < \lambda \times support(s')$, where λ is a small positive decimal.

The above definitions can be used to create a phrase taxonomy as showed in Fig. 2, where a, b, c, and d are terms, the arrows are "is-a" relation, e.g., phrase $< (a)(b) >$ is a sub-sequence of $< (a)(b)(c) >$.

If we use the frequency to define the *support* function for all patterns, we have $support(< (a)(b) >) \geq support(< (a)(b)(c) >)$. In general we may get 3 sub-sequence patterns of $< (a)(b)(c) >$. They are $< (a)(b) >$, $< (a)(c) >$ and $< (b)(c) >$. In our phrase taxonomy we remove the not very closed sequential patterns if their supports are very closed to their father, e.g., we have pruned $< (a)(c) >$ in Fig. 2.

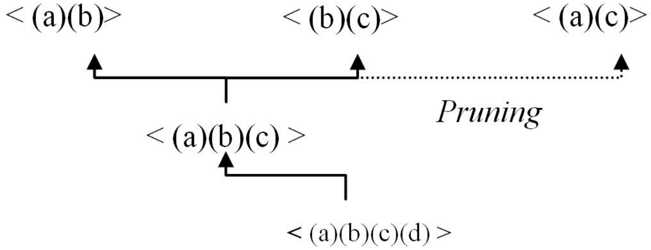


Fig. 2. Phrase taxonomy

After we have extracted a phrase taxonomy using a training set, we can use it to calculate $pr(d)$, the relevance degree of d for a given topic, for each new incoming document d. The draft procedure of making decisions is described as follows:

1. Find all longest patterns in document d; // e.g., $(< (a)(b)(c) >)$ is a longest pattern if $(< (a)(b)(c)(d) >)$ does not appear in d.
2. Determine $pr(d)$ according to the taxonomy. // e.g., $pr(d) = support(< (a)(b)(c) >) + support(< (a)(b) >) + support(< (b)(c) >)$.

5 Evaluations

The standard TREC (Text REtrieval Conference) test collections Reuters RCV1 (Reuters Corpus Volume 1) was used to test the effectiveness of the proposed model. RCV1 corpus consists of all and only English language stories produced by Reuter's journalists between August 20, 1996, and August 19, 1997 with total

806,791 documents. It is used by TREC in recent years for the adaptive filtering track. TREC has developed and provided 100 topics for the filtering track aiming at building a robust filtering system [19]. The first fifty of these were constructed by human researchers and the rest by intersecting two Reuters' topic categories. Each topic is divided into two sets: training and test set. In this paper, we only use the first fifty topics for our experimental test.

The document relevance judgments have been given for each topic. This means that every document is assigned to be either positive or negative. "Positive" means the document is relevant to the assigned topic; otherwise "negative" will be given to the document. The set of 50 TREC topics is used to represent the diverse Web user's information needs. The experiments simulated user feedback by assuming that the user would recognize as relevant the chosen some documents that were officially judged as relevant from a set of given documents.

5.1 Performance Measures

We measure the effectiveness of our model with three methods. They are F_β-measure, Average Precision (AP) and the P/R breakeven point measures. F_β is a version of the van Rijsbergen measure of retrieval performance. This measure is a function of *Recall* (R) and *Precision* (P), together with a free parameter beta which determines the relative weighting of recall and precision. It is calculated by the following function:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R};$$

The parameter $\beta = 1$ is used for our study, which means that recall and precision is weighted equally. Average precision is a hybrid measure of average precision and recall. The P/R breakeven point indicates the value at which precision equals recall. The larger a P/R breakeven point or average precision or F_β-measure score is, the better the system performs.

5.2 Results

Wu *et. al.* [23] used the pattern taxonomy rather than single words to represent documents. They have conducted experiments on TREC collections (RVC1 corpus) and have compared the performance of their model with keyword based models. They concluded that their method outperforms the keyword based method. Therefore, the patter taxonomy model (PTM) based system will be the baseline for this study.

The results of our experiments reported in Fig. 3 are the average scores of P/R breakeven point, average precision and F_β-measure for both methods calculation of scores of performs for the first 50 topics of all 100 TREC topics. Our method is called Web Information Gathering System (WIGS). It demonstrated that the performances of WIGS are better than the original PTM system. This improvement is significant and also it is consistent for all the experiments.

