Marcin Szczuka Marzena Kryszkiewicz
Sheela Ramanna Richard Jensen
Qinghua Hu (Eds.)

Rough Sets and Current Trends in Computing

LNAI 6086

7th International Conference, RSCTC 2010
Warsaw, Poland, June 2010
Proceedings

Springer

Lecture Notes in Artificial Intelligence 6086

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

Subseries of Lecture Notes in Computer Science

Marcin Szczuka Marzena Kryszkiewicz
Sheela Ramanna Richard Jensen Qinghua Hu
(Eds.)

Rough Sets and Current Trends in Computing

7th International Conference, RSCTC 2010
Warsaw, Poland, June 28-30, 2010
Proceedings

Springer

Volume Editors

Marcin Szczuka
The University of Warsaw, Poland
E-mail: szczuka@mimuw.edu.pl

Marzena Kryszkiewicz
The Warsaw University of Technology, Poland
E-mail: mkr@ii.pw.edu.pl

Sheela Ramanna
The University of Winnipeg, Canada
E-mail: s.ramanna@uwinnipeg.ca

Richard Jensen
Aberystwyth University, Wales, UK
E-mail: rkj@aber.ac.uk

Qinghua Hu
Harbin Institute of Technology, China
E-mail: huqinghua@hit.edu.cn

Library of Congress Control Number: 2010928207

CR Subject Classification (1998): I.2, H.3, I.4, F.1, I.5, H.4

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-642-13528-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-13528-6 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

This volume contains the papers selected for presentation at the 7th International Conference on Rough Sets and Current Trends in Computing (RSCTC 2010) held at the University of Warsaw, Poland, during June 28–30, 2010.

This was the seventh edition of the event that has established itself as the main forum for exchange of ideas between researchers in various areas related to rough sets. In 2010, the conference went back to its origins both in terms of its location (the first RSCTC conference was held in Warsaw, Poland in 1998) and in terms of topic coverage. The dates also mark the third anniversary of RSEISP 2007—a conference dedicated to the memory of Zdzisław Pawlak.

RSCTC 2010 was to provide researchers and practitioners interested in emerging information technologies a forum to share innovative theories, methodologies, and applications in rough sets and its extensions. In keeping with the spirit of past rough set-based conferences, RSCTC 2010 aimed to explore synergies with other closely related areas such as computational intelligence, knowledge discovery from databases and data mining, non-conventional models of computation, and Web mining. Major topics covered in these proceedings include: approximate and uncertain reasoning, bioinformatics, data and text mining, dominance-based rough set approaches, evolutionary computing, fuzzy set theory and applications, logical and mathematical foundations of rough sets, perceptual systems as well as applications of rough sets and its extensions in areas such as medicine, Web intelligence and image processing. The papers included in the special and industrial sessions cover learning methods and mining of complex data, soft computing applications to multimedia and telemedicine, knowledge representation and exchange in multi-agent systems and emerging intelligent technologies in the telecommunications industry.

There were 148 valid (out of 163 in total) submissions to RSCTC 2010. Every paper was examined by at least two reviewers. Out of the papers initially selected, some were approved subject to revision and then additionally evaluated. Finally, 76 papers were accepted, giving an acceptance ratio of just over 51% for the conference.

The Discovery Challenge was launched successfully this year, with support and sponsorship from TunedIT (http://tunedit.org). The data-mining task concerned feature selection for the analysis of DNA microarray data and classification of patients for the purpose of medical diagnosis and treatment. We especially wish to acknowledge the following Discovery Challenge organizers: Marcin Wojnarski, Andrzej Janusz, Hung Son Nguyen and Jan Bazan. Our thanks to all participants of the two tracks of the Discovery Challenge. The winners of the challenge were awarded prizes.

We wish to thank all of the authors who contributed to this volume. We are very grateful to the Chairs, advisory board members, Program Committee

members, and other reviewers not listed in the conference committee for their help in the acceptance process.

We are very grateful to the scientists who kindly agreed to give the keynote lectures: Katia Sycara, Rakesh Agrawal, Sankar K. Pal and Roman Słowiński. We also wish to express our deep appreciation to special session and industrial session organizers: Jerzy Stefanowski, Andrzej Czyżewski, Bożena Kostek, Dominik Ryżko, Henryk Rybiński and Piotr Gawrysiak.

Our thanks go to institutions that provided organizational support for RSCTC 2010 – Faculty of Mathematics, Informatics and Mechanics of the University of Warsaw, Polish Mathematical Society (PTM) and Institute of Computer Science, Faculty of Electronics and Information Technology of the Warsaw University of Technology. We also greatly appreciate the co-operation, support, and sponsorship of Springer, the International Rough Set Society and TunedIT.

The organizers wish to thank the Ministry of Science and Higher Education of the Republic of Poland for the financial support, which significantly contributed to the success of the conference.

We wish to thank several people whose hard work made the organization of RSCTC 2010 possible. We are very grateful to Stefan Jackowski and Krystyna Jaworska of PTM, as well as Marcin Kuzawiński – the author of the registration system.

Finally, we wish to express our thanks to Alfred Hofmann, Anna Kramer, Ingrid Beyer, and several anonymous technical editors of Springer for their support and co-operation during the preparation of this volume.

June 2010

Marcin Szczuka
Marzena Kryszkiewicz
Sheela Ramanna
Richard Jensen
Qinghua Hu

RSCTC 2010 Conference Organization

General Chair	Marcin Szczuka
Program Chairs	Marzena Kryszkiewicz Sheela Ramanna Richard Jensen Qinghua Hu
Special Session Chairs	Bożena Kostek, Andrzej Czyżewski Dominik Ryżko, Henryk Rybiński Jerzy Stefanowski Piotr Gawrysiak

Program Committee

Mohua Banerjee
Jan Bazan
Theresa Beaubouef
Maciej Borkowski
Cory Butz
Mihir Chakraborty
Chien-Chung Chan
Katarzyna Cichoń
Davide Ciucci
Chris Cornelis
Krzysztof Cyran
Jianhua Dai
Martine De Cock
Ivo Düntsch
Michelle Galea
Anna Gomolińska
Salvatore Greco
Jerzy Grzymała-Busse
Aboul Ella Hassanien
Jun He
Daryl Hepting
Shoji Hirano
Masahiro Inuiguchi
Ryszard Janicki
Jouni Järvinen
John A. Keane
Jacek Koronacki
A.M. Kozae
Krzysztof Krawiec
Yasuo Kudo
Yuefeng Li
Pawan Lingras
Neil Mac Parthalain
Pradipta Maji
Ernestina Menasalvas
Douqian Miao
Sushmita Mitra
Sadaaki Miyamoto
Mikhail Moshkov
Maurice Mulvenna
Michinori Nakata
Hung Son Nguyen
Sinh Hoa Nguyen Thi
Tuan Trung Nguyen
Hala Own
Sankar K. Pal
Krzysztof Pancerz
Puntip Pattaraintakorn
Alberto Guillen Perales
Georg Peters
James F. Peters
Anna Radzikowska
Zbigniew Raś
Hiroshi Sakai
Qiang Shen
Arul Siromoney

Andrzej Skowron
Roman Słowiński
Urszula Stańczyk
John Stell
Jarosław Stepaniuk
Zbigniew Suraj
Dominik Ślęzak
Li-Shiang Tsay
Shusaku Tsumoto
Aida Vitória
Alicja Wakulicz-Deja
Krzysztof Walczak
Guoyin Wang
Hui Wang
Piotr Wasilewski
Richard Weber
Szymon Wilk
Marcin Wojnarski
Marcin Wolski
Wei-Zhi Wu
JingTao Yao
Yiyu Yao
William Zhu
Wojciech Ziarko

Special Session Reviewers

Andrzej Czyżewski
Włodzisław Duch
João Gama
Ramón Garcia Gomez
Piotr Gawrysiak
Nathalie Japkowicz
Mario Koeppen
Bożena Kostek
Kristian Kroschel
Wojciech Kotłowski
Ludmila I. Kuncheva
Tomasz Martyn
Stan Matwin
Georgios V. Papanikolaou
Henryk Rybiński
Dominik Ryżko
Chiaki Sakama
Jerzy Stefanowski
Robert Susmaga
Alexey Tsymbal

External Reviewers

Stanisław Ambroszkiewicz
Piotr Andruszkiewicz
Pradeep Atrey
Krzysztof Dembczynski
Timur Fayruzov
Fernando Antônio Campos Gomide
Michał Jarociński
Rajmund Kożuszek
Qing Liu
Yang Liu
Ewa Łukasik
Sung-Kwun Oh
Hossein Pourreza
Wen Yan

Table of Contents

Keynote Talks

RSCTC 2010 Discovery Challenge

Clustering

Multimedia and Telemedicine: Soft Computing Applications

Combined Learning Methods and Mining Complex Data

Rough Sets: Logical and Mathematical Foundations

Rough Approximations: Foundations and Methodologies

Machine Learning: Methodologies and Algorithms

Multiagent Systems

Emerging Intelligent Technologies and Net-Centric Applications

Classification and Decision Support Applications

Intelligent Methods in Optimization and Control

Granularity and Granular Systems

Emergent Dynamics of Information Propagation in Large Networks

Katia Sycara

Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
katia@cs.cmu.edu

Summary

Large scale networked systems that include heterogeneous entities, e.g., humans and computational entities are becoming increasingly prevalent. Prominent applications include the Internet, large scale disaster relief and network centric warfare. In such systems, large heterogeneous coordinating entities exchange uncertain information to obtain situation awareness. Uncertain and possibly conflicting sensor data is shared across a peer-to-peer network. Not every team member will have direct access to sensors and team members will be influenced mostly by their neighbors in the network with whom they communicate directly. In this talk I will present our work on the dynamics and emergent behaviors of a large team sharing beliefs to reach conclusions about the world. Unlike past work, the nodes in the networks we study are autonomous and actively fuse information they receive. Nodes can change their beliefs as they receive additional information over time.

We find empirically that the dynamics of information propagation in such belief sharing systems are characterized by information avalanches of belief changes caused by a single additional sensor reading. The distribution of the size of these avalanches dictates the speed and accuracy with which the team reaches conclusions. A key property of the system is that it exhibits qualitatively different dynamics and system performance over different ranges of system parameters. In one particular range, the system exhibits behavior known as scale-invariant dynamics which we empirically find to correspond to dramatically more accurate conclusions being reached by team members. Due to the fact that the ranges are very sensitive to configuration details, the parameter ranges over which specific system dynamics occur are extremely difficult to predict precisely. I will present results on the emergent belief propagation dynamics in those systems, mathematical characterization of the systems' behavior and distributed algorithms for adapting the network behaviors to steer the whole system to areas of optimized performance.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, p. 1, 2010.

New Applications and Theoretical Foundations of the Dominance-based Rough Set Approach

Roman Słowiński[1,2]

[1] Institute of Computing Science, Poznań University of Technology, Poznań
[2] Systems Research Institute, Polish Academy of Sciences, 00-441 Warsaw, Poland
roman.slowinski@cs.put.poznan.pl

Summary

Dominance-based Rough Set Approach (DRSA) has been proposed as an extension of the Pawlak's concept of Rough Sets in order to deal with ordinal data [see [2,3]]. Ordinal data are typically encountered in multi-attribute decision problems where a set of objects (also called actions, acts, solutions, etc.) evaluated by a set of attributes (also called criteria, variables, features, etc.) raises one of the following questions: (i) how to assign the objects to some ordered classes (ordinal classification), (ii) how to choose the best subset of objects (optimization), or (iii) how to rank the objects from the best to the worst (ranking). The answer to everyone of these questions involves an aggregation of the multi-attribute evaluation of objects, which takes into account a law relating the evaluation and the classification, or optimization, or ranking decision. This law has to be discovered from the data by inductive learning. In case of decision problems corresponding to some physical phenomena, this law is a model of cause-effect relationships, and in case of a human decision making, this law is a decision maker's preference model. In DRSA, these models have the form of a set of "*if..., then...*" decision rules. In case of multi-attribute classification the syntax of rules is: "*if evaluation of object a is better (or worse) than given values of some attributes, then a belongs to at least (at most) given class*", and in case of multi-attribute optimization or ranking: "*if object a is preferred to object b in at least (at most) given degrees with respect to some attributes, then a is preferred to b in at least (at most) given degree*".

Since its conception, DRSA has been adapted to a large variety of decision problems [10]. Moreover, it has been adapted to handle granular (fuzzy) information [5], and incomplete information [1]. Stochastic version of DRSA has also been characterized in [9].

In this presentation, we will concentrate on two recent applications of DRSA: decision under uncertainty and time preference [6], and interactive robust multi-objective optimization [4,7]. Moreover, we will give account of topological properties of DRSA [8], using the concept of a bitopological space.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 2–3, 2010.

References

1. Dembczyński, K., Greco, S., Słowiński, R.: Rough set approach to multiple criteria classification with imprecise evaluations and assignments. European Journal of Operational Research 198, 626–636 (2009)
2. Greco, S., Matarazzo, B., Słowiński, R.: The use of rough sets and fuzzy sets in MCDM. In: Gal, T., Stewart, T., Hanne, T. (eds.) Advances in Multiple Criteria Decision Making, pp. 14.1–14.59. Kluwer, Boston (1999)
3. Greco, S., Matarazzo, B., Słowiński, R.: Rough sets theory for multicriteria decision analysis. European Journal of Operational Research 129, 1–47 (2001)
4. Greco, S., Matarazzo, B., Słowiński, R.: Dominance-based Rough Set Approach to Interactive Multiobjective Optimization. In: Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.) Multiobjective Optimization. LNCS, vol. 5252, pp. 121–155. Springer, Heidelberg (2008)
5. Greco, S., Matarazzo, B., Słowiński, R.: Granular Computing for Reasoning about Ordered Data: the Dominance-based Rough Set Approach. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Handbook of Granular Computing, ch. 15, pp. 347–373. John Wiley & Sons, Chichester (2008)
6. Greco, S., Matarazzo, B., Słowiński, R.: Dominance-based rough set approach to decision under uncertainty and time preference. Annals of Operations Research 176, 41–75 (2010)
7. Greco, S., Matarazzo, B., Słowiński, R.: DARWIN: Dominance-based rough set Approach to handling Robust Winning solutions in INteractive multiobjective optimization. In: Proc. 5th Intl Workshop on Preferences and Decisions, Trento, April 6-8, pp. 34–41 (2009)
8. Greco, S., Matarazzo, B., Słowiński, R.: Algebra and Topology for Dominance-based Rough Set Approach. In: Raś, Z.W., Tsay, L.-S. (eds.) Advances in Intelligent Information Systems. Studies in Computational Intelligence, vol. 265, pp. 43–78. Springer, Berlin (2010)
9. Kotłowski, W., Dembczyński, K., Greco, S., Słowiński, R.: Stochastic dominance-based rough set model for ordinal classification. Information Sciences 178, 4019–4037 (2008)
10. Słowiński, R., Greco, S., Matarazzo, B.: Rough Sets in Decision Making. In: Meyers, R.A. (ed.) Encyclopedia of Complexity and Systems Science, pp. 7753–7786. Springer, New York (2009)

RSCTC'2010 Discovery Challenge: Mining DNA Microarray Data for Medical Diagnosis and Treatment

Marcin Wojnarski[1,2], Andrzej Janusz[2], Hung Son Nguyen[2], Jan Bazan[3], ChuanJiang Luo[4], Ze Chen[4], Feng Hu[4], Guoyin Wang[4], Lihe Guan[4], Huan Luo[4], Juan Gao[4], Yuanxia Shen[4], Vladimir Nikulin[5], Tian-Hsiang Huang[5,6], Geoffrey J. McLachlan[5], Matko Bošnjak[7], and Dragan Gamberger[7]

[1] TunedIT Solutions,
Zwirki i Wigury 93/3049, 02-089 Warszawa, Poland
marcin.wojnarski@tunedit.org
[2] Faculty of Mathematics, Informatics and Mechanics, University of Warsaw
Banacha 2, 02-097 Warszawa, Poland
andrzejanusz@gmail.com, son@mimuw.edu.pl
[3] Institute of Mathematics, University of Rzeszow
Rejtana 16A, 35-959 Rzeszow, Poland
bazan@univ.rzeszow.pl
[4] Institute of Computer Science and Technology,
Chongqing University of Posts and Telecommunications,
Chongqing 400065, P.R.China
{hufeng,wanggy}@cqupt.edu.cn
[5] Department of Mathematics, University of Queensland
v.nikulin@uq.edu.au, gjm@maths.uq.edu.au
[6] Institute of Information Management, National Cheng Kung University, Taiwan
huangtx@gmail.com
[7] Laboratory for Information Systems, Rudjer Boskovic Institute,
Bijenicka 54, 10000 Zagreb, Croatia,
matko.bosnjak@irb.hr, dragan.gamberger@irb.hr

Abstract. RSCTC'2010 Discovery Challenge was a special event of Rough Sets and Current Trends in Computing conference. The challenge was organized in the form of an interactive on-line competition, at TunedIT.org platform, in days between Dec 1, 2009 and Feb 28, 2010. The task was related to feature selection in analysis of DNA microarray data and classification of samples for the purpose of medical diagnosis or treatment. Prizes were awarded to the best solutions. This paper describes organization of the competition and the winning solutions.

1 Introduction

In recent years, a lot of attention of researchers from many fields has been put into investigation of DNA microarray data. This growing interest is largely motivated by numerous practical applications of knowledge acquired from such data

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 4–19, 2010.

in medical diagnostics, treatment planning, drugs development and many more. When analyzing microarray data, researchers have to face the few-objects-many-attributes problem, as the usual ratio between the number of examined genes and the number of available samples exceeds 100. Many standard classification algorithms have difficulties in handling such highly dimensional data and due to low number of training samples tend to overfit. Moreover, usually only a small subset of examined genes is relevant in the context of a given task. For these reasons, feature extraction methods – in particular the ones based on rough-set theory and reducts - are an inevitable part of any successful microarray data classification algorithm. With RSCTC'2010 Discovery Challenge, the organizers wanted to stimulate investigation in these important fields of research.

The challenge was organized in the form of an interactive on-line competition, at TunedIT (http://tunedit.org) platform, in days between December 1, 2009 and February 28, 2010. The task was to design a machine-learning algorithm that would classify patients for the purpose of medical diagnosis and treatment. Patients were characterized by gene transcription data from DNA microarrays. The data contained between 20,000 and 65,000 features, depending on the type of microarrays used in a given experiment.

Organizing Committee of the challenge had four members: Marcin Wojnarski, Andrzej Janusz, Hung Son Nguyen and Jan Bazan.

2 Organization of the Challenge

Challenge comprised two independent tracks, namely *Basic* and *Advanced*, differing in the form of solutions. In Basic Track, the participant had to submit a text file with predicted decisions for test samples, which was later compared with the ground truth decisions – a typical setup used in other data mining challenges. In Advanced Track, the participant had to submit Java source code of a classification algorithm. The code was compiled on server, the classifier was trained on a subset of data and evaluated on another subset.

On one hand, Advanced Track was more challenging for participants than Basic Track because there were restrictions on the way how the algorithm was implemented – it must have been written in Java, according to API defined by one of three data mining environments: Weka, Debellor or Rseslib. On the other hand, every algorithm have been trained and tested a number of times on the same datasets, using different splits into train/test parts which allowed much more accurate evaluation of solutions. This is particularly important for the problems like DNA microarray data analysis, where datasets are small and evaluation with single train/test split is not fully objective.

Another advantage of Advanced track was the possibility to evaluate not only the accuracy of decisions made by algorithms but also their time and memory complexity. Limits were set for execution of the evaluation procedure, so if the algorithm was too slow or required too much memory, the computation was interrupted with an error. Moreover, after the end of the competition it was possible to disclose the source codes of the solutions at TunedIT server, exactly in the same form that underwent evaluation, to be used by all researchers as

a benchmark or starting point for new research. Other details of the challenge setup can be found at http://tunedit.org/challenge/RSCTC-2010-A.

3 Datasets

Twelve microarray datasets from a wide range of medical domains were used in the competition. All of them were acquired from a public microarray repository ArrayExpress[1] (to find out more about the repository see [1]). All microarray experiment results in this repository are stored in MIAME standard and their detailed description as well as previous usage is available on-line. The datasets chosen for the basic track of the challenge are related to diverse research problems: recognition of acute lymphoblastic leukemia genetic subtypes (experiment accession number E-GEOD-13425), diagnostic of human gliomas (accession number E-GEOD-4290), transcription profiling of human healthy and diseased gingival tissues (accession number E-GEOD-10334), transcription profiling of human heart samples with different failure reasons (accession number E-GEOD-5406), recognition of genomic alterations that underlie brain cancer (accession number E-GEOD-9635) and profiling of human systemic inflammatory response syndrome (SIRS), sepsis, and septic shock spectrum (accession number E-GEOD-13904). For the advanced track, selected datasets concerned prediction of response to anthracycline/taxane chemotherapy (accession number E-GEOD-6861), diagnostic of human Burkitts lymphomas (accession number E-GEOD-4475), investigation of a role of chronic hepatitis C virus in the pathogenesis of HCV-associated hepatocellular carcinoma (accession number E-GEOD-14323), profiling of several murine genotypes on subjects stimulated with purified Toll-like receptor agonists (accession number E-TABM-310), recognition of ovarian tumour genetic subtypes (accession number E-GEOD-9891) and recognition of multiple human cancer types (accession number E-MTAB-37).

For the purpose of the competition only the processed versions of the datasets were utilized and no additional microarray normalization was performed. Data preparation was done in *R System*[2] (see [2]). During preprocessing, decision classes of samples were assigned based on the available "*Sample and Data Relationship*" files. Those decision classes, which were supported only by few samples, were removed from data or they were merged with similar classes (e.g. some subtypes of a specific medical condition could have been merged together to form a decision class which is better-represented in data). Any additional information about samples (such as gender, age, smoking habits) was disregarded.

Several precautions were taken to avoid identification of the datasets by contestants. For each decision set, sample and gene identifiers were removed. After that, the samples as well as genes were randomly shuffled and a few samples were taken out with some probability. Finally, gene expression values were divided by the standard deviation of all expression levels in the corresponding sets and the datasets, for which at least one gene fulfilled a criterion that its range was more

[1] www.ebi.ac.uk/arrayexpress
[2] http://www.R-project.org

than 100 times greater than the distance between its first and the third quantile, were logarithmically scaled using the formula:

$$x' = sign(x) * \log(|x| + 1)$$

Brief characteristics of the prepared datasets are given in Table 1.