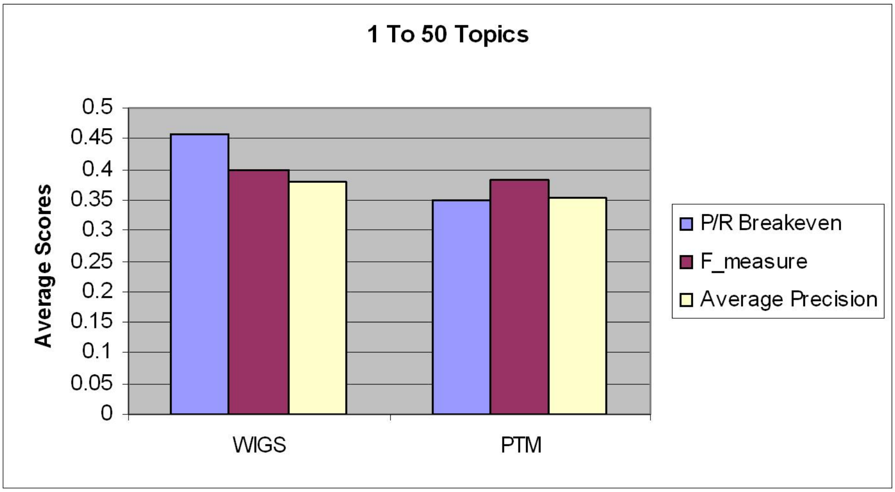


Fig. 3. The P/R breakeven point, Average Precision and F-measure for WIGS and PTM

6 Related Work

As mentioned in the introduction, currently Web mining can be classified into four categories: Web usage mining, Web structure mining, Web user profile mining and Web content mining [4,15,20,11]. The obvious difference between Web mining and data mining is that the former is based on Web-related data sources, such as unstructured documents (e.g., HTML), semi-structured documents(e.g., XML), log, services and user profiles; and the latter is based on databases.

Association mining has been used in Web text mining, which refers to the process of searching through unstructured data on the Web and deriving meaning from. The main purposes of text mining were association discovery, trends discovery, and event discovery. The association between a set of keywords and a predefined category (e.g., a term) can be described as an association rule. The trends discovery means the discovery of phrases, a sort of sequence association rules. The event discovery is the identification of stories in continuous news streams. Usually clustering based mining techniques can be used for such a purpose. It was also necessary to combine association rule mining with the existing taxonomies in order to determine useful patterns.

The disadvantage of association rule mining is that there are too many discovered patterns that make the application of the discovered knowledge inefficient. Also there are many noise patterns that make the discovered knowledge contains much uncertainties. Although pruning non-closed patterns that can improve the quality of association mining in text mining in some extents [22,23], the performance of text mining systems are still ineffectively.

Granule mining [10] can be an alternative solution to specify association rules, where a granule is kind of representation of a group of objects (transactions) that satisfy user constraints, e.g., all objects have the same attributes' values. Decision tables [14] can be a basic structure for granule mining in which attributes are divided into two groups: condition attributes and decision attributes. However, there exists ambiguities whist we use the decision rules for Web information gathering. We have demonstrated in the previous sections that rough association rule mining can be used to overcome these disadvantages.

The basic architecture we used to implement the above idea is to automatically construct and maintain an ontology for representation, application and updating of discovered knowledge. The related work for this architecture is about ontology learning algorithms. Several ontology learning algorithms have been presented such as pattern matching, hierarchical clustering and pattern taxonomy [22].

7 Conclusions

We have presented a novel approach that used two information processing phases (filtering and sophisticated data processing) for Web information gathering. We have set up an ontology-based method for the phase of information filtering, which is sound based on the rough set based decision rules. An efficient filtering algorithm is also presented and tested. For sophisticated data process, PTM based solution is provided and the experimental results are promising. The significant contribution is that this research presents a promising solution for solving the two fundamental issues in Web information gathering.

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In: ICDE95, pp. 3–14
2. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Reading (1999)
3. Chang, C.H., Hsu, C.C.: Enabling concept-based relevance feedback for information retrieval on the WWW. IEEE Transactions on Knowledge and Data Engineering 11(4), 595–609 (1999)
4. Garofalakis, M.N., Rastogi, R., Seshadri, S., Shim, K.: Data mining and the Web: past, present and future. In: WIDM99, pp. 43–47
5. Hammouda, K.M., Kamel, M.: SPhrase-based document similarity based on an index graph model. In: ICDM02, pp. 203–210
6. Hull, D.A., Roberston, S.: The TREC-8 filtering track final report. In: TREC-8 (1999)
7. Jones, K.S.: Information retrieval and artificial intelligence. Artificial Intelligence 114(1-2), 257–281 (1999)
8. Li, Y., Zhang, C., Zhang, S.: Cooperative strategy for Web data mining and cleaning. Applied Artificial Intelligence 17(5-6), 443–460 (2003)
9. Li, Y., Zhong, N.: Web mining model and its applications on information gathering. Knowledge-Based Systems 17, 207–217 (2004)
10. Li, Y., Yang, W., Xu, Y.: Multi-Tier Granule Mining for Representations of Multidimensional Association Rules. In: Perner, P. (ed.) Advances in Data Mining. LNCS (LNAI), vol. 4065, pp. 953–958. Springer, Heidelberg (2006)