Table 1. A brief summary of the microarray datasets used in the challenge

Accession number:	no. samples	no. genes	no. classes
E-GEOD-13425	190	22276	5
E-GEOD-4290	180	54612	4
E-GEOD-10334	247	54674	2
E-GEOD-5406	210	22282	3
E-GEOD-9635	186	59003	5
E-GEOD-13904	227	54674	5
E-GEOD-6861	160	61358	2
E-GEOD-4475	221	22282	3
E-GEOD-14323	124	22276	4
E-TABM-310	216	45100	7
E-GEOD-9891	284	54620	3
E-MTAB-37	773	54674	10

In order to provide reasonable baseline scores for participants, three classic features selection methods were combined with 1-Nearest-Neighbor algorithm and used to perform a classification of the test samples from Basic track. The first method was based on the *relief* algorithm (for more details see [3]). This multivariate filter approach measures usefulness of attributes in k-NN classification and can efficiently identify irrelevant features. The gene selection threshold was estimated on the training data using the random probes technique with a probability of selecting an individually irrelevant gene set to 0.05. The irrelevant genes were removed form data and the elimination process was repeated until all the genes that left in a dataset were marked as relevant. The second and the third method were utilizing univariate statistical tests (Pearson's correlation test and the t-test) to filter out unimportant genes (see [4]). For each dataset, the number of selected genes was also estimated using random probes but this time, a desired probability of choosing an irrelevant gene was tuned by *leave-one-out cross-validation* on training examples. The results achieved by the baseline methods were published on the leaderboard during the competition and are summarized in Table 3.

4 Evaluation of Solutions

Solutions were evaluated using a total of 12 datasets from a variety of microarray experiments, each one related to a different medical problem, with different number of attributes and decision classes. Thus, participants had to design algorithms which can be successfully applied to *many* problems of DNA microarrays

analysis, not only to one. Evaluation was performed automatically on TunedIT servers using TunedTester application. Every solution underwent two distinct evaluations: *preliminary* and *final*. The results of preliminary evaluation were published on the leaderboard (after they were calculated), while the final results were disclosed after completion of the challenge. Only the final results were taken into account when deciding the winners.

Each of the 12 datasets was assigned to one of the two tracks, thus solutions on every track were evaluated using six datasets, different for each track. The data used on Basic Track were divided into separate training and test sets and were published on the challenge web page. The decisions for samples from the test sets were kept secret and the task was to submit their predictions. On the server, the solutions were compared with expected decisions and their quality was calculated. To avoid bias in the final results caused by overfitting, half of the predictions were used for calculation of the preliminary results, and another half for the final results.

In Advanced Track, all datasets were kept secret, so participants could not access them. Instead, participants could have used public data from Basic Track to test their solutions before submission to the challenge. After submission, the algorithms were evaluated on each dataset with Train+Test procedure applied a number of times. Each Train+Test trial consisted of randomly splitting the data into two equal disjoint parts, training and test subsets, training the algorithm on the first part and testing it on the second. Quality measurements from all trials on a given dataset were averaged. Randomization of data splits was the same for every submitted solution so every algorithm was evaluated on the same splits. The number of repetitions of Train+Test procedure on each dataset was set to 5 for the preliminary evaluation and 20 for the final.

The datasets that were employed on Basic Track in the preliminary and the final evaluation, included: E-GEOD-13425, E-GEOD-4290, E-GEOD-10334, E-GEOD-5406, E-GEOD-9635 and E-GEOD-13904.

The preliminary evaluation on Advanced Track employed 5 datasets: E-GEOD-4475, E-GEOD-14323, E-TABM-310, E-GEOD-9891 and a half of E-MTAB-37 (part A). The final evaluation on Advanced Track employed a total of 6 datasets: 4 datasets from preliminary evaluation, another half of E-MTAB-37 and a new dataset, not used in preliminary evaluation: E-GEOD-6861, E-GEOD-4475, E-GEOD-14323, E-TABM-310, E-GEOD-9891 and E-MTAB-37 (part B).

Datasets from medical domains usually have skewed class distributions, with one dominant class represented by majority of samples and a few minority classes represented by small number of objects. This was also the case in this challenge. Typically, minority classes are more important than the dominant one and this fact should be reflected by the quality measure used to assess the performance of algorithms. For this reason, solutions were evaluated using *balanced accuracy* quality measure. This is a modification of standard classification accuracy that is insensitive to imbalanced frequencies of decision classes. It is calculated by computing standard classification accuracies (acc_k) for every decision class and

then averaging the result over all classes ($k = 1, 2, \ldots, K$). In this way, every class has the same contribution to the final result, no matter how frequent it is:

$$S_k = \#\{i : class(sample_i) = k\}$$

$$acc_k = \#\{i : prediction(sample_i) = class(sample_i) = k\}/S_k$$

$$BalancedAcc = (acc_1 + acc_2 + \ldots + acc_K)/K$$

In the case of 2-class problems with no adjustable decision threshold, balanced accuracy is equivalent to *Area Under the ROC Curve* (AUC). Thus, it may be viewed as a generalization of AUC to multi-class problems.

In the competition, the balanced accuracy of algorithms was calculated separately for each dataset used on a given track and then the results were averaged.

In the evaluation of the Advanced Track, not only accuracy, but also time-and-memory-efficiency of algorithms were considered. A time limit was set for the whole evaluation: 5 hours in the preliminary tests and 20 hours in the final tests. Therefore, a single Train+Test trial of the algorithm lasted, on average, no longer than 60 minutes. The memory limit was set to 1,500 MB, both in preliminary and final evaluation. Up to 450 MB was used by evaluation procedure to load a dataset into memory, so 1 GB was left for the algorithms. Tests were performed on a station with 1.9 GHz dual-core CPU, 32-bit Linux and 2 GB memory, running Sun Java HotSpot Server 14.2 as a JVM.

5 Results

There were 226 participants registered to the challenge. The number of *active participants* – the ones who submitted at least one solution – was 93 for Basic Track and 29 for Advanced Track. If a participant made more than one submission, the last solution was considered as the final one. The first 3 winners on each track, together with baseline results, are presented in Tables 2 and 3.

After the challenge, we calculated the results that would be obtained by an ensemble made of a number of top solutions from Basic Track, through a simple voting. These results are presented in Table 3. Combining 7 top solutions gave significantly higher accuracy than that of the best individual algorithm. We also constructed an ensemble of 54 solutions whose individual performances were better than the best baseline and an ensemble of 92 solutions which achieved higher rank than a "majority vote" classifier.

TunedIT awarded the first winners on both tracks with money prizes: 2,000 USD on Advanced track and 1,000 USD on Basic track. Additionally, conference registration fees for two participants, one from each track, were covered.

All employed datasets and the source code of evaluation procedures are available at TunedIT Repository[3] so new algorithms can be tested against challenge data, using the same experimental setup. In this way, the challenge contributed to creation of benchmark datasets that can be reused in the future by the whole scientific community.

In the following sections, the winners briefly describe their approaches.

[3] Links can be found at http://tunedit.org/challenge/RSCTC-2010-A

Table 2. Final results of the Advanced Track

Rank	Participant or Team	Username	Final Result
1	ChuanJiang Luo, Ze Chen, Feng Hu, Guoyin Wang, Lihe Guan, Inst of Computer Science and Technology, Chongqing Univ of Posts & Telecomm, China	RoughBoy	0.75661
2	Huan Luo, Juan Gao, Feng Hu, Guoyin Wang, Yuanxia Shen, Inst of Computer Science and Technology, Chongqing Univ of Posts & Telecomm, China	ChenZe	0.75180
3	wulala	wulala	0.75168

Table 3. Final results of the Basic Track

Rank	Participant or Team	Username	Final Result
1	Vladimir Nikulin, Dept. of Mathematics, University of Queensland, Australia	UniQ	0.73870
2	Matko Bošnjak, Dragan Gamberger, Rudjer Boskovic Institute, Croatia	RandomGuy	0.73485
3	Ryoji Yanashima, Keio University, Japan	yanashi	0.73108
–	Baseline_relief-1NN	–	0.65122
–	Baseline_corTest-1NN	–	0.64087
–	Baseline_tTest-1NN	–	0.63464
–	Baseline: majority classifier	–	0.28056
–	Ensemble of top 7 final solutions	–	0.7469
–	Ensemble of top 54 final solutions	–	0.7113
–	Ensemble of top 92 final solutions	–	0.6870

6 The Best Solution from Basic Track: Feature Selection with Multi-class Wilcoxon Criterion Applied to Classification of High-Dimensional Microarray Data

6.1 Initial Model and Evaluation Scheme

Initially, we decided to conduct some experiments with the Nearest Shrunken Centroids (NSC) method as it is described in [5]. The NSC method may be viewed as a sequence of two steps FS+Model, (i) feature selection (FS) and (ii) classification. The FS step depends on a very important parameter Δ, which should be selected specially for the particular dataset. As far as we were dealing with the case of a small sample size, LOO (leave-one-out) was the most appropriate evaluation scheme:

$$FS + LOO(Model). \tag{1}$$

With this scheme, we can consider several parameter settings, and the system will select the most appropriate setting depending on the LOO evaluations.

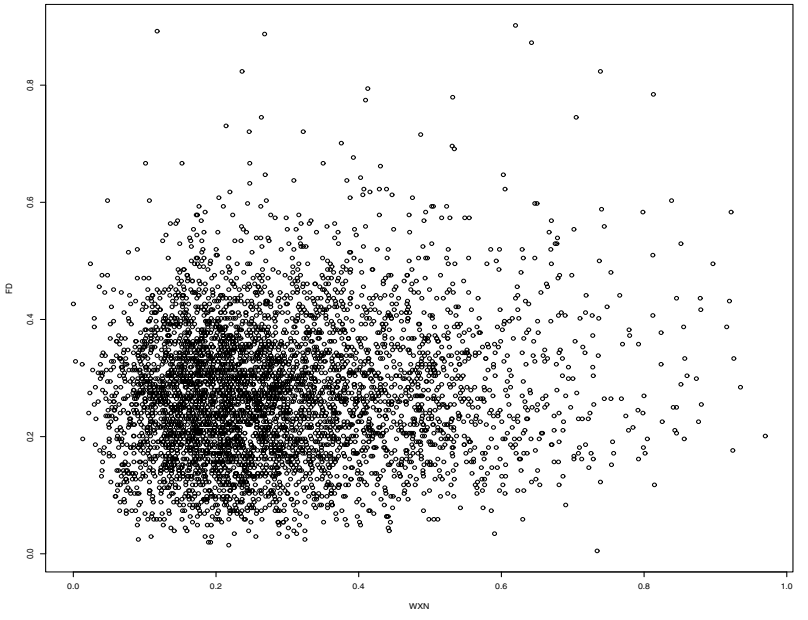


Fig. 1. Relation between WXN and FD scoring functions, where the most simplest Set3 was used as a benchmark

6.2 Wilcoxon and Fisher Discriminant Criterions for Feature Selection

Let us denote by $\mathcal{N}_a$ a set of all samples/tissues within the class a. The following criterion (named Wilcoxon) was used for the selection of the most relevant features

$$WXN(feature) = \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \max(q_{ab}(feature), q_{ba}(feature)), \tag{2}$$

where

$$q_{ab}(feature) = \sum_{i \in \mathcal{N}_a} \sum_{j \in \mathcal{N}_b} I(feature_i - feature_j \leq 0),$$

where I is an indicator function.

In addition, we conducted some experiments with Fisher Discriminant criterion:

$$FD(feature) = \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \frac{|\mu_a(feature) - \mu_b(feature)|}{s_a(feature) + s_b(feature)}, \tag{3}$$

where $\mu_a(feature)$ and $s_a(feature)$ are mean and standard deviation of *feature* within the class a. Note that both criterions WXN and FD were normalised to the range $[0, \ldots, 1]$.

Figure 1 illustrates significant structural difference between the WXN and FD criterions. We used an ensemble constructor as it is described in [6] to create an ensemble (named ENS) of WXN and FD criterions, and gained some improvement in application to Set1.

FS was conducted according to the rule:

$$ENS(feature) \geq \Delta > 0.$$

6.3 Classification Models

Very simple and fast classification (FS+FD). The following classification rule may be viewed as a simplification of the NSC:

$$decision = \underset{a}{\operatorname{argmin}} \sum_{feature=1}^{p} \frac{|feature(new.sample) - \mu_a(feature)|}{s_a(feature)}, \quad (4)$$

and may be used immediately after FS step. Note that LOO evaluation and validation results, which we observed with (4), were better compared to the NSC model. It is easy to see a close relation between (3) and (4). Consequently, we use abbreviation FS+FD for the model (4).

Table 4. Statistical characteristics of the solution, which was produced using ENS+FD method, where p_S is the number of selected features; LOO values are given in terms of the balanced accuracy; n1, n2, n3, n4, n5 are the numbers of decisions per class

N	Δ	p_S	LOO	n1	n2	n3	n4	n5
1	0.876	171	0.9088	86	37	-	-	-
2	0.801	44	0.8273	43	54	7	-	-
3	0.55	1542	0.9765	21	7	24	26	16
4	0.663	1031	0.5433	10	32	39	22	9
5	0.845	123	0.7432	17	22	29	21	-
6	0.731	679	0.7127	19	6	14	41	12

Validation experiments. During validation trials we conducted experiments with several classification models, including our own (translated from C to JAVA), CLOP (Matlab)[4] and, also, models from the Weka package[5].

In the case of the basic track the best validation result **0.76** was produced with ENS+FD (Set 1), WXN+FD (Set 5), WXN+MLP(Weka) (Sets 2-4, 6), where MLP stands for multilayer perceptron.

The top validation result **0.8089** for the advanced track was produced with WXN+MLP(Weka), where we used fixed $\Delta = 0.67$ for FS and

$$ops = \{\text{“}-L\text{”}, \text{“0.25”}, \text{“}-N\text{”}, \text{“1200”}, \text{“}-H\text{”}, \text{“11”}, \text{“}-M\text{”}, \text{“0.1”}\}$$

- settings for MLP(Weka).

[4] http://clopinet.com/CLOP/

[5] http://weka.sourceforge.net/doc/overview-tree.html

Also, we can recommend to consider SVM(Weka) with the following setting

$$ops = \{\text{“}-C\text{”}, \text{“1”}, \text{“}-R\text{”}, \text{“1”}, \text{“}-G\text{”}, \text{“9”}\}.$$

Using above model we observed validation result 0.7947 in the advanced track.

In difference to simple and fast model described in Section 6.3, models SVM(Weka) and MLP(Weka) are rather slow. As a consequence, and in order to avoid "time-out" outcome, we did not use the most natural scheme (1) for our final submission (advanced track). Our final JAR-file was prepared with fixed Δ - that means, the power of FS was about the same for all test data sets. However, based on our experiments with LOO-evaluations, we have noticed that the performance of the model is a very sensitive to the selection of the parameter Δ, see Table 4.

7 The Second Best Solution from Basic Track: Random Forest Approach for Distributed and Unbalanced Prediction Tasks

7.1 Introduction

Typical properties of microarray datasets are a large number of attributes, small number of examples and usually unbalanced class distributions. Most importantly, relevant information in these datasets is distributed across many attributes. As such, these datasets are a challenging prediction task.

Good prediction results for such datasets can be expected from systems able to appropriately reduce the attribute space to some reasonable size but retain the distributed information. Also, a classifier constructed from the reduced attribute set should still be able to integrate a relatively large number of attributes into a model. There is a significant danger of overfitting because the model must be complex and the number of available training examples is small. In such situation construction of many diverse classifiers and implementation of an appropriate voting scheme seems as the only possible solution.

The Random Forest (RF) [7] approach for supervised inductive learning enables a relatively simple and straightforward framework for building a set of diverse classifiers. Some of its advantages are the ability to cope with extremely large number of attributes even when there is a small number of instances, and an option to balance datasets through user defined weights.

In the next Section we present some basic concepts of the RF approach, describe its current parallel implementation prepared at the Rudjer Boskovic Institute, and demonstrate RF attribute importance feature. In Sections 7.3 and 7.4 we describe details of our solution which concentrated mainly on the task of reducing the size of the original attribute space, search for the optimal weights of classes thus balancing class distribution, and an approach to refine RF results by outlier detection and correction.

7.2 Random Forest

Random Forest is a general purpose classification and regression meta-learning algorithm which works by combining bagging [8] and random subspace method [9] approaches in constructing an ensemble of random decision trees.

RF is computationally efficient as it is able to learn a forest faster than bagging or boosting, it is capable of handling large datasets with thousands of categorical and continuous attributes without deletion which is very suitable for microarray analysis, and is also capable of balancing out class-wise error rates through a user-defined set of weights used in the process of tree growing. Besides this, it is capable of producing many useful data for result interpretation such as attribute importance, a feature we used for our attribute selection process.

Attribute importance is calculated by comparing the misclassification rate of the original vs. per-attribute randomly permuted data in the process of error estimation for the single tree. By subtracting the number of correct votes for the attribute-permuted data from the number of correct votes of the original data and averaging them over all trees in the forest, raw importance and later, significance levels for the attributes are obtained. Attribute importance is especially helpful when using data with a large number of attributes like microarrays.

In this work we used a parallel implementation of the RF algorithm developed by G. Topić and T. Šmuc at the Rudjer Boskovic Institute. This implementation is written in Fortran 90 and the parallelization of the algorithm has been accomplished using MPI (Message Passing Interface). PARF is licensed under GNU GPL 2.0 license. More information about the implementation, usage, help and source code can be found on PARF's homepage: `http://www.parf.irb.hr/`.

7.3 Finding Optimal Random Forest Parameters

The main problems of RF approach practical application for the RSCTC'2010 datasets has been confronted with are a) selection of the appropriate subset of attributes and b) selection of appropriate weights for classes with small number of examples in the training set. The problems have been solved with a series of experiments performed with different levels of attribute reduction and different weights for rare example classes. The optimal combination has been identified by minimal out-of-bag error estimation [7] which has been remodeled to fit the evaluation criteria of the balanced accuracy defined for the Challenge. All of the experiments were executed with a number of trees in the forest equal to 1000 to ensure stable prediction quality.

In order to implement a relatively systematic search through the space of all possibly interesting combinations, a double nested loop has been implemented. In the outer loop the threshold for attribute significance was varied from $0.1 - 0.01$ in steps of 0.01. With this parameter we have varied the size of the attribute set entering the second RF learning phase. The actual numbers of the selected attributes varied between datasets from as low as 7 to 603.

In the inner loop we had a parameter for class weight optimization. The weights for all classes with high number of instances were set to 1 while for

rare classes they have been preset inverse proportionally to the size of the class. The latter have been multiplied by the parameter for weight class optimization which varied from $1-4$ in increments of 0.2.

In the described setting a total of 160 experiments have been performed for each dataset. Typically optimal attribute significance was in the range $0.06-0.09$ with class weight parameter typically in the range $1.5-3.1$.

7.4 Outlier Detection from Integrated Training and Test Sets

The methodology described in the previous section enabled us to select optimal RF parameters for each dataset. By using these parameters we have constructed one final model for each dataset which we used to classify test set examples. Afterwards, we additionally applied a methodology for outlier detection in order to improve the solution.

For this task we have integrated each dataset with classified examples from its corresponding test set. By their construction we have been able to test if there are strong outliers in these large datasets and in cases when the outliers are from test sets, try to correct them. For this task we have applied the saturation based filtering methodology for explicit outlier detection described in [10].

The saturation based outlier detection methodology tries to estimate minimal complexity of the hypothesis that is able to correctly classify all examples in the available dataset. After that it tries to identify if there exist examples by whose elimination this complexity could be significantly reduced. If one or more such examples can be found, they are declared as potential outliers. The methodology is appropriate for domains in which useful classification information is concentrated in a small set of very important attributes. Having distributed information in microarray datasets in this Challenge we had to use it with special care. Additionally, the currently available version of the methodology can handle only two-class domains. Because of these problems we have used it in a semi-automatic mode, carefully evaluating each detected outlier and a potential change of its originally determined class.

For each enlarged dataset with C classes we have constructed C different concept learning (two-class) problems so that each class is once a positive class and examples from all other classes are treated as negative class examples. In this way one original multiclass example has once been a positive example and $C-1$ times a negative example. Outlier detection process has been repeated for all concept learning tasks independently. Finally, we have searched for examples coming from the test set that have been detected as potential outliers when they have been among positive examples and exactly once when they have been in some negative class. Only in such cases we accepted to change the original classification of the example. The new classification of the example corresponded to the positive class of the concept learning task when the example has been detected as a negative outlier. This methodology enabled correction of the classification for up to 3 examples per dataset.

8 The Best Solution from Advanced Track: A Feature Selection Algorithm Based on Cut Point Importance and Dynamic Clustering[6]

In RSCTC'2010 Discovery Challenge, the DNA data arrays [11] with large number of features (attributes) and small number of records (objects) were provided. Suppose $|U|$ be the number of objects, and $|C|$ be the number of features. According to the provided data, it is obvious that $|C| \gg |U|$. Therefore, it is urgent to select smaller subset of features from the DNA data array. In this section, an efficient solution for feature selection method, based on importance of cut points and dynamic clustering, is introduced. It is combined with SVM.

Firstly, according to [12], the importance of cut points can be computed. After that, the feature selection algorithm based on importance of cut points and dynamic clustering will be presented as follows.

Algorithm 1. Feature Selection Algorithm Based on Cut Point Importance and Dynamic Clustering:

Input: Decision table $S =< U, A = C \cup D, V, f >$ and feature number K.
Output: Selected feature set $SelectFeature$.
Step1: (Computing the importance of cut points on all features).
FOR i = 1 TO $|C|$ DO
Computing the importance value of cut points on feature c_i, according to [12];
Normalization the importance value of cut points on feature c_i.
END FOR
Step2: (Dynamic clustering)(**Due the limitation of page size, we can not present the complex algorithm in detail**)
FOR each $c_i (1 \leq i \leq |C|)$ DO
Step2.1: Sorting cut points.
Step2.2: Connecting the importance value of all cut points. It can be found that there will be only a summit on the curve of the importance value of cut points. According to the summit, dividing the cut points into two parts: $Left$ and $Right$.
Step2.3: Dynamic clustering the importance of cut points. Then, the importance of cut points in $Left$ or $Right$ can be dynamic clustered respectively.
Step2.4: Suppose the clustered classifications on feature c_i be k_i.
END FOR

[6] This work is supported by the National Natural Science Foundation of China (NSFC) under grant No.60573068 and No.60773113, Natural Science Foundation of Chongqing under grant No.2008BA2017 and No.2008BA2041, and Science & Technology Research Program of Chongqing Education Commission under grant No.KJ090512.

Step3: Sort $k_1, k_2, ..., k_{|C|}$ by increasing order. Suppose the order result be $c_{r_1}, c_{r_2}, ..., c_{r_{|c|}}$;
$SelectFeature = \{c_{r_1}, c_{r_2}, ..., c_{r_k}\}$.
Step4: RETURN $SelectFeature$.

Secondly, a good library for Support Vector Machines is adopted. The configuration of LIBSVM [13] is introduced (see Table 5). In Table 5, the changed parameters are showed. Besides, the rest parameters are default by LIBSVM.

During experiments, parameter K was set to 5000. Performance of the presented "Feature Selection + SVM" method were tested on the DNA datasets of RSCTC'2010 Discovery Challenge. The final experimental results for Advanced Track was 0.75661.

Table 5. The parameter configuration of LIBSVM

Configuration Parameter	Description	Configuration Value
svm type	set type of SVM	C-SVC
kernel type	set type of kernel function	LINEAR
gamma	set gamma in kernel function	1/num_features
cost	set the parameter C of C-SVC, epsilon-SVR, and nu-SVR	100

9 The Second Best Solution from Advanced Track: A Feature Selection Algorithm Based on Attribute Relevance[7]

When analyzing microarray data [11], we have to face the few-objects-many-attributes problem. However, there exists dependence between condition attributes and decision attribute, and only a small subset of attributes is relevant in the context of a given task. In this section, an effective feature extraction method, the feature selection algorithm based on attribute relevance, is introduced. Furthermore, combining with the proposed algorithm, the LIBSVM [13] is adopted to solve the given task in RSCTC'2010 Discovery Challenge' Advanced Track. According to [14], the ratio of condition attribute's between-groups to within-groups sum of squares can represent the relevance between condition attribute and decision attribute. We modified the ratio expression and proposed a new expression to represent attribute relevance.

[7] This work is supported by the National Natural Science Foundation of China (NSFC) under grant No.60573068 and No.60773113, Natural Science Foundation of Chongqing under grant No.2008BA2017 and No.2008BA2041, and Science & Technology Research Program of Chongqing Education Commission under grant No.KJ090512.

Firstly,feature selection algorithm based on attribute relevance can be described as following in detail.

Algorithm 2. Feature Selection Algorithm Based on Attribute Relevance:

Input: Decision table $S =< U, A = C \cup D, V, f >$ and feature number K.

Output: Selected feature set $SelectFeature$.

Step1: (Sorting objects,according to the value of decision attribute d.)

Step2: (Compute the relevance of condition attributes and decision attribute)

FOR each $c_i (1 \leq i \leq |C|)$ DO

Step2.1: Compute the standard deviation SD_i on attribute d. where

$$SD_i = \sqrt{\frac{\sum_{j}^{|U|} (x_{ji} - \overline{x}_{.i})^2}{|U|-1}}$$

Step2.2: Suppose $|U/\{d\}|$ object sets $U^1, U^2, ..., U^{|U/\{d\}|}$. Compute the standard deviation in each object set.

FOR each object set $U^k (1 \leq k \leq |U/\{d\}|)$ DO

$$SD_i^k = \sqrt{\frac{\sum_{j}^{|U^k|} (x_{ji} - \overline{x}_{ki})^2 * I(y_j = k)}{|U^k|-1}}, \text{ were } \overline{x}_{ki} = \sqrt{\frac{\sum_{j}^{|U^k|} x_{ji} * I(y_j = k)}{|U^k|}}$$

and $I(y_j = k) = \begin{cases} 1, y_j = k. \\ 0, y_j \neq k. \end{cases}$

Step2.3: Compute the relevance of condition attributes and decision attribute: $R_i = \frac{\sum_{k}^{|U/\{d\}|} SD_i^k}{SD_i}$

END FOR

Step3: Sort $R_1, R_2, \cdots, R_{|C|}$ by increasing order. Suppose the order result be $c_{r_1}, c_{r_2}, ..., c_{r_{|c|}}$;

$SelectFeature = \{c_{r_1}, c_{r_2}, ..., c_{r_k}\}$.

Step4: RETURN $SelectFeature$.

Secondly, a good library for Support Vector Machines is adopted. The configuration of LIBSVM [13] is introduced (see Table 5). In Table 5, the changed parameters are showed. Besides, the rest parameters are default by LIBSVM.

During experiments, parameter K was set to 5000. Performance of the presented "Feature Selection + SVM" method were tested on the DNA datasets of RSCTC'2010 Discovery Challenge. The final experimental results for Advanced Track was 0.75180.

Acknowledgements

The research has been supported by grants N N516 077837 and N N516 368334 from Ministry of Science and Higher Education of the Republic of Poland.

References

1. Parkinson, H.E., et al.: Arrayexpress update - from an archive of functional genomics experiments to the atlas of gene expression. Nucleic Acids Research 37(Database issue), 868–872 (2009)
2. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2008)
3. Kira, K., Rendell, L.A.: A practical approach to feature selection. In: ML92: Proceedings of the ninth international workshop on Machine learning, pp. 249–256. Morgan Kaufmann Publishers Inc., San Francisco (1992)
4. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182 (2003)
5. Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G.: Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proceedings of the National Academy of Sciences USA 99(10), 6567–6572 (2002)
6. Nikulin, V., McLachlan, G.J.: Classification of imbalanced marketing data with balanced random sets. In: JMLR: Workshop and Conference Proceedings, vol. 7, pp. 89–100 (2009)
7. Breiman, L.: Random forests. Machine Learning, 5–32 (2001)
8. Breiman, L.: Bagging predictors. Machine Learning, 123–140 (1996)
9. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Trans. Pattern Anal. Mach. Intell. 20(8), 832–844 (1998)
10. Gamberger, D., Lavrac, N.: Conditions for occam's razor applicability and noise elimination. In: van Someren, M., Widmer, G. (eds.) ECML 1997. LNCS, vol. 1224, pp. 108–123. Springer, Heidelberg (1997)
11. Wikipedia: Dna microarray – wikipedia, the free encyclopedia (2010), http://en.wikipedia.org/w/index.php?title=DNA_microarray
12. Nguyen, H.: 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)
13. Chang, C., Lin, C.: Libsvm – a library for support vector machines. [EB/OL], http://www.csie.ntu.edu.tw/~cjlin/libsvm (2008-11-17/2010-01-3)
14. Dudoit, S., Fridlyand, J., Speed, T.: Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association 97, 77–87 (2002)

TunedIT.org: System for Automated Evaluation of Algorithms in Repeatable Experiments

Marcin Wojnarski[1,2], Sebastian Stawicki[2], and Piotr Wojnarowski[1,2]

[1] TunedIT Solutions
Zwirki i Wigury 93 lok. 3049, 02-089 Warszawa, Poland
[2] Faculty of Mathematics, Informatics and Mechanics, University of Warsaw
Banacha 2, 02-097 Warszawa, Poland

Abstract. In this paper we present TunedIT system which facilitates evaluation and comparison of machine-learning algorithms. TunedIT is composed of three complementary and interconnected components: TunedTester, Repository and Knowledge Base.

TunedTester is a stand-alone Java application that runs automated tests (experiments) of algorithms. Repository is a database of algorithms, datasets and evaluation procedures used by TunedTester for setting up a test. Knowledge Base is a database of test results. Repository and Knowledge Base are accessible through TunedIT website. TunedIT is open and free for use by any researcher. Every registered user can upload new resources to Repository, run experiments with TunedTester, send results to Knowledge Base and browse all collected results, generated either by himself or by others.

As a special functionality, built upon the framework of automated tests, TunedIT provides a platform for organization of on-line interactive competitions for machine-learning problems. This functionality may be used, for instance, by teachers to launch contests for their students instead of traditional assignment tasks; or by organizers of machine-learning and data-mining conferences to launch competitions for the scientific community, in association with the conference.

1 Introduction

Almost every paper published in the field of machine learning contains experimental section, which presents empirical analysis, evaluation and comparison of described algorithms. We investigated 81 regular research papers published in Volume 9/2008 of Journal of Machine Learning Research, excluding articles assigned to special topics or to the Machine Learning Open Source Software track, as well as responses to other papers. We observed that as much as 75 (93%) of the papers contained experimental section. Experimental results constitute ultimate proof of strengths of described methods, so even a slight improvement in the methodology of conducting experiments would be beneficial to the whole community of machine learning researchers and facilitate design of even better algorithms.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 20–29, 2010.

Currently used experimental methodology has serious weaknesses. The biggest one is that experiments performed by the author of a new algorithm and described in a research article cannot be reproduced by other researchers. On paper, the author should provide full details of the algorithm and experimental procedure, sufficient to repeat the experiment. In practice:

- Providing all the details would make the article unreadable and substantially longer, so the description is rarely complete. For example, if experiments involve decision models with adaptively optimized parameters, like weights of neural networks, it is rarely described in detail how the parameters are initialized at the beginning of the training process.
- The author himself may be unaware of some important details, for example implementation bugs.
- Reimplementation of the algorithm by another researcher could take weeks of work and thus is not feasible, even if theoretically possible.
- Even if the author made implementation of the algorithm publicly available, significant effort must be put into learning how to use the implementation.
- Recreation of experimental setup may be difficult, even when all necessary elements of the experiment – data sets, implementations of algorithms etc. – are available.
- There is high risk of human mistakes. They are very hard to detect, because usually the outcomes of experiments are just numbers, only slightly different between each other. Many types of mistakes are very hard to notice.

To address these problems we designed and implemented TunedIT system (http://tunedit.org) – an integrated platform for automated evaluation of machine learning algorithms. Thanks to automation, TunedIT enables researchers to design and execute experiments that are fully repeatable and generate reproducible results. This system is presented in the following sections.

Previous attempts to address the aforementioned problems include: Delve software environment for evaluation of learning algorithms in valid experiments [4,3]; Experiment Databases for Machine Learning[1] (ExpDB) for collecting and sharing experimental results [1]; Machine Learning Open Source Software[2] (MLOSS) website for sharing implementations [5]; UCI Machine Learning Repository[3] for collecting and sharing datasets [2]; Computational Intelligence and Machine Learning (CIML) Community Portal[4] [7].

2 TunedIT System

TunedIT system combines three interrelated components (Fig. 1):

1. *TunedTester*: a Java application for automated evaluation of algorithms according to test specification provided by the user.

[1] http://expdb.cs.kuleuven.be/
[2] http://mloss.org/
[3] http://www.ics.uci.edu/~mlearn/
[4] http://www.cimlcommunity.org/

2. *Repository*: a database of machine learning resources. These include algorithms, datasets and evaluation procedures, which can be used by TunedTester to set up and execute experiments.
3. *Knowledge Base* (KB): a database of test results. On user's request, TunedTester may send results of tests to TunedIT. Here, results submitted by different researchers are merged into rich and comprehensive Knowledge Base that can be easily browsed for accurate and thorough information on specific algorithms or datasets.

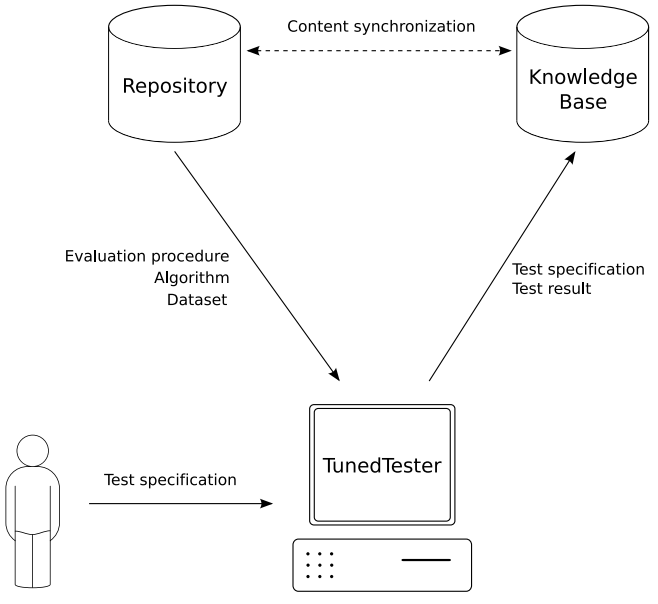


Fig. 1. Main components of TunedIT system and their interactions. *Repository*: a database of machine learning resources. *Knowledge Base*: a database of test results. *TunedTester*: an automated testing application. TunedTester takes a test specification from the user, downloads resources from Repository needed to set up the test, executes the test and sends the result together with the test specification to Knowledge Base. Additionally, upon each modification of the contents of Repository, such as deletion of a resource, the contents of Knowledge Base is synchronized accordingly, so that the results collected in Knowledge Base are always consistent with the current contents of Repository.

Repository and Knowledge Base reside on a server and can be accessed through TunedIT website at the following URL: http://tunedit.org. All registered users can upload resources to Repository, browse results collected in Knowledge Base and submit new results generated by TunedTester. Registration is open to everyone. TunedTester runs on a client computer, which typically is the user's local machine, and communicates with Repository and Knowledge Base through Internet.

2.1 Repository

Repository is a database of files – *resources* – related to machine learning and data mining. In particular, these include datasets, code of algorithms and evaluation procedures. Repository is located on TunedIT server and is accessible for all registered users – they can view and download resources, as well as upload new ones. The role of Repository in TunedIT is three-fold:

- It serves as a collection of algorithms, datasets and evaluation procedures that can be downloaded by TunedTester and used in tests.
- It provides space where users can share ML and DM resources with each other.
- It constitutes a context and point of reference for interpretation of results generated by TunedTester and logged in Knowledge Base. For instance, when the user is browsing KB and viewing results for a given test specification, he can easily navigate to corresponding resources in Repository and check their contents, so as to validate research hypotheses or come up with new ones. Thus, Repository is not only a convenient tool that facilitates execution of tests and sharing of resources, but - most of all - secures interpretability of results collected in Knowledge Base.

Repository has similar structure as a local file system. It contains a hierarchy of folders, which in turn contain files - resources. Upon registration, every user is assigned *home folder* in Repository's root folder, with its name being the same as the user's login. The user has full access to his home folder, where he can upload/delete files, create subfolders and manage access rights for resources. All resources uploaded by users have unique names (access paths in Repository) and can be used in TunedTester exactly in the same way as preexisting resources.

Access rights. Every file or folder in Repository is either *public* or *private*. All users can view and download public resources. Private files are visible only to the owner, while to other users they appear like if they did not exist - they cannot be viewed nor downloaded and their results do not show up at KB page. Private folders cannot be viewed by other users, although subfolders and files contained in them can be viewed by others, given that they are public themselves. In other words, the property of being private does not propagate from a folder to files and subfolders contained inside.

2.2 TunedTester

TunedTester (TT) is a Java application that enables fully automated evaluation of algorithms, according to test specification provided by the user. Single run of evaluation is called a *test* or *experiment* and corresponds to a triple of resources from Repository:

1. **Algorithm** is the subject of evaluation.
2. **Dataset** represents an instance of a data mining problem to be solved by the algorithm.
3. **Evaluation procedure** (EP) is a Java class that implements all steps of the experiment and, at the end, calculates a quality measure.

It is worth to note that the evaluation procedure is not hard-wired into TunedTester but is a part of test configuration just like the algorithm and dataset. Every user can implement new evaluation procedures to handle new kinds of algorithms, data types, quality measures or data mining tasks. In this way, TunedTester provides not only full automation of experiments, but also high level of flexibility and extendability.

TunedTester runs locally on user's computer. All resources that comprise the test are automatically downloaded from Repository. If requested, TunedTester can submit results of tests to Knowledge Base. They can be analysed later on with convenient web interface of KB.

All TunedIT resources are either files, like `UCI/iris.arff`, or Java classes contained in JAR files, like

`Weka/weka-3.6.1.jar:weka.classifiers.lazy.IB1` .

Typically, datasets have a form of files, while evaluation procedures and algorithms have a form of Java classes. For datasets and algorithms this is not a strict rule, though. To be executable by TunedTester, evaluation procedure must be a subclass of

`org.tunedit.core.EvaluationProcedure`

located in `TunedIT/core.jar` file in Repository. `TunedIT/core.jar` contains also

`ResourceLoader` and `StandardLoader`

classes, which can be used by the evaluation procedure to communicate with TunedTester environment and read the algorithm and dataset files. It is up to the evaluation procedure how the contents of these files is interpreted: as bytecode of Java classes, as a text file, as an ARFF, CSV or ZIP file etc. Thus, different evaluation procedures may expect different file formats and not every evaluation procedure must be compatible with a given algorithm or dataset. This is natural, because usually the incompatibility of file formats is just a reflection of more inherent incompatibility of resource types. There are many different types of algorithms – for classification, regression, feature selection, clustering etc. – and datasets – time series, images, graphs etc. – and each of them must be evaluated differently anyway. Nonetheless, it is also possible that the evaluation procedure supports several different formats at the same time.

Dataset file formats and algorithm APIs that are most commonly used in TunedIT and are supported by standard evaluation procedures include:

– ARFF file format for data representation. This format was introduced by Weka and became one of the most popular in machine learning community.

- Debellor's API defined by `org.debellor.core.Cell` class for implementation of algorithms.
- Weka's API defined by `weka.classifiers.Classifier` class for implementation of algorithms.
- Rseslib's API defined by `rseslib.processing.classification.Classifier` interface for implementation of algorithms.

It is also possible for a dataset to be represented by a Java class, the class exposing methods that return data samples when requested. This is a way to overcome the problem of custom file formats. If a given dataset is stored in atypical file format, one can put it into a JAR file as a Java resource and prepare a wrapper class that reads the data and returns samples in common representation, for example as instances of Debellor's `Sample` class.

Users may implement evaluation procedures that support any other file formats or algorithm APIs, not mentioned above. We also plan to extend this list in the future, so that basic evaluation procedures created by us can handle other formats and APIs.

More importantly, we would like to extend TunedTester with support for other programming languages, not only Java. Although this task will be more laborious, it is important for all the researchers who do not use Java for implementation of their algorithms.

Test specification. Test specification is a formal description for TunedTester of how the test should be set up. It is a combination of three identifiers – TunedIT *resource names* – of TunedIT resources which represent an evaluation procedure, an algorithm and a dataset that will be employed in the test:

$$\textit{Test specification} = \textit{Evaluation procedure} + \textit{Algorithm} + \textit{Dataset}$$

TunedIT resource name is a full access path to the resource in Repository, as it appears on Repository page. It does not include leading slash "/". For example, the file containing Iris data and located in UCI folder[5] has the following name:

`UCI/iris.arff`

Java classes contained in JARs are also treated as resources. TunedIT resource name of a Java class is composed of the containing JAR's name followed by a colon ":" and full (with package) name of the class. For instance, ClassificationTT70 class contained in `TunedIT/base/ClassificationTT70.jar` and `org.tunedit.base` package has the following name:

$$TunedIT/base/ClassificationTT70.jar : org.tunedit.base.ClassificationTT70$$

Resource names are case-sensitive.

Many algorithms expose a number of parameters that can be set by the user to control and modify algorithm's behavior. Currently, test specification does not

[5] See http://tunedit.org/repo/UCI/iris.arff

include values of parameters, and thus it is expected that the algorithm will apply default values. If the user wants to test an algorithm with non-default parameters he should write a wrapper class which internally invokes the algorithm with parameters set to some non-default values. The values must be hard-wired in the wrapper class, so that the wrapper itself does not expose any parameters. In the future we plan to add the ability to provide non-default parameter values directly in test specification.

Although the test specification has simple structure, it can represent a broad range of different tests. This is because the contents of the algorithm and dataset is interpreted by evaluation procedure, which is pluggable itself. If some kind of algorithm or dataset is not handled by existing evaluation procedures, a new procedure can be implemented for them. Thus, pluggability of evaluation procedures gives the power to test any kinds of algorithms on any kinds of data, while keeping the results interpretable thanks to simple and consistent test specifications.

Sandbox. Users of TunedTester may safely execute tests of any algorithms present in Repository, even if the code cannot be fully trusted. TunedTester exploits advanced features of Java Security Architecture to assure that the code executed during tests do not perform any harmful operation, like deleting files on disk or connecting through the network. Code downloaded from Repository executes in a *sandbox* which blocks the code's ability to interact with system environment. This is achieved through the use of a dedicated Java class loader and custom security policies. Similar mechanisms are used in web browsers to protect the system from potentially malicious applets found on websites.

Local cache. Communication between TunedTester and TunedIT server is efficient thanks to the cache directory which keeps local copies of resources from Repository. When the resource is needed for the first time and must be downloaded from the server, its copy is saved in the cache. In subsequent tests, when the resource is needed again, the copy is used instead. In this way, resources are downloaded from Repository only once. TunedTester detects if the resource has been updated in Repository and downloads the newest version in such case. Also, any changes introduced to the local copies of resources are detected, so it is not possible to run a test with corrupted or intentionally faked resources.

2.3 Knowledge Base

Knowledge Base (KB) is an open-access database containing test results from TunedTester, built collaboratively by TunedIT users. It is located on TunedIT server. Every user registered in TunedIT may submit results to KB by checking the "Send results to Knowledge Base" option in TunedTester GUI. Thanks to standardization and automation of tests, results submitted by different users are all comparable and thus can be merged together in a single database. In this way, TunedIT gives researchers an opportunity to build collectively an experiment database of unprecedented size and scope. This is impossible to achieve using previous approaches. For example, in the system by [1] only the administrators

have full access to the database and can insert new results. As of January 2010, KB contains over 150,000 atomic results and 6,700 aggregated results (see below for the definition of atomic and aggregated results).

To guarantee that results in KB are always consistent with the contents of Repository and that Repository can serve indeed as a context for interpreration of the results, when a new version of resource is uploaded, KB gets automatically cleaned out of all out-dated results related to the old version of the resource. Thus, there is no way to insert results into KB that are inconsistent with the contents of Repository.

Knowledge Base has a web interface that allows for easy querying and browsing the database.

2.4 Nondeterminism of Test Results

It is very common for machine learning implementations to include nondeterministic factors. For example:

- Evaluation procedures may split data randomly into training and test parts, which yields different splits in every trial. This is the case with standard procedures available in TunedIT: ClassificationTT70 and RegressionTT70[6].
- Algorithms for training of neural networks may perform random initialization of weights at the beginning of learning.
- Data samples may be generated randomly from a given probabilistic distribution, resulting in a different dataset in each repetition of the experiment.

Consequently, experiments executed with TunedTester may also include nondeterministic factors and generate different outcomes on every run. For this reason, instead of analysing single test outcomes, we need to analyse probabilistic distribution of results produced for a given test specification. To this end, TunedIT introduces the notions of *atomic result* and *aggregated result*.