11. Li, Y., Zhong, N.: Mining ontology for automatically acquiring Web user information needs. IEEE Transactions on Knowledge and Data Engineering 18(4), 554–568 (2006)
12. Li, Y., Zhong, N.: Mining Rough Association from Text Documents. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 368–377. Springer, Heidelberg (2006)
13. Li, Y., Zhong, N.: Rough Association Rule Mining in Text Documents for Acquiring Web User Information Needs. In: IEEE/WIC/ACM International Conference on Web Intelligence, WI06, pp. 226–232 (2006)
14. Li, Y., Zhong, N.: Interpretations of association rules by granular computing. In: 3rd IEEE International Conference on Data Mining (ICDM03), pp. 593–596 (2003)
15. Madria, S.M., Bhowmick, S.S., Ng, W.K., Lim, E.-P.: Research issues in Web data mining. In: Zaki, M.J., Ho, C.-T. (eds.) KDD99, LNCS (LNAI), vol. 1759, pp. 303–312. Springer, Heidelberg (2000)
16. Mobasher, B., Dai, H., Luo, T., Sun, Y., Zhu, J.: Combining Web usage and content mining for more effective personalization. In: International Conference on Ecommerce and Web Technologies (2000)
17. Pal, S.K., Talwar, V.: Web mining in soft computing framework: relevance, state of the art and future directions. IEEE Transactions on Neural Networks 13(5), 1163–1177 (2002)
18. Robertson, S., Hull, D.A.: The TREC-9 filtering track final report, TREC-9 (2000)
19. Rose, T., Stevenson, M., Whitehead, M.: The Reuters Corpus Volume 1 - From yesterday's news to today's language resources. In: International Conference on Language Resources and Evaluation (2002)
20. Srivastava, J., Cooley, R., Deshpande, M., Tan, P.-N.: Web usage mining: Discovery and applications of usage pattern from Web data. SIGKDD Explorations 1(2), 1–12 (2000)
21. Yan, X., Han, J., Afshar, R.: CloSpan: Mining closed sequential patterns in large datasets. In: SDM2003, pp. 166–177 (2003)
22. Wu, S.-T., Li, Y., Xu, Y.: Deploying Approaches for Pattern Refinement in Text Mining. In: Perner, P. (ed.) ICDM 2006. LNCS (LNAI), vol. 4065, pp. 1157-1161. Springer, Heidelberg (2006)
23. Wu, S.-T., Li, Y., Xu, Y., Pham, B., Chen, P.: Automatic pattern taxonomy extraction for Web mining. In: IEEE/WIC/ACM International Conference on Web Intelligence (WI04), China, pp. 242–248 (2004)

An Investigation of Human Problem Solving System: Computation as an Example

Shinichi Motomura[1], Akinori Hara[1], Ning Zhong[1,2], and Shengfu Lu[2]

[1] Department of Life Science and Informatics
Maebashi Institute of Technology, Japan
[2] The International WIC Institute
Beijing University of Technology, China
motomura@maebashi-it.org

Abstract. Although human problem solving has been investigated in a behavior based approach, it has been recognized that ignoring what goes on in human brain and focusing instead on behavior has been a large impediment to understand how human being does complex adaptive, distributed problem solving and reasoning. In the paper, we propose a methodology for investigating human problem solving process by combining ERP mental arithmetic tasks, as a case study, with multi-aspect data analysis. Preliminary results show the usefulness of our methodology.

1 Introduction

Problem-solving is one of main capabilities of human intelligence and has been studied in both cognitive science and AI [9], where it is addressed in conjunction with reasoning centric cognitive functions such as attention, control, memory, language, reasoning, learning, and so on, using a logic based symbolic and/or connectionist approach. Although logic based problem-solving is "perfect", mathematical systems with no real time and memory constraints, Web-based problem-solving systems need real-time and dealing with global, multiple, huge, distributed information sources.

Furthermore, in order to develop a Web based problem-solving system with human level capabilities, we need to better understand how human being does complex adaptive, distributed problem solving and reasoning, as well as how intelligence evolves for individuals and societies, over time and place [3,11,12,13,17]. In other words, ignoring what goes on in human brain and focusing instead on behavior has been a large impediment to understand how human being does complex adaptive, distributed problem solving and reasoning.

In the light of Brain Informatics [16,17], we need to investigate specifically the following issues:

- What are the existing problem-solving models in AI, cognitive science, and neuroscience?
- How to design fMRI/EEG experiments and analyze such fMRI/EEG data to understand the principle of human problem solving in depth?

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 824–834, 2007.

- How to build the cognitive model to understand and predict user profile and behavior in a problem solving process?
- How to implement human-level problem solving on the Web based portals that can serve users wisely?

As a result, the relationships between classical problem solving and biologically plausible problem solving need to be defined and/or elaborated [17].