Definition 1 (Atomic result). *Atomic result is the result of a single test executed by TunedTester for a given test specification.*

Atomic result gives a snapshot of algorithm's behavior in a single random scenario of the experiment. It is possible to execute many tests for the same specification and log all their results in KB. Thus, there can be many atomic results present in KB which correspond to the same specification.

Definition 2 (Aggregated result). *Aggregated result is the aggregation of all atomic results present in KB and corresponding to a given test specification. Here, aggregation means a set of statistics: arithmetic mean, standard deviation etc.*

There can be only one aggregated result for a given specification. Aggregated results are dynamically calculated at TunedIT server and presented on KB web

[6] See http://tunedit.org/repo/TunedIT/base

page[7]. Currently, users of TunedIT do not have direct access to atomic results stored in KB, but we plan to introduce this functionality in the future. At the moment, users can only see atomic results of their own experiments, in TunedTester, printed onto output console.

Presence of nondeterminism in experiments is highly desirable. If tests are fully deterministic, they always produce the same outcome and thus the aggregated result (mean) is the same as all atomic results, with standard deviation equal to zero. With nondeterminism, experimental results give broader knowledge about the tested algorithm – non-zero deviation measures how reliably and repeatably the algorithm behaves – and more reliable estimation of its expected quality (mean of multiple atomic results which are different between each other). Therefore, when implementing new algorithms, evaluation procedures or data generators, it is worth to introduce nondeterminism, if only this does not disturb essential functionality of the implementation. For instance, if an evaluation procedure splits the data into training and test subsets, it is better to perform this split at random instead of picking every time the same samples, e.g., with the *first* 70% of samples always falling into training subset.

2.5 Security Issues: Validity of Results

The user may assume that results generated by others and collected in KB are valid, in a sense that if the user runs the same tests by himself he would obtain the same expected results. In other words, results in KB can be trusted even if their authors – unknown users of TunedIT – cannot be trusted. This is possible thanks to numerous security measures built into Repository, TunedTester and KB, which ensure that KB contents cannot be polluted neither by accidental mistakes nor intentional fakery of any user. It is also worth to note that all results from KB can be easily verified, because corresponding test specifications are known and all involved resources are present in Repository, so reproducing an experiment is a matter of launching TunedTester, typing the test specification and pressing "Run...".

3 Conclusions

In this paper, we presented TunedIT system, which enables automated evaluation of machine-learning and data-mining algorithms; execution of repeatable experiments; sharing of experimental results and resources – datasets, algorithms, evaluation procedures – among researchers. The idea and design of such an integrated system, which supports all important aspects of experimentations in computational intelligence, is unique. TunedIT has already attracted researchers' attention and after just 5 months of functioning has 400 registered users. We hope that it will enjoy wide adoption in the scientific community and facilitate communication between researchers, exchange of ideas and design of yet better algorithms.

[7] See http://tunedit.org/results

Acknowledgements

The research has been partially supported by grants N N516 077837 and N N516 368334 from Ministry of Science and Higher Education of the Republic of Poland.

References

1. Blockeel, H., Vanschoren, J.: Experiment databases: Towards an improved experimental methodology in machine learning. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 6–17. Springer, Heidelberg (2007)
2. Newman, D.J., Hettich, S., Blake, C., Merz, C.: UCI repository of machine learning databases (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
3. Neal, R.: Assessing relevance determination methods using delve. In: Neural Networks and Machine Learning, pp. 97–129. Springer, Heidelberg (1998), http://www.cs.utoronto.ca/~radford/ftp/ard-delve.pdf
4. Rasmussen, C.E., Neal, R.M., Hinton, G.E., van Camp, D., Revow, M., Ghahramani, Z., Kustra, R., Tibshirani, R.J.: The DELVE Manual. University of Toronto, 1.1 edn. (December 1996), http://www.cs.utoronto.ca/~delve
5. Sonnenburg, S., Braun, M.L., Ong, C.S., Bengio, S., Bottou, L., Holmes, G., LeCun, Y., Müller, K.R., Pereira, F., Rasmussen, C.E., Rätsch, G., Schölkopf, B., Smola, A., Vincent, P., Weston, J., Williamson, R.C.: The need for open source software in machine learning. Journal of Machine Learning Research 8, 2443–2466 (2007)
6. Wojnarski, M.: Debellor: a data mining platform with stream architecture. In: Peters, J.F., Skowron, A., Rybiński, H. (eds.) Transactions on Rough Sets IX. LNCS, vol. 5390, pp. 405–427. Springer, Heidelberg (2008)
7. Zurada, J.M., Wojtusiak, J., Chowdhury, F., Gentle, J.E., Jeannot, C.J., Mazurowski, M.A.: Computational intelligence virtual community: Framework and implementation issues. In: IJCNN, pp. 3153–3157. IEEE, Los Alamitos (2008), http://www.mli.gmu.edu/jwojt/papers/08-5.pdf

Consensus Multiobjective Differential Crisp Clustering for Categorical Data Analysis

Indrajit Saha[1], Dariusz Plewczyński[1],
Ujjwal Maulik[2], and Sanghamitra Bandyopadhyay[3]

[1] Interdisciplinary Centre for Mathematical and Computational Modeling (ICM), University of Warsaw, 02-089 Warsaw, Poland
{indra,darman}@icm.edu.pl
[2] Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, West Bengal, India
drumaulik@cse.jdvu.ac.in
[3] Machine Intelligence Unit, Indian Statistical Institute, Kolkata-700108, West Bengal, India
sanghami@isical.ac.in

Abstract. In this article, an evolutionary crisp clustering technique is described that uses a new consensus multiobjective differential evolution. The algorithm is therefore able to optimize two conflicting cluster validity measures simultaneously and provides resultant Pareto optimal set of non-dominated solutions. Thereafter the problem of choosing the best solution from resultant Pareto optimal set is resolved by creation of consensus clusters using voting procedure. The proposed method is used for analyzing the categorical data where no such natural ordering can be found among the elements in categorical domain. Hence no inherent distance measure, like the Euclidean distance, would work to compute the distance between two categorical objects. Index-coded encoding of the cluster medoids (centres) is used for this purpose. The effectiveness of the proposed technique is provided for artificial and real life categorical data sets. Also statistical significance test has been carried out to establish the statistical significance of the clustering results. Matlab version of the software is available at http://bio.icm.edu.pl/∼darman/CMODECC.

Keywords: Crisp clustering, differential evolution, multiobjective optimization, Pareto optimal, statistical significance test.

1 Introduction

Clustering [1] is a useful unsupervised data mining technique which partitions the input space into K regions depending on some similarity/dissimilarity metric where the value of K may or may not be known a priori. K-means [1] is a traditional partitional clustering algorithm which starts with K random cluster centroids and the centroids are updated in successive iterations by computing the numerical averages of the feature vectors in each cluster. The objective of the K-means algorithm is to maximize the global compactness of the clusters. K-means clustering algorithm cannot be applied for clustering categorical data

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 30–39, 2010.

sets, where there is no natural ordering among the elements of an attribute domain. Thus no inherent distance measures, such as Euclidean distance, can be used to compute the distance between two feature vectors. Hence it is not feasible to compute the numerical average of a set of feature vectors. To handle such categorical data sets, a variation of K-means algorithm, namely K-medoids clustering has been proposed in [2]. In K-medoids algorithm, instead of computing the mean of feature vectors, a representative feature vector (cluster medoid) is selected for each cluster. A cluster medoid is defined as the most centrally located element in that cluster, i.e., the point from which the distance of the other points of the cluster is the minimum. K-medoids algorithm is also known as Partitioning Around Medoids (PAM) [2]. The major disadvantage of K-means and K-medoids clustering algorithms is that these algorithms often tend to converge to local optimum solutions.

In 1995 a new floating point encoded evolutionary algorithm for global optimization called Differential Evolution (DE) was proposed in [3] that uses a special kind of differential operator. Recently DE has found a wide spread application in different fields of engineering and science. Moreover, the K-means and K-medoids clustering algorithms optimize a single objective function which may not work equally well for different kinds of data sets. This fact motivated us to model a new evolutionary crisp clustering algorithm using DE on the multiobjective optimizations (MOO) framework, called consensus multiobjective differential evolution based crisp clustering (CMODECC), where search is performed over a number of, often conflicting, objective functions. Unlike single objective optimization, which yields a single best solution, in MOO, the final solution set contains a number of Pareto optimal solutions. But the problem of selecting the best solution among the Pareto optimal solutions is encountered and resolved by creation of consensus clusters using voting procedure. The two objective functions, the K-medoids error function [2] and the summation of separation among the cluster medoids, are optimized simultaneously where one has to be minimized and other has to be maximized for getting the proper partitions. The superiority of the proposed method over K-medoids, single objective version of both differential evolution and genetic algorithm based crisp clustering has been demonstrated on different synthetic and real life data sets. Also statistical significance tests have been carried out in order to confirm that the superior performance of the multiobjective clustering scheme is significant and has not occurred by chance.

2 Crisp Clustering Algorithms

This section describes some clustering algorithms used for categorical data.

2.1 K-medoids Clustering

Partitioning around medoids (PAM), also called K-medoids clustering [2], is a variation of K-means with the objective to minimize the within cluster variance $com(K)$.

$$com(K) = \sum_{i=1}^{K} \sum_{x \in C_i} D(x, m_i) \quad (1)$$

Here m_i is the medoid of cluster C_i and $D(x, m_i)$ denotes the distance between the point x and m_i. K denotes the number of clusters. $com(K)$ provides the resulting clustering of the data set X. The idea of PAM is to select K representative points, or medoids, in X and assign the rest of the data points to the cluster identified by the nearest medoid. Initial set of K medoids are selected randomly. Subsequently, all the points in X are assigned to the nearest medoid. In each iteration, a new medoid is determined for each cluster by finding the data point with minimum total distance to all other points of the cluster. After that, all the points in X are reassigned to their clusters in accordance with the new set of medoids. The algorithm iterates until $com(K)$ does not change any more.

2.2 Genetic Algorithm based Crisp Clustering

In Genetic Algorithm based Crisp Clustering (GACC), the chromosomes are represented as a vector of indices of the points which represent the medoids of the partitions. Each index point in a chromosome implies that the corresponding point is a cluster medoid [4]. If chromosome i encodes the medoids of K clusters then its length l is K. For initializing a chromosome, the K medoids are randomly selected index points from the data set while ensuring that they are distinct. The fitness of a chromosome indicates the degree of goodness of the solution it represents. For this purpose we used either $com(K)$ or $sep(K)$ as a cluster validity measure. The objective is therefore to minimize for achieving optimal clustering. Given a chromosome, the medoids encoded in it are first extracted. Let the chromosome encode K medoids, and let these be denoted as $z_1, z_2, \ldots, z_K$. The corresponding fitness is computed either using Eqn. 1 or Eqn. 7. The medoids encoded in a chromosome are updated by new set of medoids. Conventional proportional selection implemented by the roulette wheel strategy is applied on the population of chromosomes. The standard single point crossover is applied stochastically with probability μ_c. The cluster medoids are considered to be indivisible, i.e., the crossover points can only lie in between two clusters medoids. Each chromosome undergoes mutation with a fixed probability μ_m. The mutation operation has been defined as following: for the string to be mutated, a random element is chosen and it is replaced by a different index of point in the range $\{1, 2, \ldots, n\}$ such that no element is duplicated in the chromosome. The algorithm is terminated after it has executed a fixed number of generations. The elitist model of GAs has been used, where the best string seen so far is stored in a location within the population. The best string of the last generation provides the solution to the clustering problem.

2.3 Differential Evolution based Crisp Clustering

Differential Evolution based Crisp Clustering (DECC) algorithm also uses the same encoding policy as GACC to represent the chromosomes. The fitness of

each vector is computed either using Eqn. 1 or Eqn. 7. Subsequently, the medoids encoded in a vector are updated by set of new medoids. The process of mutation is computed as following:

$$\vartheta_{k,l}(t+1) = \vartheta_{m,l}(t) + F(\vartheta_{r,l}(t) - \vartheta_{p,l}(t)) \tag{2}$$

Here $\vartheta_{m,l}(t)$, $\vartheta_{r,l}(t)$ and $\vartheta_{p,l}(t)$ are randomly taken vectors from the current population (indicated by t time stamp) with the l dimensions for the mutant vector $\vartheta_{k,l}(t+1)$. F is the scaling factor usually $\in [0, 1]$. Note that if the index value of $\vartheta_{k,l}(t+1)$ lies beyond the permissible range of $\{1, \ldots, n\}$ then it is scaled using one of the following two operations

$$\vartheta_{k,l}(t+1) - n \tag{3}$$

and

$$\vartheta_{k,l}(t+1) + n \tag{4}$$

In order to increase the diversity of the perturbed parameter vectors, crossover is introduced and computed as following:

$$U_{jk,l}(t+1) = \begin{cases} \vartheta_{jk,l}(t+1) \\ \quad if\ rand_j(0,1) \leq CR\ or\ j = rand(i) \\ \vartheta_{jk,l}(t) \\ \quad if\ rand_j(0,1) > CR\ and\ j \neq rand(i) \end{cases} \tag{5}$$

In Eqn. (5), $rand_j(0,1)$ is the jth evaluation of a uniform random number generator with outcome $\in [0, 1]$. CR is the crossover constant $\in [0, 1]$ which has to be determined by the user. $rand(i)$ is a randomly chosen index $\in 1, 2, \ldots, l$ which ensures that $U_{k,l}(t+1)$ gets at least one parameter from $\vartheta_{k,l}(t+1)$. To make the population for the next generation, the trial vector $U_{k,l}(t+1)$ is compared to the target vector $\vartheta_{k,l}(t)$ using the greedy criterion. If vector $U_{k,l}(t+1)$ yields a smaller fitness value than $\vartheta_{k,l}(t)$, then $U_{k,l}(t+1)$ is set to $\vartheta_{k,l}(t)$; otherwise, the old value $\vartheta_{k,l}(t)$ is retained. DECC algorithm is also terminated after execution of fixed number of generations.

3 Proposed Consensus Multiobjective Differential Evolution based Crisp Clustering

In this section, we describe the proposed Consensus Multiobjective Differential Evolution based Crisp Clustering (CMODECC) technique in detail.

3.1 Vector Representation and Initial Population

The length of each vector is equal to the number of clusters K. Each element of the vector has a value chosen randomly from the set 1, 2, ..., n, where n is the number of points. Hence a string is represented as a vector of indices of the points in the data set. A vector is valid if no point index occurs more than once in the chromosome. The population is initialized by generating P such random strings, where P is the population size and it is fixed.

1. Initialize the vectors of the population.
2. Evaluate $com(K)$ and $sep(K)$ values for each parent vectors.
Repeat
   3. Mutation
   4. Crossover
   5. Evaluate $com(K)$ and $sep(K)$ values for each offspring vector.
   6. Combine parent vectors with offspring vectors to create new population.
   7. Perform Non-dominated sorting for assigning rank.
   8. Select the vectors of the combined population based on non-dominated lowest rank vectors of size same as population of the next generation.
Until (termination criteria are met)
9. Create consensus clusters from resultant Pareto optimal solutions using voting.

Fig. 1. CMODECC Algorithm

3.2 Computation of the Objectives

In this process, first the clusters are formed from a vector by taking the points encoded in it as the medoid points and assigning other points in the data set to their nearest medoids. After forming the clusters, new medoids for each cluster are found by selecting the most centrally located point of each cluster and the vector is updated with the indices of those medoids. Two objective functions used in this article are the K-medoids error function [2] and the summation of separations among the cluster medoids. The first objective function $com(K)$ is given in Eqn. 1. The second objective function is computed as following: let $C_1, C_2, \ldots, C_K$ be the medoids of the respective clusters and the objective function $sep(K)$ is defined as:

$$S(K) = \sum_{i=1}^{K-1} \sum_{j>i}^{K} D(C_i, C_j) \tag{6}$$

$$sep(K) = \frac{1}{S(K)} \tag{7}$$

The first objective function $com(K)$ is to be minimized to get compact clusters, whereas the second objective function $S(K)$ is to be maximized to get compact and well separated clusters. As the problem is modeled as minimization of objectives, we take the second objective as $sep(K) = \frac{1}{S(K)}$.

3.3 Other Processes

After evaluating the fitness of all vectors, it goes through mutation (described in Eqn. 2) to generate the new offspring and crossover (described in Eqn. 5) for increasing the diversity of the mutant vector. Note that if the index values of mutant vector lie beyond the permissible range of $\{1, \ldots, n\}$ then they are scaled either using Eqn. 4 or Eqn. 3. The created offspring pool combined with its parent

pool in the next step for performing the non-dominated sort [5]. Thereafter the selection process has been performed basing on the lowest rank assigned by the non-dominated sort as well as least crowding distance [5]. These processes are executed for a fixed number of iterations and final nondominated front is used for creation of consensus clusters using voting procedure. The different steps of CMODECC are shown in Fig. 1.

4 Experimental Results

4.1 Synthetic and Real Life Data Sets

Cat_100_8_3: This synthetic data set[1] has a one-layer clustering structure with 8 attributes and 100 points. It has 3 clusters.

Cat_300_15_5: This is a synthetic data set with 300 points and 15 attributes. The data set has 5 clusters.

Congressional Votes: This data set contains 435 number of records. Each row corresponds to one Congress man's votes on 16 different issues (e.g., education spending, crime etc.). A classification label of this real life data set[2] is Republican or Democrat which is provided with each data record.

Soybean: The Soybean data set contains 47 data points on diseases in soybeans and it is also a real life data set. Each data point has 35 categorical attributes and is classified as one of the four diseases, i.e., number of clusters in the data set is 4.

4.2 Performance Metrics

Performance of the proposed method is evaluated by the measure of Adjusted Rand Index (ARI) [6], DB-index [7] and Dunn's index [8]. Note that, measuring the DB and Dunn's indices for categorical data, the medoid are taken into consideration instate of mean.

4.3 Distance Measures

Distance between two categorical objects is computed as in [9]. Let two categorical objects described by p categorical attributes are $x_i = [x_{i1}, x_{i2}, \ldots, x_{ip}]$, and $x_j = [x_{j1}, x_{j2}, \ldots, x_{jp}]$. The distance measure between x_i and x_j , $D(x_i, x_j)$, can be defined by the total number of mismatches of the corresponding attribute categories of the two objects. Formally,

$$D(x_i, x_j) = \sum_{k=1}^{p} \delta(x_{ik}, x_{jk}) \tag{8}$$

where

$$\delta(x_{ik}, x_{jk}) = \begin{cases} 0 & \text{if } x_{ik} = x_{jk} \\ 1 & \text{if } x_{ik} \neq x_{jk} \end{cases} \tag{9}$$

[1] http://www.datgen.com
[2] http://www.ics.uci.edu/∼mlearn/MLRepository.html

4.4 Input Parameters

The DE (both for single objective and multiobjective) and GA (for single objective) based algorithms are executed for 100 generations with fixed population size = 50. The crossover probability and mutation factors (F) for DE (both for single objective and multiobjective) based algorithms are set to be 0.8 and 1, respectively. The crossover and mutation probabilities for GA (for single objective) based algorithms are taken to be 0.8 and 0.3, respectively. The K-medoids algorithm is executed till it converges to the final solution. Results reported in the tables are the average values obtained over 20 runs of the algorithms.

Table 1. Average ARI values over 20 runs of different algorithms for four categorical data sets

Algorithms	Cat_100_8_3	Cat_300_15_5	Votes	Soybean
CMODECC	0.8102	0.8273	0.9472	0.9901
DECC ($com(K)$)	0.7605	0.7508	0.9085	0.9793
DECC ($sep(K)$)	0.7872	0.7482	0.8702	0.9831
GACC ($com(K)$)	0.7425	0.7171	0.8653	0.9237
GACC ($sep(K)$)	0.7514	0.7002	0.8502	0.9302
K-medoids	0.6302	0.6482	0.7602	0.8302

4.5 Performance

Table1 and Table2 show the comparative results obtained for the four data sets. It can be noted from the table that the proposed method consistently outperforms the single objective as well as K-medoids algorithms in terms of the ARI, DB-index and Dunn's index score. It is also evident from the Table1 and Table2, that individually neither the $com(K)$ nor $sep(K)$ objective optimization is sufficient for proper clustering. For example, for Cat_300_15_5, the proposed CMODECC technique achieves better average AIR score of 0.8273 while the DECC ($com(K)$), DECC ($sep(K)$), GACC ($com(K)$), GACC ($sep(K)$) and K-medoids provide values of 0.750, 0.7482, 0.7171, 0.7002 and 0.6482, respectively. Moreover, in Table2, the DB and Dunn's indices values are promising for this

Table 2. Average DB-index and Duun's index values over 20 runs of different algorithms for four categorical data sets

Algorithms	Cat_100_8_3		Cat_300_15_5		Votes		Soybean	
	DB	Dunn	DB	Dunn	DB	Dunn	DB	Dunn
CMODECC	1.5551	1.8974	1.7247	2.0204	1.3621	1.7936	1.0488	1.4083
DECC ($com(K)$)	1.7305	1.7508	1.8082	1.8225	1.4803	1.6225	1.3046	1.2253
DECC ($sep(K)$)	1.6614	1.8052	1.8504	1.7536	1.5302	1.5603	1.2554	1.3077
GACC ($com(K)$)	1.9473	1.4284	2.0553	1.5004	1.7431	1.3883	1.5735	1.0405
GACC ($sep(K)$)	1.8052	1.5071	2.1702	1.4573	1.8225	1.3304	1.5004	1.1725
K-medoids	2.3362	1.0802	2.5082	1.2062	2.2081	0.9402	1.9792	0.8503

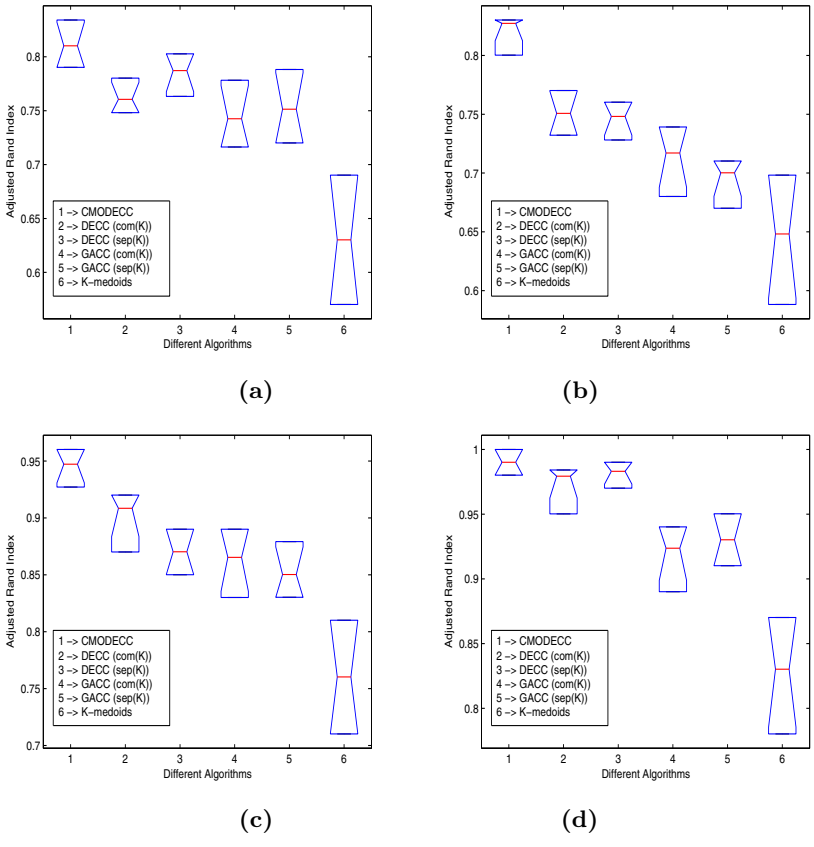


Fig. 2. Boxplot of ARI values of different clustering algorithms for (a) Cat_100_8_3 (b) Cat_300_15_5 (c) Votes (d) Soybean

data set. Similar results are also found for the other data sets. It is shown graphically as boxplots in Fig. 2 that the results in terms of ARI scores produced by several algorithms on the synthetic and real life data sets. However, the solutions generated by the CMODECC method is better than that produced by the other algorithms. The median values of ARI scores given by the proposed technique is superior than that for all other algorithms.