As a step in this direction, we observe that fMRI brain imaging data and EEG brain wave data extracted from human problem solving system are peculiar ones with respect to a specific state or the related part of a stimulus. Based on this point of view, we propose a way of peculiarity oriented mining (POM) for knowledge discovery in multiple human brain data, without using conventional imaging processing to fMRI brain images and frequency analysis to EEG brain waves. The proposed approach provides a new way for automatic analysis and understanding of fMRI brain images and EEG brain waves to replace human-expert centric visualization. The mining process is a multi-step one, in which various psychological experiments, physiological measurements, data cleaning, modeling, transforming, managing, and mining techniques are cooperatively employed to investigate human problem solving system.

The rest of the paper is organized as follows. Section 2 provides a mining process for multi-aspect human brain data analysis of human problem solving system. Sections 3 and 4 explain how to design the experiment of an ERP mental arithmetic task with visual stimuli, and describe how to do multi-aspect analysis in the obtained ERP data, respectively, as an example to investigate human problem solving and to show the usefulness of the proposed mining process. Finally, Section 5 gives concluding remarks.

2 A Mining Process for Multi-aspect Human Brain Data Analysis

The future of Brain Informatics will be affected by the ability to do large-scale mining of fMRI and EEG brain activations. The key issues are how to design the psychological and physiological experiments for obtaining various data from human problem solving system, as well as how to analyze and manage such data from multiple aspects for discovering new models of human problem solving system. Although several human-expert centric tools such as SPM (MEDx) have been developed for cleaning, normalizing and visualizing the fMRI images, researchers have also been studying how the fMRI images can be automatically analyzed and understood by using data mining and statistical learning techniques [4,6,10,11,15]. We are concerned with how to extract significant features from multiple brain data measured by using fMRI and EEG in preparation for multi-aspect data mining that uses various data mining techniques for analyzing multiple data sources.

A mining process is shown in Figure 1, in which various tools can be cooperatively used in the multi-step process for pre-processing (data cleansing, modeling and transformation), mining and post-processing. Our purpose is to understand activities of human problem solving system by investigating the spatiotemporal features and flow of human problem solving system, based on functional relationships between activated areas of human brain for each given task; More specifically, at the current stage, we want to understand:

- how a peculiar part (one or more areas) of the brain operates in a specific time;
- how the operated part changes along with time;
- how the activated areas work cooperatively to implement a whole problem solving system;
- how the activated areas are linked, indexed, navigated functionally, and what are individual differences in performance.

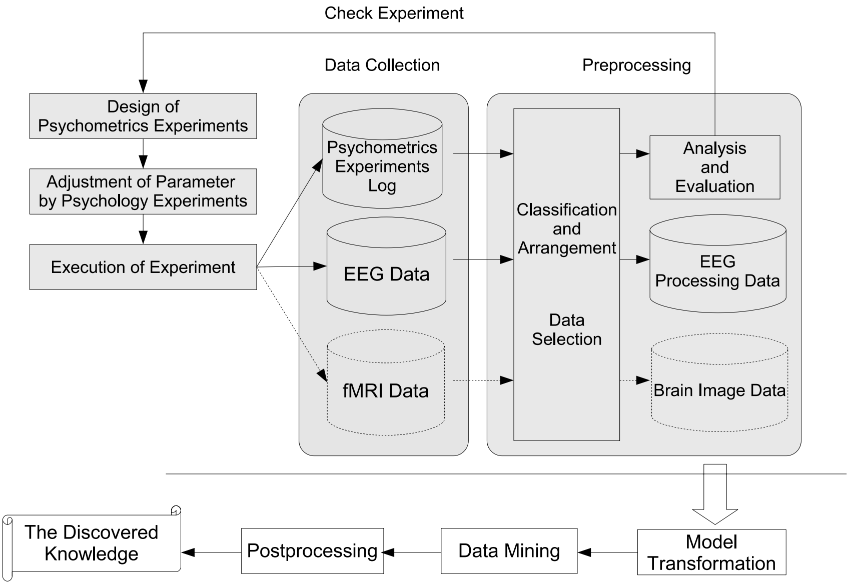


Fig. 1. The mining process

3 The Experiment of an ERP Mental Arithmetic Task with Visual Stimuli

In this work, the ERP (event-related potential) human brain waves are derived by carrying out a mental arithmetic task with visual stimuli, as an example to investigate human problem solving process. ERP is a light, sound, and brain potential produced with respect to the specific phenomenon of spontaneous movement [2]. Since the potential is very weak, the same stimulus can be repeated and given, and furthermore addition average processing can be performed. It argues about ERP in time until a certain wave-like peak appears from a stimulus presentation time called P300. This is called latent time and various knowledge about the mental activity whose measurement is impossible is acquired from outside.