4.6 Statistical Significance Test

In this article, a statistical significance test called t-test [10] has been carried out at the 5% significance level, to establish that the better average ARI scores provided by CMODECC is statistically significant and does not come by chance.

Table 3 reports the results of the t-test for the four data sets. The null hypothesis (The means of two groups are equal) are shown in the table. The alternative

Table 3. The t-test results for four categorical data sets

Data Sets	Test No.	Null hypothesis (H0 : $\mu_1 = \mu_2$)	t-test statistic	P-value	Accept/Reject
Cat_100_8_3	1	$\mu_{CMODECC} = \mu_{DECC(com(K))}$	21.0473	3.0370e-009	Reject
	2	$\mu_{CMODECC} = \mu_{DECC(sep(K))}$	18.6508	1.6796e-008	Reject
	3	$\mu_{CMODECC} = \mu_{GACC(com(K))}$	27.8760	4.7790e-012	Reject
	4	$\mu_{CMODECC} = \mu_{GACC(sep(K))}$	23.0043	2.6340e-010	Reject
	5	$\mu_{CMODECC} = \mu_{K-medoids}$	30.9301	1.8915e-014	Reject
Cat_300_15_5	1	$\mu_{CMODECC} = \mu_{DECC(com(K))}$	7.6713	3.0901e-005	Reject
	2	$\mu_{CMODECC} = \mu_{DECC(sep(K))}$	9.1419	6.3482e-006	Reject
	3	$\mu_{CMODECC} = \mu_{GACC(com(K))}$	9.3309	7.5111e-006	Reject
	4	$\mu_{CMODECC} = \mu_{GACC(sep(K))}$	10.6626	4.7580e-007	Reject
	5	$\mu_{CMODECC} = \mu_{K-medoids}$	13.9226	2.1515e-009	Reject
Votes	1	$\mu_{CMODECC} = \mu_{DECC(com(K))}$	16.4575	5.0292e-007	Reject
	2	$\mu_{CMODECC} = \mu_{DECC(sep(K))}$	21.3666	1.0649e-008	Reject
	3	$\mu_{CMODECC} = \mu_{GACC(com(K))}$	23.7970	5.9505e-009	Reject
	4	$\mu_{CMODECC} = \mu_{GACC(sep(K))}$	37.8128	3.1403e-011	Reject
	5	$\mu_{CMODECC} = \mu_{K-medoids}$	71.0916	1.0898e-013	Reject
Soybean	1	$\mu_{CMODECC} = \mu_{DECC(com(K))}$	8.0625	8.3797e-005	Reject
	2	$\mu_{CMODECC} = \mu_{DECC(sep(K))}$	6.7442	2.4201e-005	Reject
	3	$\mu_{CMODECC} = \mu_{GACC(com(K))}$	18.1986	5.0840e-008	Reject
	4	$\mu_{CMODECC} = \mu_{GACC(sep(K))}$	17.2767	1.2880e-008	Reject
	5	$\mu_{CMODECC} = \mu_{K-medoids}$	22.9460	2.6939e-009	Reject

hypothesis is that the mean of the first group is larger than the mean of the second group. For each test, the degree of freedom is $M + N$ - 2, where M and N are the sizes of two groups considered. Here $M = N = 20$. Hence the degree of freedom is 38. Also the values of t-statistic and the probability (P-value) of accepting the null hypothesis are shown in the table. It is clear from the table that the P-values are much less than 0.05 (5% significance level) which are strong evidences for rejecting the null hypothesis. This proves that the better average ARI values produced by the CMODECC scheme is statistically significant and has not come by chance.

5 Conclusion

Majority of the clustering algorithms designed for categorical data optimize a single objective function, which may not be equally applicable for different kinds of data sets. Moreover the differential evolution is used for crisp clustering of categorical data in a multiobjective optimization framework is a new contribution to this field. Also the problem of selecting the best solution among non-dominated Pareto optimal solutions is solved by the creation of consensus clusters using voting technique. The proposed technique optimizes $com(K)$ and $sep(K)$ simultaneously. The superiority of the proposed scheme has been demonstrated on a number of synthetic and real life data sets. Also statistical significance test has been conducted to judge the statistical significance of the clustering solutions produced by different algorithms. In this regard results have been shown quantitatively and visually.

Acknowledgment

This work was supported by the Polish Ministry of Education and Science (grants N301 159735, N518 409238, and others).

References

1. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs (1988)
2. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data, An Itroduction to Cluster Analysis. John Wiley and Sons, Brussels (1990)
3. Storn, R., Price, K.: Differential evolution - A simple and efficient heuristic strategy for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359 (1997)
4. Maulik, U., Bandyopadhyay, S.: Genetic algorithm based clustering technique. Pattern Recognition 33, 1455–1465 (2000)
5. Deb, K., Agrawal, S., Pratab, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
6. Yeung, K., Ruzzo, W.: An empirical study on principal component analysis for clustering gene expression data. Bioinformatics 17, 763–774 (2001)
7. Davies, L.D., Bouldin, W.D.: A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1(2), 224–227 (1979)
8. Dunn, J.C.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. Journal of Cybernetics 3(3), 32–57 (1973)
9. Vermeulen-Jourdan, L., Dhaenens, C., Talbi, E.G.: Clustering nominal and numerical data: a new distance concept for an hybrid genetic algorithm. In: Gottlieb, J., Raidl, G.R. (eds.) EvoCOP 2004. LNCS, vol. 3004, pp. 220–229. Springer, Heidelberg (2004)
10. Ferguson, G.A., Takane, Y.: Statistical analysis in psychology and education (2005)

Probabilistic Rough Entropy Measures in Image Segmentation

Dariusz Małyszko and Jarosław Stepaniuk

Department of Computer Science
Bialystok University of Technology
Wiejska 45A, 15-351 Bialystok, Poland
{d.malyszko,j.stepaniuk}@pb.edu.pl

Abstract. In numerous data clustering problems, the main priority remains a constant demand on development of new improved algorithmic schemes capable of robust and correct data handling. This requirement has been recently boosted by emerging new technologies in data acquisition area. In image processing and image analysis procedures, the image segmentation procedures have the most important impact on the image analysis results.

In data analysis methods, in order to improve understanding and description of data structures, many innovative approaches have been introduced. Data analysis methods always strongly depend upon revealing inherent data structure. In the paper, a new algorithmic **R**ough **E**ntropy **F**ramework - (REF, in short) has been applied in the probabilistic setting. Crisp and Fuzzy RECA measures (**R**ough **E**ntropy **C**lustering **A**lgorithm) introduced in [5] are extended into probability area. The basic rough entropy notions, the procedure of rough (entropy) measure calculations and examples of probabilistic approximations have been presented and supported by comparison to crisp and fuzzy rough entropy measures. In this way, uncertainty measures have been combined with probabilistic procedures in order to obtain better insight into data internal structure.

1 Introduction

Data clustering routines have emerged as most prominent and important data analysis methods that are primarily applied in unsupervised learning and classification problems. Most often data clustering presents descriptive data grouping that identifies homogenous groups of data objects on the basis of the feature attributes assigned to clustered data objects. In this context, a cluster is considered as a collection of similar objects according to predefined criteria and dissimilar to the objects belonging to other clusters.

Fuzzy sets perform data analysis on the assumption, that data objects may belong in some degree not only to one concept or class but may partially participate in other classes. Rough set theory on the other hand assigns objects to class lower and upper approximations on the base of complete certainty about object

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 40–49, 2010.

belongingness to the class – lower approximation, and on the determination of the possible belongingness to the class – upper approximation. Probabilistic approaches have been introduced to rough set theory in several settings, including decision-theoretic analysis, variable precision analysis, and information-theoretic analysis. Most often, probabilistic methods are based on rough membership functions and rough inclusion functions.

The presented research is based on combining the concept of rough sets and entropy measure in the area of image segmentation in the introduced **R**ough **E**ntropy **F**ramework. The rough entropy concept has been introduced in [7], [4] and extended in [2], [3] and [5]. The REF platform deals with the data analysis approaches incorporating the concept of rough entropy measures. The application of the REF platform in image analysis routines is performed by means of **R**ough **E**ntropy **C**lustering **A**lgorithm. The REF imposes the theoretical and algorithmic structure. On the other hand, RECA schemes define how practically data object assignment to cluster approximations is performed and subsequently the way the approximation measures are calculated.

Probability rough measures are assessed relative to correlation degree with standard segmentation quality measures and rough entropy measures. Probability rough measures are considered as a type of rough entropy measures. In approaches based on probability rough measures all data points are assigned to predefined number of centers. Each cluster center has lower and upper approximations. Data points are assigned to lower and upper approximations of the cluster on the basis of data points probability of belonging to the cluster center.

This paper is structured in the following way. In Section 2 the introductory information about rough sets, rough sets extensions and rough measures has been presented. In Section 3 RECA concepts and probabilistic RECA measures have been described. Presentation and experimental material has been included and discussed in Section 4 followed by concluding remarks.

2 Rough Set Data Models

2.1 Rough Set Theory Essentials

Information granules [10] are considered and interpreted as linked collections of objects, for example data points, drawn together by the criteria of indistinguishability, similarity or functionality. Information granules and the ensuing process of information granulation is a powerful medium of abstraction leading to the emergence of high-level concepts. In this context, a granule most often is defined as a closely coupled group or clump of objects such as for example, pixels in image processing.

An information system is a pair (U, A) where U represents a non-empty finite set called the universe and A a non-empty finite set of attributes. Let $B \subseteq A$ and $X \subseteq U$. Taking into account these two sets, it is possible to approximate the set X making only the use of the information contained in B by the process of construction of the lower and upper approximations of X and further to express numerically the roughness $R(AS_B, X)$ of a set X with respect to B by assignment

$$R(AS_B, X) = 1 - \frac{card(LOW(AS_B, X))}{card(UPP(AS_B, X))}. \quad (1)$$

In this way, the value of the roughness of the set X equal 0 means that X is crisp with respect to B, and conversely if $R(AS_B, X) > 0$ then X is rough (i.e., X is vague with respect to B). Detailed information on rough set theory is provided in [9,10,12].

Some probabilistic rough set theory extensions are presented in [1], [13]. Variable precision rough set model improves upon rough set theory by the change of the subset operator definition. The Variable precision rough set (VPRS) model has been designed to analysis and recognition of statistical data patterns rather than functional trends. In the variable precision rough set setting, the objects are allowed to be classified within an error not greater than a predefined threshold. Other probabilistic extensions include decision-theoretic framework and Bayesian rough set model.

2.2 Rough Entropy Clustering Framework - REF

Rough entropy framework in image segmentation has been primarily introduced in [7] in the domains of image thresholding routines. In [7], rough set notions, such as lower and upper approximations have been put into image thresholding domain. The introduced rough entropy measure has been applied during image thresholding to two objects: foreground and background object. This type of thresholding has been extended into multilevel thresholding for one-dimensional and two-dimensional domains. In the research [4] rough entropy notion have been extended into multilevel granular rough entropy evolutionary thresholding of 1D data in the form of 1D MRET algorithm. Additionally, in [2] the authors extend this algorithm into 2D MRET thresholding routine of 2D image data. Further, rough entropy measures have been employed in image data clustering setting in [3], [5] described as **R**ough **E**ntropy **C**lustering **A**lgorithm.

3 Rough Entropy Measures

3.1 General REF and RECA Concepts

Rough entropy considered as a measure of quality for data clustering gives possibility and theoretical background for development of robust clustering schemes. These clustering algorithms incorporate rough set theory, fuzzy set theory and entropy measure. Three basic rough properties that are applied in clustering scheme include: selection of the threshold metrics (crisp, fuzzy, probabilistic) - tm , the threshold type (thresholded or difference based) - tt - and the measure for lower and the upper approximations - ma - (crisp, fuzzy, probabilistic). Data objects are assigned to lower and upper approximation on the base of the following criteria:

1. assignment performed on the basis of the distance to cluster centers within given threshold value,
2. assignment performed on the basis of the difference of distances to the cluster centers within given threshold value.

The combination of these three data structure properties makes possible the design of distinct rough entropy measures and algorithms optimizing these measures. Rough entropy value should be maximized. The search for cluster centers with optimal high rough entropy values is possible by coupling the evolutionary algorithm as described in Algorithm 1. Selection of cluster centers makes possible to determine approximations for each cluster. Data objects are assigned to approximations, each approximation is given a measure. Afterwards, the cluster roughness and rough entropy are calculated as given in Algorithm 2.

Algorithm 1. General *RECA* Algorithm Flow

Data: **Input Image,** k **– number of clusters,** *Size* **– number of chromosomes in evolutionary population**
Result: **Optimal Cluster Centers**

Create X **population with** *Size* **random chromosomes (solutions) each encoding** k **cluster centers**
repeat
  forall *chromosomes of* X **do**
    Calculate their Rough_Entropy
  end
  Create mating pool Y **from parental** X **population**
  Apply selection, cross-over and mutation to Y **population**
  Handle empty clusters
  Replace X **population with** Y **population**
until *termination criteria (most often predefined number of iterations)* ;

Algorithm 2. Rough_Entropy Calculation

Data: **Rough Approximations**
Result: **R - Roghness, RE - Rough Entropy Value**

for $l = 1$ **to** k *(number of data clusters)* **do**
  if Upper(C_l) **!= 0 then R**(C_l) **= 1 - Lower**(C_l) **/ Upper**(C_l)
end
RE = 0
for $l = 1$ **to** k *(number of data clusters)* **do**
  if R(C_l) **!= 0 then RE = RE -** $\frac{exp}{2}$ $\cdot$ **R**(C_l) $\cdot$ log**(R**(C_l)**)** ;
  (see Eq. 1)
end

3.2 Introduction to Probabilistic RECA Measures

Introduced probability rough measures are primarily based on the similar pattern as in case of crisp and fuzzy RECA measures. The probability rough measures

require selecting adequate probability measure. Data points closest to the given cluster center relative to the selected threshold metrics (crisp, fuzzy, probabilistic) are assigned to its lower and upper approximation. The upper approximations are calculated in the specific, dependant upon threshold type and measure way presented in the subsequent paragraphs. Probability distributions in RECA measures are required during measure calculations of probabilistic distance between data objects and cluster centers. Gauss distribution has been selected as probabilistic distance metric for data point $x_i \in U$ to cluster center C_m calculated as follows

$$d_{pr}(x_i, C_m) = (2\pi)^{-d/2} |\Sigma_m|^{-1/2} exp\left(-\frac{1}{2}(x_i - \mu_m)^T \Sigma_m^{-1}(x_i - \mu_m)\right) \tag{2}$$

where $|\Sigma_m|$ is the determinant of the covariance matrix Σ_m and the inverse covariance matrix for the C_m cluster is denoted as Σ_m^{-1}. Data dimensionality is denoted as d. In this way, for standard color RGB images $d = 3$, for gray scale images $d = 1$. Mean value for Gauss distribution of the cluster C_m has been denoted as μ_m. The summary of the parameters for the probabilistic RECA measures has been given in Table 1.

Table 1. Probabilistic - difference and threshold based measures and related algorithms - RECA

Difference metric based measures			
Algorithm	Measure	Threshold	Condition
CPDRECA	1	pr	$\lvert d_{pr}(x_i, C_m) - d_{pr}(x_i, C_l)\rvert \leq \epsilon_{pr}$
FPDRECA	μ_{C_m}	pr	$\lvert d_{pr}(x_i, C_m) - d_{pr}(x_i, C_l)\rvert \leq \epsilon_{pr}$
PPDRECA	$d_{pr}(x_i, C_m)$	pr	$\lvert d_{pr}(x_i, C_m) - d_{pr}(x_i, C_l)\rvert \leq \epsilon_{pr}$
PCDRECA	$d_{pr}(x_i, C_m)$	crisp	$\lvert d(x_i, C_m) - d(x_i, C_l)\rvert \leq \epsilon_{dist}$
PFDRECA	$d_{pr}(x_i, C_m)$	fuzzy	$\lvert(\mu_{C_m}(x_i) - \mu_{C_l}(x_i)\rvert \leq \epsilon_{fuzz}$
Threshold metric based measures			
Algorithm	Measure	Threshold	Condition
CPTRECA	1	pr	$d_{pr}(x_i, C_m) \geq \epsilon_{pr}$
FPTRECA	μ_{C_m}	pr	$d_{pr}(x_i, C_m) \geq \epsilon_{pr}$
PPTRECA	$d_{pr}(x_i, C_m)$	pr	$d_{pr}(x_i, C_m) \geq \epsilon_{pr}$
PCTRECA	$d_{pr}(x_i, C_m)$	crisp	$d(x_i, C_m) \leq \epsilon_{dist}$
PFTRECA	$d_{pr}(x_i, C_m)$	fuzzy	$\mu_{C_m}(x_i) \geq \epsilon_{fuzz}$

Fuzzy membership value $\mu_{C_l}(x_i) \in [0, 1]$ for the data point $x_i \in U$ in cluster C_l is given as

$$\mu_{C_l}(x_i) = \frac{d(x_i, C_l)^{-2/(\mu-1)}}{\sum_{j=1}^{k} d(x_i, C_j)^{-2/(\mu-1)}} \tag{3}$$

where a real number $\mu > 1$ represents fuzzifier value and $d(x_i, C_l)$ denotes distance between data object x_i and cluster (center) C_l.

3.3 Probabilistic RECA

(Cr, Fz, Pr)(Pr)(D) RECA

Crisp, Fuzzy, Pr – Pr probabilistic threshold, difference metric - (Cr, Fz, Pr) PDRECA algorithms. Rough measure general calculation routine has been given in Algorithms 2, 3. For each data object x_i, distance to the closest cluster C_l is denoted as $d_{pr}(x_i, C_l)$ - the measures of the lower (denoted as Lower) and upper (denoted as Upper) approximations for this cluster are increased by the PM value (as described in Algorithms 5, 4 and 3). Additionally, the measures of the upper approximations of the clusters C_m that satisfy the condition

$$|d_{pr}(x_i, C_m) - d_{pr}(x_i, C_l)| \leq \epsilon_{pr} \tag{4}$$

are increased by the PM value. Calculation of probabilistic distances requires prior all data object assignment to the closest clusters and determination of the clusters mean values and covariance matrices.

Algorithm 3. *RECA* **Approximations**

Data: **Image, Cluster Centers, M** $\in$ **{Cr, Fz, Pr}, S** $\in$ **{Difference, Threshold}**
Result: **RECA Approximations**

Prepare data structures
Assign data points to the clusters
foreach *Cluster* C_m **do**
  Calculate mean value μ_m **and covariance matrix** Σ_m
end
foreach *Data object* x_i **do**
  Determine the closest cluster C_l **for** x_i
  Determine the measure $PM(x_i, C_l, M)$
  Increment Lower(C_l) **and Upper**(C_l) **by** PM
  Determine the measure UA(x_i, **S, M**)
end

Algorithm 4. PM-Measure

Data: **M** $\in$ **{Cr, Fz, Pr},** x_i, C_m
Result: $PM(x_i, C_m, M)$
if $M = Cr$ **then return** PM = 1.0;
else if $M = Fz$ **then return** PM = $\mu_{C_m}(x_i)$ (see Eq. 3) ;
else return PM = $d_{pr}(x_i, C_m)$ (see Eq. 2) ;

(Cr, Fz, Pr)(Pr)(T) RECA

Crisp, Fuzzy, Pr – Pr probabilistic threshold, threshold metric - (Cr, Fz Pr) PTRECA algorithms. Rough measure general calculation follows steps outlined in Algorithm 3. For each data point x_i, distance to the closest cluster C_l is

Algorithm 5. UA-Measure

Data: **M ∈ {Cr, Fz, Pr}, S ∈ {Difference, Threshold}**, x_i, C_l
Result: **Upper Approximation Measure**

if $S = Threshold$ **then**
foreach *Cluster* $C_m \neq C_l$ *with* $d_M(x_i, C_m) \geq_M \epsilon_M$ **do**
  if $(d_{M=Cr}(x_i, C_m) \leq \epsilon_{M=Cr})$ *OR* $(d_{M=Fz,Pr}(x_i, C_m) \geq \epsilon_{M=Fz,Pr})$ **then**
    Determine the measure $PM(x_i, C_m, M)$
    Increment Upper(C_m) **by** PM
  end
end
else
foreach *Cluster* $C_m \neq C_l$ *with* $|d_M(x_i, C_m) - d_M(x_i, C_l)| \leq \epsilon_M$ **do**
  Determine the measure $PM(x_i, C_m, M)$
  Increment Upper(C_m) **by** PM
end

denoted as $d_{pr}(x_i, C_l)$ - the measures of the lower and upper approximations for this cluster are increased by the PM value (as described in Algorithm 4).

$$d_{pr}(x_i, C_m) \geq \epsilon_{pr} \tag{5}$$

are increased by the PM value. Similar to the Pr Difference RECA, the determination of the clusters mean values and covariance matrices is required for proper probabilistic distance calculations.