3.1 Outline of Experiments

The experiment conducted this time shows a numerical calculation problem to a subject, and asks the subject to solve it in mental arithmetic, and the shown sum has hit, or it pushes a button, and performs a judging of corrigenda. The form of the numerical calculation to be shown is the addition problem of "augend + addend = sum". The wrong sum occurs at half the probability, and the distribution is not uniform. Figure 2 gives an example of the screen state transition. Type 1 is two digits addition. Type 2 is eight numbers appear, but it is not necessary to calculate. Both of them, the figure does not remain on the screen.

	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8
1	1	9	+	2	6	=	4	5
2	6	3	5	8	1	2	7	4

Type 1 : 19+26=45
Type 2 : random number

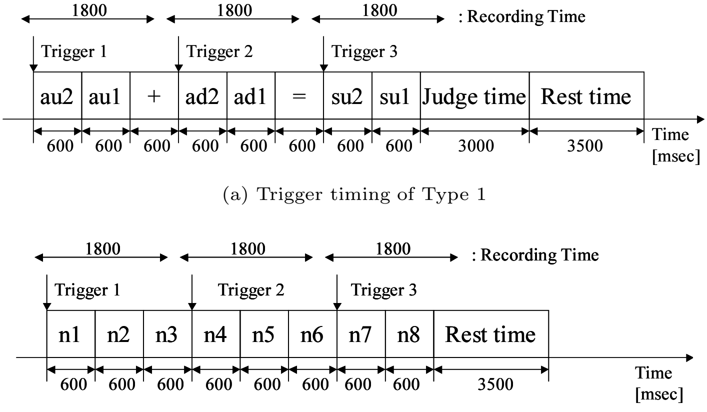
Fig. 2. Example of the screen state transition

3.2 Visual Stimuli

In the experiments, three states (tasks), namely, *visual on-task*, *visual off-task*, and *no-task*, exist by the difference in the stimulus given to a human subject. *Visual on-task* is the state which is calculating by looking a number. *Visual off-task* is the state which is looking the number that appears at random. *No-task* is the relaxed state which does not work at all. We try to compare and analyze how brain waves change along with the different tasks stated above.

3.3 Trigger Signal and Timing Chart

It is necessary to measure EEG relevant to a certain event to the regular timing in measurement of ERP repeatedly. In this research, since the attention was paid to each event of augend, addend, and sum presentation in calculation activities, three trigger signals with respect to these events were set up, respectively. Pretrigger was set to 200 [msec], and addition between two digits are recorded in 1800 [msec], respectively. Figure 3 gives an example of the time chart for a two-digit addition and off-task. "au" is augend, "ad" is addend, and "su" is sum. "n" is the random number (1-digits). Therefore "au2" is MSD (last 2-digits) of augend, and "au1" is LSD (last 1-digits) of augend.



(a) Trigger timing of Type 1

(b) Trigger timing of Type 2

Fig. 3. Trigger timing

3.4 Experimental Device and Measurement Conditions

Electroencephalographic activity was recorded using a 64 channel BrainAmp amplifier (Brain Products, Munich, Germany) with a 32 electrode cap as shown in Figure 4, which are based on an extended international 10-20 system. The channels with the mark of double circles in Figure 4 will be mainly discussed with their ERP data in this paper. Furthermore, two additional channels, eye movement measurement (2ch) and trigger signal detection (3ch), are also used.

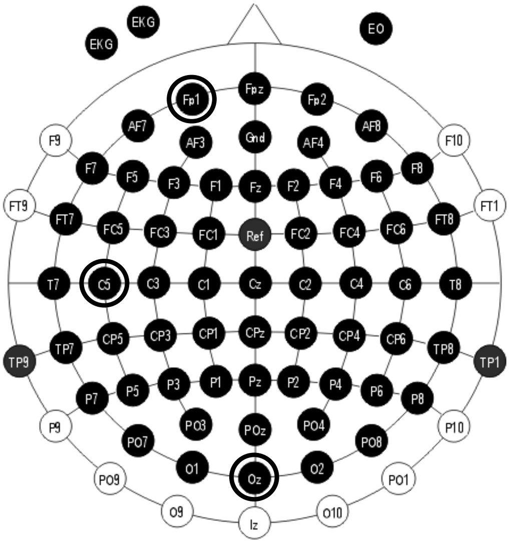


Fig. 4. EEG cap electrode

The electrode adopted for this experiment is the cap electrode and is used as a standard electrode for measuring both earlobes. The sampling frequency is 2500Hz to be processed. The number of experimental subjects is 20.

4 Multi-aspect Data Analysis

4.1 ERP

For the measured EEG data, a maximum of 40 addition average processing were performed, and the ERP was derived by using Brain Vision Analyzer (Brain Products, Munich, Germany). Generally speaking, the Wernicke's area of a left temporal lobe and the prefrontal area are related to the calculation process [5]. In this study, we compare calculation activities and non-calculation activities by focusing on some important channels (Fp1, C5, Oz). We pay attention to recognition of the number, short-term memory and attentiveness, as well as compare ERP of Type 1 and Type 2, and study a problem solving process for a calculation in a macro view.

Figure 5 shows the ERPs in channels Fp1, C5, Oz. First, we discuss the channel Fp1, which is the prefrontal area related with attention. The presence of the calculation activity is closely related to the depth of attention to the number. The activation of Fp1 is earlier than that of the visual area, and it appears remarkably with On-task. Next, we discuss the channel Oz, which is the visual related area with respect to the gaze. We can see that it is activated by a numerical appearance, regardless of the presence of the calculation activity. However, Type 2 shows high positive potential in all almost time. And, it is expected that an appearance of P400 in Oz is related to processing and memory of an afterimage in a brain. Hence, it is necessary to investigate this phenomenon deeply with the temporal change around the visual area. Finally, we discuss the channel C5, which is part of the left temporal lobe with respect to the logical interpretation. Contrary to our expectation, the difference of clear ERP to the presence of the calculation activity was not found. It is guessed that what number appeared without any relation to the calculation is unconsciously confirmed.

Furthermore, we pay attention to the calculated time zone and study response in each part. In this time zone, we can see that the prefrontal area related channels (Fp1, Fp2, AF8, F10) are activated. An interesting point we observed is that the behavior of the left and right brain when calculating is with some individual differences. Let us to analyze this phenomenon from the topography with respect to Trigger 2, as shown in Figure 6. After displayed LSD, the display time zone is from 200 to 320 milliseconds. We can see that the potential distribution of the left and right brain is different between subjects, and subject A used the whole brain thoroughly. We try to understand it in depth by analyzing influence by good of the calculation and not good as well as how to solve problems, from the viewpoint of multi-aspect mining.

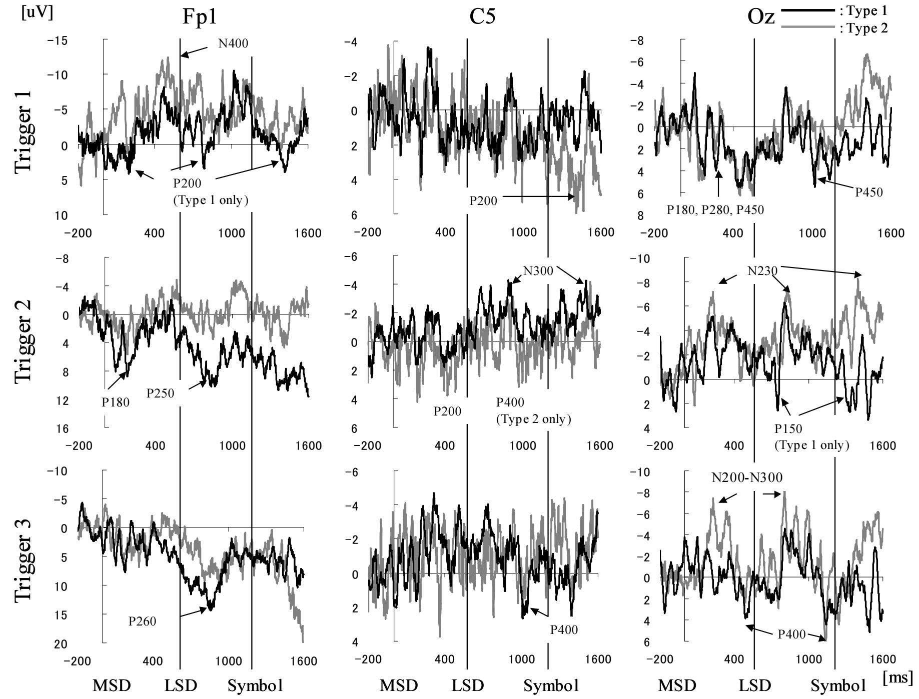


Fig. 5. ERP (Fp1,C5,Oz)

4.2 Peculiarity Oriented Mining

It is clear that ERPs are different for channels over the time. Although detecting the concavity and convexity (P300 etc.) is easy by using the existing tool, it is difficult to find a peculiar one in the multiple channels with the concavity and convexity [7,8]. In order to discover new knowledge of human information processing activities, it is necessary to pay attention to the peculiar channel and time in ERPs for investigating the spatiotemporal features and flow of human information processing system. This subsection introduces our *peculiarity oriented mining (POM)* approach for ERP data analysis.

POM in the Attribute-Value Level. The main task of POM is the identification of peculiar data. An attribute-oriented method, which analyzes data from a new view and is different from traditional statistical methods, is recently proposed by Zhong *et al.* and applied in various real-world problems [14,15].