(Pr)(Cr, Fz)(D, T) RECA

In this algorithm type, the operations performed are analogous to the ones described in two previous subsections. The only difference consists in approximation measure that is probabilistic and threshold metric that is crisp and fuzzy. Conditions in Algorithm 5 should be accommodated for crisp and fuzzy thresholds.

4 Experimental Setup and Results

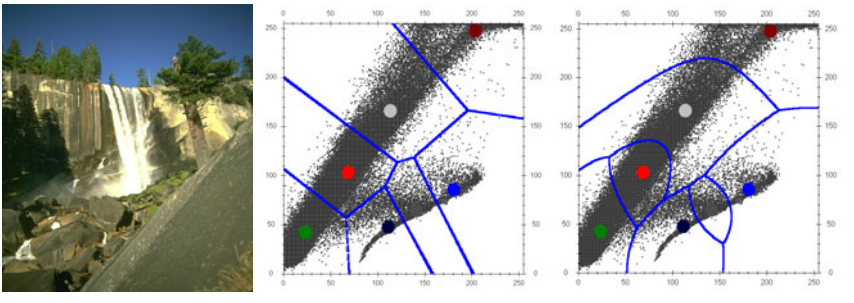
Image data - in the subsequent material, the image 27059 from Berkeley image database [6] has been selected for experiments. The original color RGB image as given in Figure 1 (a) have been preprocessed and 2D data for bands R and B have been obtained. In Figure 1 (b) and 1 (c) data points in 2D attribute domain R-B are displayed as single gray points.

Cluster centers - for the presentation purposes, 7 cluster centers have been created as shown in Figure 1 (b) and 1 (c) and displayed in distinct colors. Data points assignment and border of data points that are the closest to the given clusters depends heavily on the metric. In Figure 1 (b) cluster borders and cluster assignment (it means data points belonging to the nearest cluster center)

have been given in case of crisp or fuzzy metric. In Figure 1 (c) cluster borders and cluster assignment have been given in case of probabilistic metric.

RECA parameters. In the experimental material, introduced probabilistic RECA measures have been compared to crisp and fuzzy RECA measures. In the subsequent material, the following parameters have been selected for the calculated RECA measures. Crisp threshold $\epsilon_{dist} = 15$, fuzzy threshold $\epsilon_{fuzz} = 0.2$, probabilistic threshold for PTRECA - $\epsilon_{pr} = 5E\text{-}11$ and probabilistic difference threshold for PDRECA $\epsilon_{pr} = 5E\text{-}4$.

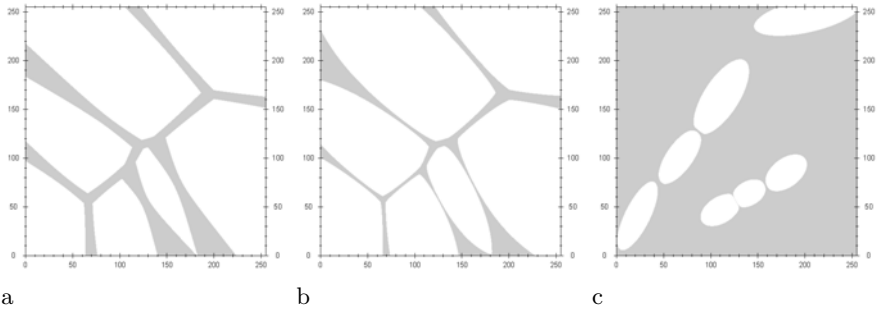
In order to make the presented concepts more understandable, in Figure 2 three types of boundary regions have been displayed: (a) crisp boundary, (b) fuzzy boundary and (c) probabilistic boundary.



a b c

Fig. 1. Berkeley dataset image: (a) 27059, (b) image 27059, R-B bands: crisp and fuzzy data object assignment, (c) image 27059, R-B bands: probabilistic data object assignement

In the experiments, the image 27059 R-B bands have been segmented by means of k-means algorithm with selected number of clusters $k = 7$. Afterwards, six solutions have been presented for experimental tests. For each k-means solution with ID 1 - 6, adequate RECA measures have been presented in Table 2. In addition to rough entropy measures (their values should be maximized), Dunn



a b c

Fig. 2. Berkeley 27059 dataset image, R-B bands: (a) crisp boundary regions, (b) fuzzy boundary regions, (c) probabilistic boundary regions

Table 2. Three standard indices: Dunn, Davies-Bouldin and β-index compared to rough entropy crisp, fuzzy and probabilistic RECA measures for the experimental 27059 R-B image

	Population					
ALG	1	2	3	4	5	6
Dunn	0.71	0.94	0.94	0.97	0.91	0.50
DB	8.41	8.42	8.20	8.10	8.40	8.10
β-index	12.86	13.60	13.48	13.11	12.08	10.83
CrPrT RECA	2.47	2.58	2.57	2.57	2.49	2.64
FzPrT RECA	0.86	0.89	0.85	0.85	0.88	0.93
PrPrT RECA	1.43	1.81	1.70	1.66	1.41	1.70
CrPrD RECA	2.05	2.09	2.12	2.13	2.15	2.05
FzPrD RECA	0.31	0.39	0.34	0.33	0.31	0.41
PrPrD RECA	0.03	0.05	0.03	0.05	0.04	0.02
CrFzT RECA	1.22	1.20	1.89	1.18	1.19	1.16
FzFzT RECA	1.11	1.25	1.18	1.14	1.12	1.16
PrFzT RECA	0.87	1.07	0.99	0.95	0.87	0.97
CrFzD RECA	0.84	0.82	0.82	0.82	0.85	0.88
FzPrD RECA	0.79	0.84	0.79	0.79	0.81	0.87
PrFzD RECA	0.20	0.15	0.16	0.17	0.20	0.15
CrCrT RECA	2.30	2.55	2.25	2.35	2.50	2.33
FzCrT RECA	1.16	1.22	1.05	1.11	1.28	1.00
PrCrT RECA	1.34	1.54	1.28	1.26	1.35	1.35
CrCrD RECA	2.38	2.63	2.48	2.44	2.37	2.37
FzCrD RECA	1.31	1.46	1.35	1.33	1.32	1.32
PrCrD RECA	1.00	1.22	1.08	1.02	0.99	1.06

index and β-index values are displayed (also should be maximized) and Davies-Bouldin index (their values should be minimized). High correlation between standard and crisp, fuzzy and probabilistic rough measures is easily seen.

5 Conclusions and Future Research

In the study, a probabilistic extension of the crisp and fuzzy rough (entropy) measures have been introduced and presented. Probabilistic rough entropy measures are capturing and revealing data structure properties that seems to be complementary to crisp and fuzzy rough measures. High correlation between standard measures on one hand and between different types of RECA measures on the other hand supports and confirms their suitability in the area of data analysis. Combination of crisp, fuzzy and probabilistic rough entropy measures seems to be promising emerging area of image segmentation and analysis.

Acknowledgments

The research is supported by the grants N N516 0692 35 from the Ministry of Science and Higher Education of the Republic of Poland.

References

1. Katzberg, J.D., Ziarko, W.: Variable Precision Extension of Rough Sets. Fundamenta Informaticae 27, 155–168 (1996)
2. Malyszko, D., Stepaniuk, J.: Granular Multilevel Rough Entropy Thresholding in 2D Domain. In: 16th International Conference Intelligent Information Systems, IIS 2008, Zakopane, Poland, June 16-18, pp. 151–160 (2008)
3. Małyszko, D., Stepaniuk, J.: Standard and Fuzzy Rough Entropy Clustering Algorithms in Image Segmentation. In: Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) RSCTC 2008. LNCS (LNAI), vol. 5306, pp. 409–418. Springer, Heidelberg (2008)
4. Malyszko, D., Stepaniuk, J.: Adaptive multilevel rough entropy evolutionary thresholding. Information Sciences 180(7), 1138–1158 (2010)
5. Malyszko, D., Stepaniuk, J.: Adaptive Rough Entropy Clustering Algorithms in Image Segmentation. Fundamenta Informaticae 98(2-3), 199–231 (2010)
6. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV 2001, vol. (2), pp. 416–423. IEEE Computer Society, Los Alamitos (2001)
7. Pal, S.K., Shankar, B.U., Mitra, P.: Granular computing, rough entropy and object extraction. Pattern Recognition Letters 26(16), 2509–2517 (2005)
8. Pawlak, Z., Skowron, A.: Rough membership functions. In: Yager, R.R., Fedrizzi, M., Kacprzyk, J. (eds.) Advances in the Dempster-Shafer Theory of Evidence, pp. 251–271. John Wiley and Sons, New York (1994)
9. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007)
10. Pedrycz, W., Skowron, A., Kreinovich, V. (eds.): Handbook of Granular Computing. John Wiley & Sons, New York (2008)
11. Skowron, A., Stepaniuk, J.: Tolerance Approximation Spaces. Fundamenta Informaticae 27(2-3), 245–253 (1996)
12. Stepaniuk, J.: Rough–Granular Computing in Knowledge Discovery and Data Mining. Springer, Heidelberg (2008)
13. Slezak, D., Ziarko, W.: The investigation of the Bayesian rough set model. International Journal of Approximate Reasoning 40, 81–91 (2005)

Distance Based Fast Hierarchical Clustering Method for Large Datasets

Bidyut Kr. Patra, Neminath Hubballi, Santosh Biswas, and Sukumar Nandi

Department of Computer Science and Engineering,
Indian Institute of Technology Guwahati, Assam 781039, India
{bidyut,neminath,santosh_biswas,sukumar}@iitg.ernet.in

Abstract. Average-link (AL) is a distance based hierarchical clustering method, which is not sensitive to the noisy patterns. However, like all hierarchical clustering methods AL also needs to scan the dataset many times. AL has time and space complexity of $O(n^2)$, where n is the size of the dataset. These prohibit the use of AL for large datasets. In this paper, we have proposed a distance based hierarchical clustering method termed *l*-AL which speeds up the classical AL method in any metric (vector or non-vector) space. In this scheme, first leaders clustering method is applied to the dataset to derive a set of leaders and subsequently AL clustering is applied to the leaders. To speed-up the leaders clustering method, reduction in distance computations is also proposed in this paper. Experimental results confirm that the *l*-AL method is considerably faster than the classical AL method yet keeping clustering results at par with the classical AL method.

Keywords: distance based clustering, leaders clustering, average-link, large datasets.

1 Introduction

Clustering technique is required in numerous fields of engineering namely Data Mining, Pattern Recognition, Statistical Data Analysis, Bio-informatics, etc. [1,2]. Given a set of data points (called patterns), clustering involves grouping these patterns into different disjoint subsets termed as clusters based on some similarity measures. In other words, patterns in a cluster are more similar to each other than patterns in other clusters.

The clustering methods are mainly divided into two categories *viz.,* partitional clustering and hierarchical clustering, based on the way they produce the results. Partitional clustering methods create a single clustering (flat clustering) of the dataset. Partitional clustering can be classified into two classes based on the criteria used *viz.,* distance based and density based. Distance based methods optimize a global criteria based on the distance between the patterns. k-means, CLARA, CLARANS are examples of distance based clustering method. Density based methods optimize local criteria based on density information of the patterns. DBSCAN, DenClue are some well known density based clustering methods.

Hierarchical clustering methods create a sequence of nested clusterings of the dataset. Like partitional clustering, hierarchical clustering methods can also be classified in two classes *viz.,* density based (e.g., OPTICS [3], Chameleon) and distance based (e.g., single-link (SL) [4], complete-link (CL) [5], average-link (AL) [6]).

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 50–59, 2010.

The above three distance based hierarchical clustering methods namely SL, CL and AL differ in the "distance measure" between a pair of clusters. In SL (CL), distance between a pair of clusters C_1 and C_2 (say), is the distance between two closest (farthest) patterns one from C_1 and the other from C_2. In other words, only a pair of patterns decide the distance and it is independent of number of patterns present in the clusters. Therefore, SL and CL clustering methods are sensitive to outliers or noisy patterns. To minimize the effect of noisy patterns, inter-cluster distance in AL technique is computed using all patterns present in both clusters.

For some applications like network intrusion detection system (NIDS) proportion of the data points is unequal (i.e., number of the data points of abnormal/attack type is very less compared to normal data points). These low proportional data points (abnormal/attack data points) look like outliers in the feature space. These abnormal data points are likely to get merged with the clusters of normal data points in SL and CL methods as they are sensitive to the noisy (outlier) points. However, AL method works well even in the presence of noisy (outlier) data points. So, AL clustering method is more suitable for these type of applications.

AL needs to scan the dataset many times and has time and space complexity of $O(n^2)$. These prohibit use of AL for large datasets. In this paper, we have proposed a distance based hierarchical clustering method termed leader-average-link (*l*-AL) which speeds up the classical AL method. *l*-AL method is suitable for any metric space.

In the proposed scheme, we have used leaders clustering method to derive a set of leaders of the dataset. Later, AL is used to cluster the leaders. The final clustering is obtained by just replacing the leaders by their corresponding followers. *l*-AL has lower time complexity because AL is used only on the leaders which are much smaller in number compared to the dataset. Further, technique is proposed to reduce the number of distance computations in leaders clustering method.

The contributions of our paper are:

- Technique has been proposed to reduce the number of distance calculations required in leaders clustering. Triangle inequality property of metric space has been used for this reduction.
- A distance based hierarchical clustering method termed *l*-AL is proposed which speeds up the classical AL method and scans the dataset once. *l*-AL uses the accelerated leader clustering technique to generate the leaders.
- *l*-AL does not use any vector space properties[1]. It utilizes only the distance information between the data points. Therefore, *l*-AL method is suitable for vector as well as non-vector metric space.

Rest of the paper is organized as follows. Section 2 describes a summary of related works. Section 3 describes the brief background of the proposed clustering method. Section 4 describes the proposed *l*-AL method and also a relationship between the AL method and the *l*-AL method is formally established. Experimental results and conclusion are discussed in Section 5 and Section 6, respectively.

[1] Vector addition and scalar multiplication.

2 Related Work

In this section, a brief review of related works is reported for distance based hierarchical clustering methods.

T. Zhang et al. in [7] introduced a clustering method called BIRCH for large datasets. The core concept of BIRCH is *Clustering Feature (CF)*. The *CF* utilizes the vector space (Euclidean space) properties to store the summary of k data points $\{\overrightarrow{X_i}\}_{i=1..k}$. The *CF* is defined as $CF = (k, \sum_{i=1}^{k} \overrightarrow{X_i}, \sum_{i=1}^{k} \overrightarrow{X_i}^2)$. One can easily compute average intra-cluster and inter-cluster distances from the CF values. However, in many applications, datasets are from non-vector metric space. Therefore, BIRCH method is not suitable for those applications.

Dash et al. in [8] proposed a fast hierarchical clustering method based on the partially overlapping partitioning (POP). First, dataset is partitioned into a number of overlapping cells and these are progressively merged into a numbers of clusters. Next, traditional hierarchical agglomerative clustering (centroid based) method is applied.

Nanni et al. in [9] exploited the triangle inequality property of the distance metric to speed-up the hierarchical clustering methods (SL and CL).

Recently, Koga et al. [10] proposed a fast approximation algorithm for SL method. Unlike classical SL method it quickly finds close clusters in linear time using a probabilistic approach.

These methods successfully speedup the traditional clustering methods. However, these methods are not suitable either for large datasets (entire dataset in main memory of machine) or categorical datasets (non-vector space). The proposed *l*-AL method speeds up the exiting AL clustering method but needs to store only leaders in the main memory of the machine (as AL is applied only on the leaders) and uses only the distance information. So, for large datasets *l*-AL method is more suitable instead of classical AL.

3 Background of the Proposed Method

As already discussed, the proposed *l*-AL builds on two clustering methods *viz.,* leaders clustering and average-link clustering method; they are discussed in this section.

Leaders clustering method. Leaders clustering method [1] is a distance based single scan partitional clustering method. Recently, leaders clustering method has been used in preclustering phase of many data mining applications [11,12]. For a given threshold distance τ, it produces a set of leaders $\mathcal{L}$ incrementally. For each pattern x, if there is a leader $l \in \mathcal{L}$ such that $||x - l|| <= \tau$, then x is assigned to the cluster represented by l; otherwise, x becomes a new leader. The time complexity of the leaders clustering is $O(mn)$, where $m = |\mathcal{L}|$. The space complexity is $O(m)$, if only leaders are stored; otherwise $O(n)$. However, it can only find convex shaped clusters.

Average-link clustering method. Average-link [6,13] is a distance based agglomerative hierarchical clustering method. In average-link, distance between two clusters C_1 and C_2 is the average of distances between all pairs in $C_1 \times C_2$. That is,
$Distance(C_i, C_j) = \frac{1}{|C_i|*|C_j|} \sum_{x_i \in C_i} \sum_{x_j \in C_j} ||x_i - x_j||$

Algorithm 1. AL($\mathcal{D}$,h)

Place each pattern $x \in \mathcal{D}$ in a separate cluster. This is the initial clustering $\pi_1 = \{C_1, C_2, .. \ldots, C_n\}$ of $\mathcal{D}$. Compute the inter-cluster distance matrix and set $i = 1$.
while There is a pair of clusters $C_x, C_y \in \pi_i$ such that *Distance*$(C_x, C_y) \leq h$ **do**
  Select two closest clusters C_l and C_m and merge into a single new cluster $C = C_l \cup C_m$.
  Next clustering is $\pi_{i+1} = \pi_i \cup \{C\} \setminus \{C_l, C_m\}$; $\quad i = i + 1$
  Update the distances from C to all other clusters in the current clustering π_i.
end while
Output all clustering $\pi_1, \pi_2, \ldots, \pi_p$.

The average-link method with inter-cluster distance (h) is depicted in Algorithm 1. The AL method is not sensitive to noisy patterns. The time and space complexity of the AL method are $O(n^2)$ [6,13]. It scans the dataset many times. Therefore, AL method is not suitable for large datasets.

4 Proposed Clustering Method

To overcome the deficiencies of the AL method, we propose a clustering method termed as *l*-AL, which is the combination of leaders and average-link. In this section, we first discuss the technique to speed up the leaders clustering followed by the proposed *l*-AL scheme.

4.1 Accelerating Leader Clustering Method

We use triangle inequality property to reduce the number of distance computations of the leaders clustering method. We term this approach as *Accelerated leader*. In recent years, triangle inequality property of the metric space has been used to reduce the distance computations in the clustering methods [14,15]. The triangle inequality property can be stated as follows.

$$\forall a, b, c \in \mathcal{D}, d(a, b) \leq d(b, c) + d(a, c) \tag{1}$$

where $\mathcal{D}$ is the set of data points, d is a distance function over the metric space $M = (\mathcal{D}, d)$. Let l_1, l_2 be the two leaders and x be an arbitrary pattern of the dataset. Form equation (1),

$$d(x, l_2) \geq |d(l_1, l_2) - d(x, l_1)| \tag{2}$$

From equation(2) it may be noted that a lower bound on the distance between leader l_2 and pattern x (termed as $d^{lower}(x, l_2)$) can be obtained from $d(l_1, x)$ and $d(l_1, l_2)$ without calculating the exact distance between l_2 and x.

Accelerated leader works as follows. It computes a distance matrix for leaders. This distance matrix can be generated hand-in-hand during the generation of leaders (without any extra distance computation). Therefore, one can easily estimate $d^{lower}(x, l_2)$ only by computing distance $d(l_1, x)$.

Let τ be the leader's threshold. Let $\mathcal{L} = \{l_1, l_2, \ldots, l_k\}$ be the set of leaders generated at an instant and all be marked as "unprocessed" leaders. The scheme starts with

calculating the distance between a new pattern x and leader l_f (where l_f is the earliest generated leader among the set of "unprocessed" leaders). If $d(x, l_f) \leq \tau$, then x becomes the follower of leader l_f. If $d(x, l_f) > \tau$, we can avoid the distance computations from all leaders $l_i \in L - \{l_f\}$ where estimated lower bound $d^{lower}(x, l_i) > \tau$. Leaders l_i, l_f are marked as "processed" (pruned) leaders. If all leaders are pruned then x becomes a new leader and added to $\mathcal{L}$. If all leaders are not marked as "processed", we repeat same procedure of calculating distance between x with next unprocessed leader $l_u \in \mathcal{L}$ if $d(x, l_u) < d(x, l_f)$. If no (unprocessed) $l_u \in \mathcal{L}$ is found such that $d(x, l_u) > d(x, l_f)$, then there cannot be a leader l_j such that $d(x, l_j) \leq \tau$; so x becomes a new leader and added to $\mathcal{L}$. The whole procedure of *Accelerated leaders* is depicted in Algorithm 2.

Algorithm 2. Accelerated leader($\mathcal{D}$, τ)

```
1: L ← {l_1}; { Let l_1 ∈ D be the first scanned pattern}
2: for each x ∈ D \ l_1 do
3:    S ← L;   MIN = ∞;
4:    while (x does not become a follower and S is not empty) do
5:       Pick a leader l_i and delete from S. //* l_i is earliest generated leader among the leaders
         in S *//
6:       if d(x, l_i) ≤ τ then
7:          x becomes a follower of l_i; break;
8:       else if d(x, l_i) < MIN then
9:          MIN = d(x, l_i);
10:         for each leader l_k ∈ S(l_k ≠ l_i) do
11:            if d^lower(x, l_k) > τ then
12:               delete l_k from set S.
13:            end if
14:         end for
15:      end if
16:   end while
17:   if (x not be follower of any exsisting leaders in L) then
18:      x becomes new leader and added to L.
19:   end if
20: end for
21: Output L* = {(l, followers(l)) | l ∈ L}.
```

4.2 Leader-Average-Link(*l*-AL) Method

In this sub-section, we discuss the proposed l-AL scheme. The l-AL method works as follows. First, a set of leaders ($\mathcal{L}$) is obtained applying the Accelerated leaders clustering method to the dataset (as discussed in previous subsection). Next, these leaders are clustered using classical AL method with minimum inter-cluster distance h. Finally, each leader is replaced by its followers set to produce the final sequence of clustering. The l-AL method is depicted in Algorithm 3. The time and space complexity of the proposed method are analyzed as follows.

Algorithm 3. l-AL($\mathcal{D}$, τ, h)

Apply Accelerated leader($\mathcal{D}$, τ) as given in Algorithm 2. Let the set of leaders be $\mathcal{L}$.
Apply AL($\mathcal{L}, h$) as given in Algorithm 1. Let output be $\pi_1^{\mathcal{L}}, \pi_2^{\mathcal{L}}, \ldots, \pi_k^{\mathcal{L}}$ { A sequence of clustering of leaderset}
Each leader in clustering $\pi_i^{\mathcal{L}}$ is replaced by its followers set. This gives a sequence of clustering of the dataset (say $\pi_1^{\mathcal{D}}, \pi_2^{\mathcal{D}}, \ldots, \pi_k^{\mathcal{D}}$).
Output $\pi_1^{\mathcal{D}}, \pi_2^{\mathcal{D}}, \ldots, \pi_k^{\mathcal{D}}$.

1. The step of obtaining set of all leaders $\mathcal{L}$ takes time of $O(mn)$, where m is the size of the leader set. The space complexity is $O(m)$. It scans the dataset once.
2. The time complexity of the AL($\mathcal{L}$, h) is $O(m^2)$. The space complexity is $O(m^2)$.

The overall running time of l-AL is $O(mn + m^2) = O(mn)$. Experimentally, we also show that l-AL is considerably faster than that of the classical AL method, since AL works with the whole dataset, whereas the l-AL works with set of leaders . The space complexity of the l-AL method is $O(m^2)$.

Relationship between AL and *l*-AL methods. As discussed in previous sub-section l-AL generates clustering of dataset at a computational cost significantly lower than classical AL. It may be noted that l-AL may overestimate or underestimate the distance between a pair of clusters with compared to the classical AL method (termed as distance error). This may lead to deviation in clustering results obtained by l-AL compared to AL. In this subsection a theoretical upper bound of the distance error is established.

Let $l_1, l_2 \in \mathcal{L}$ be two leaders obtained using the threshold τ. Let $F(l_1) \subseteq \mathcal{D}$ be the set of followers of leader l_1 including l_1. Similarly, $F(l_2)$ is the set of followers of l_2.

Lemma 1. *If the leaders threshold is τ, then l-AL may introduce an error of average value $Er(l_1, l_2) < 2\tau$, while measuring the distance between a pair of leaders (l_1, l_2).*

Proof: Let $||l_1 - l_2|| = T > 2\tau$. We have three cases.

1. We assume that all followers of l_1 are more than T distance away from the followers of l_2, except the leaders themselves. (This case is illustrated in Fig. 1(a)). Formally, $||x_i - x_j|| > T$ where $x_i \in F(l_1) \setminus \{l_1\}$ and $x_j \in F(l_2) \setminus \{l_2\}$. Therefore, distance between a pair of followers (x_i, x_j) can be at most $T + 2\tau$. So, for all followers (of this case) l-AL underestimates the distance and approximates to T (as $||l_1 - l_2|| = T$). Therefore, error incurred by a pair of such followers is at most 2τ. The average error $Er(l_1, l_2)$ introduced by the l-AL method for a pair of leader can be computed as follows.
$Er(l_1, l_2) = \frac{(m_1-1)(m_2-1)2\tau+(m_1-1)\tau+(m_2-1)\tau}{m_1 m_2} \leq \frac{m_1 m_2 * 2\tau}{m_1 m_2} = 2\tau$
where $m_1 = |F(l_1)|$ and $m_2 = |F(l_2)|$.
The first term of the numerator in above equation appears due to errors introduced by the followers of l_1 ($m_1 - 1$ in number) and l_2 ($m_2 - 1$ in number). Second (third) term captures errors introduced by $l_2(l_1)$ and followers of $l_1(l_2)$.

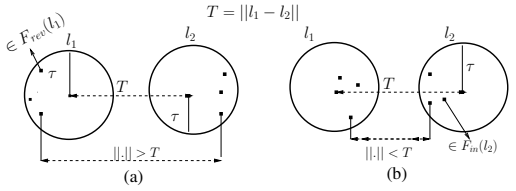


Fig. 1. (a) l-AL underestimates the distance (b) l-AL overestimates the distance

2. We assume that $||x_i - x_j|| < T$ such that $x_i \in F(l_1) \setminus \{l_1\}$ and $x_j \in F(l_2) \setminus \{l_2\}$, distance between x_i, x_j cannot be less than $T - 2\tau$. Similar to case 1 we obtain the average error $Er(l_1, l_2) < 2\tau$. (Fig. 1(b)). Here, l-AL overestimates the distance.
3. If distance between any pair of followers is $||x_i - x_j|| = (T - 2\tau, T + 2\tau)$, the average error is less than 2τ.

From all three cases, we obtain that average error $Er(l_1, l_2)$ is less than 2τ □

The distance error computation between two leaders can easily be extended for a pair of clusters, as follows.

Theorem 1. *If the leaders threshold is τ, then l-AL may introduce an average error $Er(C_1, C_2) < 2\tau$ in measuring the distance between a pair of clusters (C_1, C_2) .*

Proof: From Lemma 1, we know that average error between a pair of leaders $Er(l_1, l_2) < 2\tau$. Let the upper bound on the average error $Er(l_1, l_2)$ be $2\tau - \epsilon$, where $0 < \epsilon << \tau$. Then the average error between a pair of clusters (C_1, C_2) is as follows.

$Er(C_1, C_2) = \frac{(2\tau - \epsilon) * m_1^l m_2^l}{m_1^l m_2^l} = 2\tau - \epsilon < 2\tau,$

where m_1^l and m_2^l are the numbers of leaders of the clusters C_1 and C_2, respectively. □

For large datasets, numbers of leaders are considerably less compared to the size of the data. Number of followers per leader are considerably large. As a result, there is high probability that followers of leader are distributed evenly. This leads to error in distance computation between leaders by l-AL method is marginal, which is also reflected in our experimental results. So, Corollary 1 can be deduced.

Corollary 1. *If the followers of leaders are distributed uniformly, the average distance error for those leaders is* 0. □

5 Experimental Results

In this section, we discuss the experimental evaluation of our proposed clustering method. We conducted the experiments with synthetic and real world datasets (Table 1) (UCI Repository) after removing the class labels. We implemented leaders clustering and *Accelerated leader* using C language and executed on Intel Xeon Processor $(3.6GHz)$ with 8GB RAM IBM Workstation. These two methods are tested with Circle and Shuttle datasets. The detailed results are shown in Table 2 and Fig 2. Proposed *Accelerated leader* performs significantly less computations compared to that of classical leaders method (Table 2). For example with the Circle dataset and $\tau = 0.1$, *Accelerated leader*

Table 1. Datasets Used

Dataset	# Pattern	# Features
Circle (Synthetic)	28000	2
Gaussian(Synthetic)	4078	2
Pendigits	7494	16
Letter	20000	16
Shuttle	58000	9

Table 2. Performance of Accelerated leaders for Circle dataset

Threshold (τ)	Method	# Computations (in Million)
0.1	Leaders	90.13
	Accelerated leader	**28.66**
0.2	Leaders	20.03
	Accelerated leader	**2.37**
0.3	Leaders	11.33
	Accelerated leader	**0.96**
0.4	Leaders	5.97
	Accelerated leader	**0.49**
0.5	Leaders	3.85
	Accelerated leader	**0.35**
0.6	Leaders	2.81
	Accelerated leader	**0.30**

computes 60 millions less distance calculations to achieve same results as that of the classical leaders method (Table 2). To show the performance of the proposed leaders clustering speeding-up technique with variable dataset size, experiments are conducted on Shuttle dataset with leaders threshold $\tau = 0.001$. This is reported in Fig 2. It may be noted that with increase of the dataset size, number of distance calculations reduces significantly compared to classical leaders.

Performance of *l-AL* method. To show the performance of the l-AL method, we implemented AL and l-AL methods using C language and executed on Intel Xeon Processor ($3.6GHz$) with 8GB RAM IBM Workstation. We computed the Rand Index (RI)([16]) between the final clustering results of the l-AL and the AL method. We conducted experiments with synthetic (Gaussian) (Fig. 3) as well as real world large datasets. Detailed results are provided in Table 3, Table 4 and Table 5. Gaussian is a 2 dimensional data with four clusters. Three clusters are drawn from the normal distribution with means $((0\ 0)^T, (0\ 8)^T, (7\ 7)^T)$ and covarience matrix I_2 (Identity Matrix of size 2.) Fourth cluster is drawn from a uniform random distribution (Fig. 3). For this dataset, leaders thresholds τ were chosen as $0.25, 0.50$. The clustering results of the proposed l-AL method are same as the classical AL method with cut-off distances (h) $5.0, 7.0, 8.0$

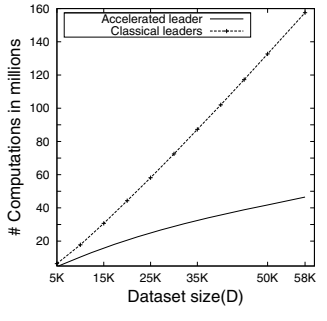


Fig. 2. Number of distance computations Vs dataset size for Shuttle data ($\tau = 0.001$)

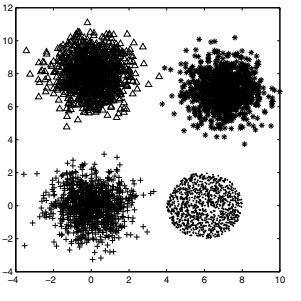


Fig. 3. Gaussian (Synthetic) Dataset

Table 3. Results for Gaussian Dataset

Threshold (τ)	Cut-off (h)	Method	Time (Sec.)	Rand Index (RI)
0.25	4.0	***l*-AL**	**0.04**	**0.999**
	4.0	AL	14.96	–
	4.5	***l*-AL**	**0.04**	**0.999**
	4.5	AL	14.96	—
	5.0	***l*-AL**	**0.04**	**1.000**
	5.0	AL	14.96	—
0.50	6.5	***l*-AL**	**0.01**	**0.870**
	6.5	AL	14.96	—
	7.0	***l*-AL**	**0.01**	**1.000**
	7.0	AL	14.96	—
	8.0	***l*-AL**	**0.01**	**1.000**
	8.0	AL	14.96	—

and results are very close to AL method for the cut-off distances (h) $4.0, 4.5, 6.5$ (Table 3). The execution time of the proposed method is less than 0.3% of AL method. To show the effectiveness of the proposed method in the real world large datasets, we experimented with Pendigits, Letter and Shuttle datasets (Table 1). For Pendigits dataset, clustering results of l-AL method is very close ($RI = 0.899, 0.897, 0.911, 0.904, 0.935, 0.933, 0.913, 0.909$) to that of the classical AL method with different τ $(30, 40)$ and different h $(145, 150, 155, 160)$ (Table 4). The l-AL consumes less than 0.5% of CPU time compared to the AL method.

For **Letter dataset**, with $\tau = 4$ and different cut-off distances ($h = 10, 12, 15$) l-AL method produces clustering results ($RI = 0.811, 0.835, 0.977$) close to that of the classical AL method (Table 5). However, l-AL is more than 400 times faster than that of the classical AL method. For **Shuttle dataset**, we executed AL and l-AL methods and results are reported in Table 5. It is noted that clustering results $(RI = 0.999, 1.000)$ are at par or same with the AL method at $\tau = 0.001$ and $h = 0.8, 0.9, 1.0, 1.2$.

Table 4. Results for Pendigits data

Dataset	Threshold (τ)	Cut-off (h)	Method	Time (Sec.)	Rand Index (RI)
Pendigits	30	145	***l*-AL**	**1.13**	**0.899**
		145	AL	201.55	–
		150	***l*-AL**	**1.13**	**0.897**
		150	AL	201.55	—
		155	***l*-AL**	**1.13**	**0.911**
		155	AL	201.55	—
		160	***l*-AL**	**1.13**	**0.904**
		160	AL	201.55	—
	40	145	***l*-AL**	**0.31**	**0.935**
		145	AL	201.55	—
		150	***l*-AL**	**0.31**	**0.933**
		150	AL	201.55	—
		155	***l*-AL**	**0.31**	**0.913**
		155	AL	201.55	—
		160	***l*-AL**	**0.31**	**0.909**
		160	AL	201.55	—

Table 5. Results for Large Real Datasets

Dataset	Threshold (τ)	Cut-off (h)	Method	Time (Sec.)	Rand Index (RI)
Letter	4	10	***l*-AL**	**3.28**	**0.811**
		10	AL	1464.10	—
		12	***l*-AL**	**3.28**	**0.835**
		12	AL	1464.10	—
		15	***l*-AL**	**3.28**	**0.977**
		15	AL	1464.10	—
Shuttle	0.001	0.8	***l*-AL**	**55.55**	**0.999**
		0.8	AL	7140.54	—
		0.9	***l*-AL**	**55.55**	**0.999**
		0.9	AL	7140.54	—
		1.0	***l*-AL**	**55.55**	**0.999**
		1.0	AL	7140.54	—
		1.2	***l*-AL**	**55.55**	**1.000**
		1.2	AL	7140.54	—

6 Conclusions

In this paper, we proposed a clustering method *l*-AL for the large dataset in any metric space. We first apply leaders clustering to derive a set of prototypes of the dataset and subsequently AL method is applied. A technique to reduce the number of distance computations in the leaders method is also proposed. The clustering results produced by the *l*-AL method are at par with that of the AL method. The *l*-AL method takes significantly less time compared to that of the AL method. Like AL, *l*-AL is immune to clustering of data with noise. As *l*-AL is faster, it can be used in application like network intrusion detection system where data size is very large and spurious patterns are very less.

Acknowledgement. Bidyut Kr. Patra is supported by CSIR, New Delhi, Govt. of India.

References

1. Hartigan, J.A.: Clustering Algorithms. John Wiley & Sons, Inc., New York (1975)
2. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Computing Surveys 31(3), 264–323 (1999)
3. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: Ordering points to identify the clustering structure. In: Proceedings ACM SIGMOD, pp. 49–60 (1999)
4. Sneath, A., Sokal, P.H.: Numerical Taxonomy. Freeman, London (1973)
5. King, B.: Step-Wise Clustering Procedures. Journal of the American Statistical Association 62(317), 86–101 (1967)
6. Murtagh, F.: Complexities of hierarchic clustering algorithms: state of the art. Computational Statistics Quarterly 1, 101–113 (1984)
7. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: An Efficient Data Clustering Method for Very Large Databases. In: Proceedings of the 1996 ACM SIGMOD, pp. 103–114 (1996)
8. Dash, M., Liu, H., Scheuermann, P., Tan, K.L.: Fast hierarchical clustering and its validation. Data Knowl. Eng. 44(1), 109–138 (2003)
9. Nanni, M.: Speeding-up hierarchical agglomerative clustering in presence of expensive metrics. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 378–387. Springer, Heidelberg (2005)
10. Koga, H., Ishibashi, T., Watanabe, T.: Fast agglomerative hierarchical clustering algorithm using Locality-Sensitive Hashing. Knowledge and Information Systems 12(1), 25–53 (2007)
11. Viswanath, P., Babu, V.: Rough-dbscan: a fast hybrid density based clustering method for latge data sets. Pattern Recognition Letters 30(16), 1477–1488 (2009)
12. Patra, B.K., Nandi, S.: A Fast Single Link Clustering Method Based on Tolerance Rough Set Model. In: Sakai, H., et al. (eds.) RSFDGrC 2009. LNCS (LNAI), vol. 5908, pp. 414–422. Springer, Heidelberg (2009)
13. Olson, C.F.: Parallel algorithms for hierarchical clustering. Parallel Computing 21, 1313–1325 (1995)
14. Elkan, C.: Using the triangle inequality to accelerate k-means. In: ICML, pp. 147–153 (2003)
15. Nassar, S., Sander, J., Cheng, C.: Incremental and effective data summarization for dynamic hierarchical clustering. In: Proceedings of SIGMOD Conference, pp. 467–478 (2004)
16. Rand, W.M.: Objective Criteria for Evaluation of Clustering Methods. Journal of American Statistical Association 66(336), 846–850 (1971)

TI-DBSCAN: Clustering with DBSCAN by Means of the Triangle Inequality

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. Grouping data into meaningful clusters is an important data mining task. DBSCAN is recognized as a high quality density-based algorithm for clustering data. It enables both the determination of clusters of any shape and the identification of noise in data. The most time-consuming operation in DBSCAN is the calculation of a neighborhood for each data point. In order to speed up this operation in DBSCAN, the neighborhood calculation is expected to be supported by spatial access methods. DBSCAN, nevertheless, is not efficient in the case of high dimensional data. In this paper, we propose a new efficient TI-DBSCAN algorithm and its variant TI-DBSCAN-REF that apply the same clustering methodology as DBSCAN. Unlike DBSCAN, TI-DBSCAN and TI-DBSCAN-REF do not use spatial indices; instead they use the triangle inequality property to quickly reduce the neighborhood search space. The experimental results prove that the new algorithms are up to three orders of magnitude faster than DBSCAN, and efficiently cluster both low and high dimensional data.

1 Introduction

Grouping data into meaningful clusters is an important data mining task. The quality of clustering depends on a used algorithm. The DBSCAN algorithm (Density-Based Spatial Clustering of Applications with Noise) [3] is recognized as a high quality scalable algorithm for clustering low dimensional data. The most time-consuming operation in DBSCAN is the calculation of a neighborhood for each data point. In order to speed up this operation in DBSCAN, it is expected to be supported by spatial access methods such as R*-tree [1] (R-tree [4]). DBSCAN, nevertheless, is not able to cluster high dimensional data efficiently. A method for improving the performance of DBSCAN based on early removal of core points has been offered in [6]. There, the carried out experiments showed that using the proposed method speeded up DBSCAN's performance by 50%.

In this paper, we propose a new efficient TI-DBSCAN algorithm and its variant TI-DBSCAN-REF that apply the same clustering methodology as DBSCAN. Unlike DBSCAN, TI-DBSCAN and TI-DBSCAN-REF do not use spatial indices; instead they use the triangle inequality property to quickly reduce the neighborhood search space. To the best of our knowledge, our proposal is the first one that relates to a

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 60–69, 2010.

density-based clustering; the other trials of using the triangle inequality in clustering were related to *k*-means algorithm [2][7] and hierarchical algorithms following the results presented in [7].

The paper has the following layout. Section 2 recalls the notion of a cluster and noise according to [3]. In Section 3, we a offer theoretical basis of our approach to optimizing DBSCAN-like clustering. In Section 4, we propose the TI-DBSCAN algorithm and its modification TI-DBSCAN-REF. Section 5 reports the performance of TI-DBSCAN, TI-DBSCAN-REF as well as the performance of DBSCAN. Section 6 concludes the obtained results.

2 Basic Notions

In the context of the DBSCAN algorithm [3], a cluster is an area of high density. Data points in a low density area constitute noise. A point in space is considered a member of a cluster if there is a sufficient number of points within a given distance. In the sequel, the distance between two points p and q will be denoted by distance(p,q). Please, note that one may use a variety of distance metrics. Depending on an application, one metric may be more suitable than the other. In particular, if Euclidean distance is used, a neighborhood of a point has a spherical shape; when Manhattan distance is used, the shape is rectangular. For simplicity of the presentation, in our examples we will refer to Euclidean distance, although our approach is suitable for any distance metric. Below, we recall definitions of a density based cluster and related notions after [3].

Eps-neighborhood of a point p (denoted by $N_{Eps}(p)$) is defined as the set of points q in dataset D that are distant from p by no more than Eps; that is, $N_{Eps}(p) = \{q \in D \mid distance(p,q) \leq Eps\}$.

A point p is defined as a *core point* if its *Eps-neighborhood* contains at least MinPts points; that is, if $|N_{Eps}(p)| \geq MinPts$.

A point p is defined as *directly density-reachable* from a point q with respect to Eps and MinPts if $p \in N_{Eps}(q)$ and q is a core point.

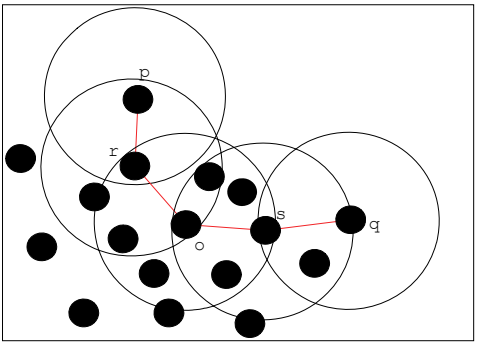


Fig. 1. Density-reachability of points (MinPts = 6)

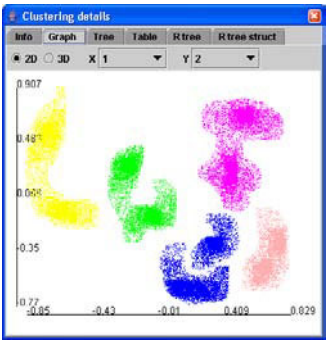


Fig. 2. Sample result of clustering with DBSCAN

A point p is defined as *density-reachable* from a point q with respect to Eps and MinPts if there is a sequence of points p_1, …, p_n such that $p_1 = q$, $p_n = p$ and p_{i+1} is directly density-reachable from p_i, i = 1..n-1.

Example 1. Let MinPts = 6. Point r in Figure 1 has 6 neighbors (including itself) in its neighborhood $N_{Eps}(r)$, so it is a core point. Point p has 2 neighbors in $N_{Eps}(p)$, so it is not a core point. Point p, however, belongs to $N_{Eps}(r)$, so it is directly density-reachable from r. To the contrary r is not directly density-reachable from p despite r belongs to $N_{Eps}(p)$. Point p in Figure 1 is density-reachable from point o, since there is a point (e.g. point r) such that p is directly density-reachable from it and it is directly density-reachable from o. Please note that p, which is density-reachable from core point o, is not a core point. □

A point p is defined as a *border point* if it is not a core point, but is density-reachable from some core point. Hence, a point is a border one if it is not a core point, but belongs to the *Eps-neighborhood* of some core point.

Let C(o) determine all points in D that are density-reachable from point o. Clearly, if o is not a core point, then C(o) is empty. In Figure 1, points p and q are density-reachable from core point o. Hence, p and q belong to C(o).

A *cluster*[1] is defined as a non-empty set of all points in D that are density-reachable from a core point. Hence, each C(p) is a cluster provided p is a core point. Interestingly, if p and q are core points belonging to the same cluster, then C(p) = C(q); that is, both points determine the same cluster [3]. Thus, a core point p belongs to exactly one cluster, namely to C(p). We note, however, that a border point may belong to more than one cluster.

Noise is defined as the set of all points in D that do not belong to any cluster; that is, the set of all points in D that are not density-reachable from any core point. Hence, each point that is neither a core point, nor border one, constitutes noise.

Fig. 2 presents the results of clustering with DBSCAN for a sample dataset.

3 Using the Triangle Inequality for Efficient Determination of Eps-Neighborhoods

Let us start with recalling the triangle inequality property:

Property 1. (Triangle inequality property). For any three points p, q, r:

$$distance(p,r) \leq distance(p,q) + distance(q,r)$$

Property 2 presents its equivalent form, which is more suitable for further considerations.