Peculiar data are a subset of objects in the database and are characterized by two features: (1) very different from other objects in a dataset, and (2) consisting of a relatively low number of objects. The first property is related to the notion of distance or dissimilarity of objects. Intuitively speaking, an object is different from other objects if it is far away from other objects based on certain distance functions. Its attribute values must be different from the values of other objects.

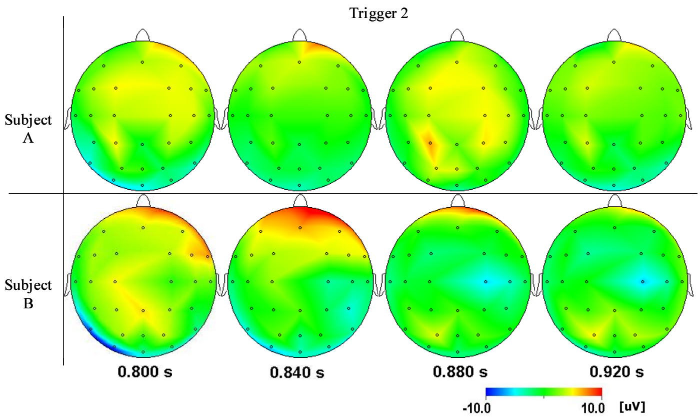


Fig. 6. Topography

One can define distance between objects based on the distance between their values. The second property is related to the notion of support. Peculiar data must have a low support.

At the attribute-value level, the identification of peculiar data can be done by finding attribute values having properties (1) and (2). Let x_{ij} be the value of attribute A_j of the i-th tuple in a relation, and n the number of tuples. Zhong *et al.* [14] suggested that the peculiarity of x_{ij} can be evaluated by a *Peculiarity Factor*, $PF(x_{ij})$,

$$PF(x_{ij}) = \sum_{k=1}^{n} N(x_{ij}, x_{kj})^{\alpha} \tag{1}$$

where N denotes the conceptual distance, α is a parameter to denote the importance of the distance between x_{ij} and x_{kj}, which can be adjusted by a user, and $\alpha = 0.5$ as default.

Based on the peculiarity factor, the selection of peculiar data is simply carried out by using a threshold value. More specifically, an attribute value is peculiar if its peculiarity factor is above minimum peculiarity p, namely, $PF(x_{ij}) \geq p$. The threshold value p may be computed by the distribution of PF as follows:

$$\begin{aligned} threshold = mean\ of\ PF(x_{ij}) + \\ \beta \times standard\ deviation\ of\ PF(x_{ij}) \end{aligned} \tag{2}$$

where β can be adjusted by a user, and $\beta = 1$ is used as default. The threshold indicates that a data is a peculiar one if its PF value is much larger than the mean of the PF set. In other words, if $PF(x_{ij})$ is over the threshold value, x_{ij} is a peculiar data. By adjusting the parameter β, a user can control and adjust threshold value.

Peculiarity Vector Oriented Mining. Unfortunately, the POM in the attribute-value stated above is not fit for ERP data analysis. The reason is that the useful aspect for ERP data analysis is not amplitude, but the latent time. After smoothing enough by moving average processing, in the time series, we pays the attention to each potential towards N pole or P pole. Furthermore, the channel with the direction different from a lot of channels is considered to be a peculiar channel at that time. Hence, the distance between the attribute-values is expressed at the angle. And this angle can be obtained from the inner product and the norm in the vector. Let inclination of wave i in a certain time t be x_{it}. The extended PF corresponding to ERP can be defined by the following Eq. (3).

$$PF(x_{it}) = \sum_{k=1}^{n} \theta(x_{it}, x_{kt})^{\alpha}. \tag{3}$$

However, θ in Eq. (3) is an angle which the wave in time t makes. For the θ, we can compute for an angle using Eq. (4).

$$cos\theta = \frac{1 + x_{it} \cdot x_{kt}}{\sqrt{1 + x_{it}^2}\sqrt{1 + x_{kt}^2}}. \tag{4}$$

Application of the Extended POM Method. The extended POM method has been used for the ERP data analysis. Figure 7 shows a result in which the peculiarity in ERP data with respect to addition Type 1 (between 2 digits with the visual stimulus Trigger 1) is presented. All channels show high peculiarity from 200ms to 400ms after presented stimuli. The reason is that a wavy ruggedness of ERP changes violently, and it is a remarkable response in the frontal cortex and lobus occipitalis. On the other hand, the PF values of temporal lobes, such as C5, C6 and P6 are high in all time zones. These channels have a unique property of latent time. The higher PF value is related to the interestingness in the ERP data. Hence, it is necessary to investigate this phenomenon deeply with the medical standpoint.

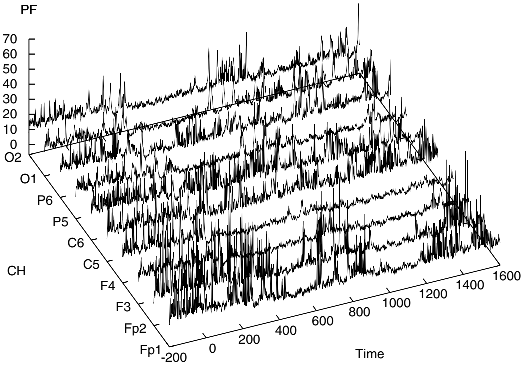


Fig. 7. Peculiarity in ERP data

5 Conclusion

In this paper, we described a more whole process from the design of the ERP experiments of a mental arithmetic task with visual stimuli, carrying out such an experiment to collect the EEG data, to multi-aspect EEG data analysis by using the proposed Peculiarity Vector Oriented Mining method etc. Some preliminary results showed the usefulness of our methodology. By introducing multiple trigger signals, it is possible to analyze human calculation process in detail, as a case study for investigating human problem solving system. The previous design of ERP experiments only gave simple stimulus (e.g., sound stimulus, light stimulus, etc.) and very few of them are with respect to investigating a more whole human problem solving mechanism.

Our future work includes obtaining and analyzing more subject data, combining with fMRI human brain image data for multi-aspect analysis in various approaches of data mining and reasoning.

References

1. Gazzaniga, M.S. (ed.): The Cognitive Neurosciences III. MIT Press, Cambridge (2004)
2. Handy, T.C.: Event-Related Potentials, A Methods Handbook. MIT Press, Cambridge (2004)
3. Liu, J., Jin, X., Tsui, K.C.: Autonomy Oriented Computing: From Problem Solving to Complex Systems Modeling. Springer, Heidelberg (2005)
4. Megalooikonomou, V., Herskovits, E.H.: Mining Structure-Function Associations in a Brain Image Database. In: Cios, K.J. (ed.) Medical Data Mining and Knowledge Discovery, pp. 153–179. Physica-Verlag, Heidelberg (2001)
5. Mizuhara, H., Wang, L., Kobayashi, K., Yamaguchi, Y.: Long-range EEG Phase-synchronization During an Arithmetic Task Indexes a Coherent Cortical Network Simultaneously Measured by fMRI. NeuroImage 27(3), 553–563 (2005)
6. Mitchell, T.M., Hutchinson, R., Niculescu, R.S., Pereira, F., Wang, X., Just, M., Newman, S.: Learning to Decode Cognitive States from Brain Images. Machine Learning 57(1-2), 145–175 (2004)
7. Nittono, H., Nageishi, Y., Nakajima, Y., Ullsperger, P.: Event-related Potential Correlates of Individual Differences in Working Memory Capacity. Psychophysiology 36, 745–754 (1999)
8. Picton, T.W., Bentin, S., Berg, P., Donchin, E., Hillyard, S.A., Johnson, R. et al.: Guidelines for using human event-related potentials to study cognition: Recording standards and publication criteria. Psychophysiology 37, 127–152 (2000)
9. Newell, A., Simon, H.A.: Human Problem Solving. Prentice-Hall, Englewood Cliffs (1972)
10. Sommer, F.T., Wichert, A. (eds.): Exploratory Analysis and Data Modeling in Functional Neuroimaging. MIT Press, Cambridge (2003)
11. Sternberg, R.J., Lautrey, J., Lubart, T.I.: Models of Intelligence. American Psychological Association (2003)
12. Yao, Y.Y.: A Partition Model of Granular Computing. In: Yao, Y.Y. (ed.) Transactions on Rough Sets. LNCS, vol. 1, pp. 232–253. Springer, Heidelberg (2004)

13. Zadeh, L.A.: Precisiated Natural Language (PNL). AI Magazine 25(3), 74–91 (2004)
14. Zhong, N., Yao, Y.Y., Ohshima, M.: Peculiarity Oriented Multi-Database Mining. IEEE Transaction on Knowlegde and Data Engineering 15(4), 952–960 (2003)
15. Zhong, N., Wu, J.L., Nakamaru, A., Ohshima, M., Mizuhara, H.: Peculiarity Oriented fMRI Brain Data Analysis for Studying Human Multi-Perception Mechanism. Cognitive Systems Research Elsevier 5(3), 241–256 (2004)
16. Zhong, N.: Building a Brain-Informatics Portal on the Wisdom Web with a Multi-Layer Grid: A New Challenge for Web Intelligence Research. In: Torra, V., Narukawa, Y., Miyamoto, S. (eds.) MDAI 2005. LNCS (LNAI), vol. 3558, pp. 24–35. Springer, Heidelberg (2005)
17. Zhong, N.: Impending Brain Informatics (BI) Research from Web Intelligence (WI) Perspective. International Journal of Information Technology and Decision Making, World Scientific 5(4), 713–727 (2006)

Author Index