Property 2. (Triangle inequality property). For any three points p, q, r:

$$distance(p,q) \geq distance(p,r) - distance(q,r).$$

[1] This definition differs from the original one provided in [3]. However, it is equivalent to the original one by Lemma 1 in [3], and is more suitable for our presentation.

Lemma 1. Let D be a set of points. For any two points p, q in D and any point r:

$$\text{distance}(p,r) - \text{distance}(q,r) > \text{Eps} \Rightarrow q \notin N_{\text{Eps}}(p) \wedge p \notin N_{\text{Eps}}(q).$$

Proof. Let distance(p,r) − distance(q,r) > Eps (*). By Property 2, distance(p,q) ≥ distance(p,r) − distance(q,r) (**). By (*) and (**), distance(p,q) > Eps, and distance(q,p) = distance(p,q). Hence, $q \notin N_{\text{Eps}}(p)$ and $p \notin N_{\text{Eps}}(q)$. □

By Lemma 1, if we know that the difference of distances of two points p and q to some point r is greater than Eps, we are able to conclude that $q \notin N_{\text{Eps}}(p)$ without calculating the actual distance between p and q. Theorem 1 is our proposal of effective determination of points that do not belong to Eps-neighborhood of a given point p.

Theorem 1. Let r be any point and D be a set of points ordered in a non-decreasing way with respect to their distances to r. Let p be any point in D, q_f be a point following point p in D such that distance(q_f,r) − distance(p,r) > Eps, and q_b be a point preceding point p in D such that distance(p,r) − distance(q_b,r) > Eps. Then:
a) q_f and all points following q_f in D do not belong to $N_{\text{Eps}}(p)$.
b) q_b and all points preceding q_b in D do not belong to $N_{\text{Eps}}(p)$.

Proof. Let r be any point and D be a set of points ordered in a non-decreasing way with respect to their distances to r.
a) Let p be any point in D, q_f be a point following point p in D such that distance(q_f,r) − distance(p,r) > Eps (*), and s be either point q_f or any point following q_f in D. Then distance(s,r) ≥ distance(q_f,r) (**). By (*) and (**), distance(s,r) − distance(p,r) > Eps. Thus, by Lemma 1, $s \notin N_{\text{Eps}}(p)$.
b) The proof is analogous to the proof of Theorem 1a). □

Corollary 1. Let r be any point and D be a set of points ordered in a non-decreasing way with respect to their distances to r. Let p be any point in D, q_f be the first point following point p in D such that distance(q_f,r) − distance(p,r) > Eps, and q_b be the first point preceding point p in D such that distance(p,r) − distance(q_b,r) > Eps. Then, only the points that follow q_b in D and precede q_f in D have a chance to belong to $N_{\text{Eps}}(p)$, and p certainly belongs to $N_{\text{Eps}}(p)$.

Example 2. Let r be a point (0,0). Figure 3 shows sample set D of two dimensional points. Table 1 illustrates the same set D ordered in a non-decreasing way with respect to the distance of its points to point r. Let us consider the determination of the Eps-neighborhood of point p = F, where Eps = 0.5, by means of Corollary 1. We note that distance(F,r) = 3.2, the first point q_f following point F in D such that distance(q_f,r) − distance(F,r) > Eps is point C (distance(C,r) − distance(F,r) = 4.5 − 3.2 = 1.3 > Eps), and the first point q_b preceding point p in D such that distance(F,r) − distance(q_b,r) > Eps is G (distance(F,r) − distance(G,r) = 3.2 − 2.4 = 0.8 > Eps). By Corollary 1, only the points that follow G and precede C in D (here, points F and H) may belong to $N_{\text{Eps}}(F)$. Clearly, $F \in N_{\text{Eps}}(F)$. Hence, H is the only point for which it is necessary to calculate its actual distance to F in order to determine $N_{\text{Eps}}(F)$ properly. □

In the sequel, a point r to which the distances of all points in D have been determined will be called a *reference point*.

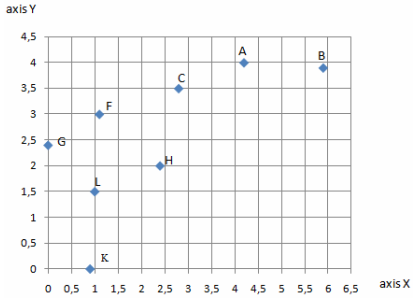


Fig. 3. Set of points D

Table. 1. Ordered set of points D from Fig. 3 with their distance to reference point r(0,0)

Q	X	Y	distance(q,r)
K	0,9	0,0	0,9
L	1,0	1,5	1,8
G	0,0	2,4	2,4
H	2,4	2,0	3,1
F	1,1	3,0	3,2
C	2,8	3,5	4,5
A	4,2	4,0	5,8
B	5,9	3,9	7,1

4 New Algorithms: TI-DBSCAN and TI-DBSCAN-REF

In this section, we propose a new clustering algorithm called TI-DBSCAN and its version TI-DBSCAN-REF. The result of clustering of core points and identifying of noise by our algorithms is the same as the one produced by the DBSCAN algorithm [3]. The border points may be assigned to different clusters[2]. In our algorithms we use Corollary 1 for efficient determination of the Eps-neighborhoods of points. In addition, we adopt the solution from [6] that consists in removing a point from the analyzed set D as soon as it is found to be a core point. Here, we remove each analyzed point, even if it not a core point. Let us start with the description of TI-DBSCAN.

Notation for TI-DBSCAN

- D – the set of points that is subject to clustering;
- Eps – the radius of the point neighborhood;
- MinPts – the required minimal number of points MinPts within Eps-neighborhood;
- r – a reference point assumed to be fixed, e.g. to the point with all coordinates equal to 0 or minimal values in the domains of all coordinates;
- fields of any point p in D:
  - p.ClusterId – label of a cluster to which p belongs; initially assigned the UNCLASSIFIED label;
  - p.dist – the distance of point p to reference point r;
  - p.NeighborsNo – the number of neighbors of p already found; initially assigned 1 to indicate that a point itself belongs to its own Eps-neighborhood;
  - Border – the information about neighbors of p that turned out non-core points for which it is not clear temporary if they are noise ones or border ones; initially assigned an empty set;
- D' – the result of clustering of D (initially an empty set);

[2] Although a border point may belong to many clusters, DBSCAN assigns it arbitrarily only to one of them, so does TI-DBSCAN. It is easy to modify the algorithms so that border points are assigned to all including clusters.

TI-DBSCAN takes as an input a set of points D, a radius Eps and a threshold MinPts. Its output, i.e. clustered points, shall be stored in D'. A reference point r is assumed to be fixed, e.g. to the point with all coordinates equal to 0 or minimal values in the domains of all coordinates. With each point p in D, there are associated the following fields: ClusterId, dist, NeighborsNo, and Border.

TI-DBSCAN starts with the initialization of D' and the fields of all points in D. Then it sorts all points in D in a non-decreasing way w.r.t. their distance to reference point r. Next it generates a label for the first cluster to be found. Then it scans point after point in D. For each scanned point p, the TI-ExpandCluster function (described later) is called. If p is a core point, then the function assigns the current cluster's label to all points in C(p), moves them from D to D', and TI-DBSCAN generates a next label for a new cluster to be created. Otherwise, point p is assigned label NOISE and is moved from D to D'. After all points in D are scanned, each point is assigned either to a respective cluster identified by ClusterId or is identified as noise.

```
Algorithm TI-DBSCAN(set of points D, Eps, MinPts);
/* assert: r denotes a reference point */
D' = empty set of points;
for each point p in set D do
    p.ClusterId = UNCLASSIFIED;
    p.dist = Distance(p,r); p.NeighborsNo = 1; p.Border = ∅
endfor
sort all points in D non-decreasingly w.r.t. field dist;
ClusterId = label of first cluster;
for each point p in the ordered set D starting from
    the first point until last point in D do
     if TI-ExpandCluster(D, D', p, ClusterId, Eps, MinPts) then
         ClusterId = NextId(ClusterId)
     endif
endfor
return D'                              // D' is a clustered set of points
```

The TI-ExpandCluster function starts with calling TI-Neighborhood function (decribed later) to determine Eps-neighborhood of a given point p in, possibly reduced, set D (more precisely, TI-Neighborhood determines $N_{Eps}(p)\backslash\{p\}$ in D) and stores it in the seeds variable. Clearly, $N_{Eps}(p)$ determined in a reduced set D will not contain the neighboring points that were already moved from D to D'. In order to determine the real size of $N_{Eps}(p)$ in the original, non-reduced D, the auxiliary NeighborsNo field of point p is used. Whenever, a point p is moved from D to D', the NeighborsNo field of each of its neighboring points in D is incremented. As a result, the sum of the size of $N_{Eps}(p)$ found in the reduced D and p.NeighborsNo equals the size of $N_{Eps}(p)$ in the original, non-reduced set D.

If p is found not to be a core point, it is temporary labeled as a noise point, NeighborsNo field of each of its neighboring points in D is incremented, the information about p is added to the Border field of each of its neighboring points in D, p itself is moved from D to D', and the function reports failure of expanding a cluster.

Otherwise, the examined point is a core point and all points that are density-reachable from it will constitute a cluster. First, all points in the Eps-neighborhood of the analyzed point are assigned a label (ClId) of the currently built cluster and their NeighborsNo fields are incremented. Next all non-core points indicated by the

p.Border, which were stored in D', are found to be border points, and are assigned cluster label ClId, p.Border is cleared, and p is moved from D to D'. Now, each core point in seeds further extends the seeds collection with the points in its Eps-neighborhood that are still unclassified. After processing a seed point, it is deleted from seeds. The function ends when all points found as cluster seeds are processed.

Note that TI-ExpandCluster calculates Eps-neighborhood for each point only once.

```
function TI-ExpandCluster(var D, var D', point p, ClId, Eps, MinPts)
/* assert: D is ordered in a non-decreasing way w.r.t.  */
/* distances of points in D from the reference point r. */
/* assert: TI-Neighborhood does not return p.           */
seeds = TI-Neighborhood(D, p, Eps);
p.NeighborsNo = p.NeighborsNo + |seeds|;           // including p itself
if p.NeighborsNo < MinPts then    // p is either noise or a border point
    p.ClusterId = NOISE;
    for each point q in seeds do
        append p to q.Border; q.NeighborsNo = q.NeighborsNo + 1
    endfor
    p.Border = ∅; move p from D to D';     // D' stores analyzed points
    return FALSE
else
    p.ClusterId = ClId;
    for each point q in seeds do
        q.ClusterId = ClId; q.NeighborsNo = q.NeighborsNo + 1
    endfor
    for each point q in p.Border do
        D'.q.ClusterId = ClId;    //assign cluster id to q in D'
    endfor
    p.Border = ∅; move p from D to D';     // D' stores analyzed points
    while |seeds| > 0 do
        curPoint = first point in seeds;
        curSeeds = TI-Neighborhood(D, curPoint, Eps);
        curPoint.NeighborsNo = curPoint.NeighborsNo + |curSeeds|;
        if curPoint.NeighborsNo < MinPts then //curPoint is border point
            for each point q in curSeeds do
                q.NeighborsNo = q.NeighborsNo + 1
            endfor
        else                                  // curPoint is a core point
            for each point q in curSeeds do
                q.NeighborsNo = q.NeighborsNo + 1
                if q.ClusterId = UNCLASSIFIED then
                    q.ClusterId = ClId;
                    move q from curSeeds to seeds
                else
                    delete q from curSeeds
                endif
            endfor
            for each point q in curPoint.Border do
                 D'.q.ClusterId = ClId;   //assign cluster id to q in D'
            endfor
        endif
        curPoint.Border = ∅; move curPoint from D to D';
        delete curPoint from seeds
    endwhile
    return TRUE
endif
```

The TI-Neighborhood function takes the ordered point set D, point p in D, and Eps as input parameters. It returns $N_{Eps}(p)\backslash\{p\}$ as the set theoretical union of the point sets found by the TI-Backward-Neighborhood function and the TI-Forward-Neighborhood function. TI-Backward-Neighborhood examines points preceding currently analyzed point p, for which Eps-neighborhood is to be determined. The function applies Lemma 1 to identify first point, say q_b, preceding p in D such that distance(p,r) - distance(q_b,r) > Eps. All points preceding point q_b in D are not checked at all, since they are guaranteed not to belong to $N_{Eps}(p)$ (by Theorem 1). The points that precede p and, at the same time, follow q_b in D have a chance to belong to $N_{Eps}(p)$. For these points, it is necessary to calculate their actual distance to p (When using the Euclidean distance metric, the functions may apply the square of Distance and the square of Eps for efficiency reasons). The TI-Backward-Neighborhood function returns all points preceding p in D with the distance to p not exceeding Eps. The TI-Forward-Neighborhood function is analogous to TI-Backward-Neighborhood. Unlike TI-Backward-Neighborhood, TI-Forward-Neighborhood examines points following currently analyzed point p, for which Eps-neighborhood is to be determined. The TI-Forward-Neighborhood function returns all points following p in D with the distance to p not exceeding Eps.

```
function TI-Neighborhood(D, point p, Eps)
return TI-Backward-Neighborhood(D, p, Eps) ∪
       TI-Forward-Neighborhood(D, p, Eps)
```

```
function TI-Backward-Neighborhood(D, point p, Eps)
/* assert: D is ordered non-decreasingly w.r.t. dist */
seeds = {};
backwardThreshold = p.dist - Eps;
for each point q in the ordered set D starting from
    the point immediately preceding point p until
    the first point in D do
    if q.dist < backwardThreshold then       // p.dist – q.dist > Eps?
        break;
    endif
    if Distance(q, p) ≤ Eps then append q to seeds endif
endfor
return seeds
```

```
function TI-Forward-Neighborhood(D, point p, Eps)
/* assert: D is ordered non-decreasingly w.r.t. dist */
seeds = {};
forwardThreshold = Eps + p.dist;
for each point q in the ordered set D starting from
    the point immediately following point p until
    the last point in D do
    if q.dist > forwardThreshold then        // q.dist – p.dist > Eps?
        break;
    endif
    if Distance(q, p) ≤ Eps then append q to seeds endif
endfor
return seeds
```

Except for TI-DBSCAN, we have also implemented its variant TI-DBSCAN-REF [5] that uses many reference points instead of one for estimating the distance among pairs of points. Additional reference points are used only when the basic reference point according to which the points in D are sorted is not sufficient to estimate if a given point q belongs to Eps-neighborhood of another point p. The estimation of the distance between q and p by means of an additional reference point is based on Lemma 1. The actual distance between the two points q and p is calculated only when all reference points are not sufficient to estimate if $q \in N_{Eps}(p)$.

5 Performance Evaluation

In this section, we report the results of our experimental evaluation of TI-DBSCAN and TI-DBSCAN-REF as well as the original DBSCAN with R-Tree as an index. The number of reference points used by TI-DBSCAN-REF was set to the number of dimensions of a clustered dataset. In the experiments, we used a number of datasets (and/or their subsamples) of different cardinality and dimensionality. In particular, we used widely known datasets such as: birch [9], SEQUOIA 2000 [8], covtype [7] and kddcup 98 [10] as well as datasets generated automatically (random) or manually.

Table 2. Datasets used in experiments and run times (in milliseconds) of examined algorithms. Notation: dim. – number of point's dimensions, card. – number of points in a dataset, sort. – time of sorting of points, ind. – time of building of an index, clust. – clustering.

				TI-DBSCAN		TI-DBSCAN-REF		DBSCAN with R-tree	
No.	dataset	dim.	card.	sort.	clust.	sort.	clust.	ind.	clust.
1	birch	2	100000	176828	297	176828	1922	62829	229953
2	sequoia 2000	2	1252	47	15	47	16	1031	2406
3	sequoia 2000	2	2503	125	16	125	32	1536	3328
4	sequoia 2000	2	3910	219	46	219	62	2375	5500
5	sequoia 2000	2	5213	235	46	235	63	2891	8125
6	sequoia 2000	2	6256	406	62	406	47	3125	10969
7	sequoia 2000	2	62556	63312	266	63312	328	27813	201734
8	manual	2	2658	32	125	32	140	421	2219
9	manual	2	14453	812	2203	812	2172	1656	16516
10	random	3	50000	32984	77265	32984	41157	30515	264141
11	random	5	100000	151375	460281	151375	125078	100954	1289750
12	random	10	10000	1047	8531	1047	1312	17593	416297
13	random	10	20000	4094	33000	4094	5156	-	-
14	random	10	50000	25750	197422	25750	32609	-	-
15	random	20	500	0	0	0	15	484	1875
16	random	40	500	0	16	0	31	969	2906
17	random	50	10000	1453	41406	1453	5453	-	-
18	random	50	20000	4625	162781	4625	12985	-	-
19	covtype	54	150000	5137500	11936750	5137500	1036969	-	-
20	kddcup 98	56	56000	160172	1023953	160172	372062		
21	random	100	1000	47	938	47	1515	35937	84313
22	random	100	10000	1172	80578	1172	14391	-	-
23	random	100	20000	5610	345453	5610	38078	-	-
24	random	200	500	46	500	46	2968	-	-
25	random	200	1000	63	1813	63	6079	-	-

The run times of clustering with TI-DBSCAN, TI-DBSCAN-REF, and DBSCAN using R-Tree as an index are presented in Table 2. As follows from Table 2, TI-DBSCAN and TI-DBSCAN-REF are more efficient than DBSCAN even up to 600 times. TI-DBSCAN-REF tends to be faster than TI-DBSCAN for large high dimensional datasets. For small low dimensional datasets, TI-DBSCAN tends to be faster than TI-DBSCAN-REF.

6 Conclusions

In the paper, we have proposed two versions of our new algorithm: TI-DBSCAN and TI-DBSCAN-REF that produce the same results as DBSCAN, but use the triangle inequality to speed up the clustering process. TI-DBSCAN uses only one reference point, while TI-DBSCAN-REF uses many reference points. As follows from the experiments, both versions of our algorithm are much more efficient than the original DBSCAN algorithm, which is supported by a spatial index. Unlike DBSCAN, TI-DBSCAN and TI-DBSCAN-REF enable efficient clustering of high-dimensional data. The usage of many reference points is particularly useful in the case of large high dimensional datasets.

References

[1] Beckmann, N., Kriegel, H.P.: The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. In: Proc. of ACM SIGMOD, Atlantic City, pp. 322–331 (1990)
[2] Elkan, C.: Using the Triangle Inequality to Accelerate k-Means. In: Proc. of ICML 2003, Washington, pp. 147–153 (2003)
[3] Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Database with Noise. In: Proc. of KDD 1996, Portland, pp. 226–231 (1996)
[4] Guttman, A.: R-Trees: A Dynamic Index Structure For Spatial Searching. In: Proc. of ACM SIGMOD, Boston, pp. 47–57 (1984)
[5] Kryszkiewicz, M., Lasek, P.: TI-DBSCAN: Clustering with DBSCAN by Means of the Triangle Inequality, ICS Research Report, Warsaw University of Technology (April 2010)
[6] Kryszkiewicz, M., Skonieczny, Ł.: Faster Clustering with DBSCAN. In: Proc. of IIPWM 2005, Gdańsk, pp. 605–614 (2005)
[7] Moore, A.W.: The Anchors Hierarchy: Using the Triangle Inequality to Survive High Dimensional Data. In: Proc. of UAI, Stanford, pp. 397–405 (2000)
[8] Stonebraker, M., Frew, J., Gardels, K., Meredith, J.: The SEQUOIA 2000 Storage Benchmark. In: Proc. of ACM SIGMOD, Washington, pp. 2–11 (1993)
[9] Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: A New Data Clustering Algorithm and its Applications. Data Mining and Knowledge Discovery 1(2), 141–182 (1997)
[10] http://kdd.ics.uci.edu/databases/kddcup98/kddcup98.html

Vehicle Classification Based on Soft Computing Algorithms

Piotr Dalka and Andrzej Czyżewski

Gdansk University of Technology, Multimedia Systems Department,
Narutowicza 11/12, 80-233, Gdansk, Poland
{dalken,andcz}@sound.eti.pg.gda.pl

Abstract. Experiments and results regarding vehicle type classification are presented. Three classes of vehicles are recognized: sedans, vans and trucks. The system uses a non-calibrated traffic camera, therefore no direct vehicle dimensions are used. Various vehicle descriptors are tested, including those based on vehicle mask only and those based on vehicle images. The latter ones employ Speeded Up Robust Features (SURF) and gradient images convolved with Gabor filters. Vehicle type is recognized with various classifiers: artificial neural network, K-nearest neighbors algorithm, decision tree and random forest.

Keywords: vehicle type classification, SURF, Gabor filter, artificial neural network, K-nearest neighbors, decision tree, random forest.

1 Introduction

Vehicle type classification is a necessary component of every advanced road traffic monitoring system. Car monitoring is necessary also in parking lots, e. g. in a proximity of malls, sport stadiums, etc. Vehicle type information forms a very useful input for traffic density estimation and prediction. It can also be used for automatic detection of dangerous or prohibited events involving a particular group of vehicles (i. e. a truck overtaking another vehicle). Knowing the type of a vehicle is also helpful in systems of automatic toll collection on highways.

Classical, non-image based solutions designed to detect the type of a vehicle are based on counting the number of axles of vehicles via inductive loops or other types of sensors installed in the road surface or in its immediate vicinity. The disadvantage of such solutions, however, is their high cost (both installation and maintenance), little flexibility (there is no practical possibility of moving the detector to another position) and low effectiveness in case of high traffic. Therefore, non-invasive methods of vehicle detection by image analysis using video cameras are becoming more popular. They allow for easy installation, do not destroy the road surface and may classify vehicles from a large distance.

Video-based vehicle type classification systems presented in the literature usually follow the same basic scheme. First, all moving vehicles are detected and tracked in a video stream acquired from a camera in order to obtain their exact locations. On this basis, certain vehicle image parameters are calculated. They

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 70–79, 2010.

include various statistical parameters describing vehicle mask shape [1] or vehicle image itself [2,3].

Feature vectors are fed to the decision-making system. The classification is based on a set of defined rules [4,5] or is performed by comparing a feature vector with a database of samples. For this purpose, various distance-based methods (e.g. K-Nearest Neighbors algorithm) are used the most often [1,6]. Alternatively, artificial neural networks are employed [7].

Vehicle type classification may be based also on three-dimensional models of vehicles. During recognition, models are fit to the current vehicle image using various constraints. Best-fitting model is chosen as a result of classification [8,9].

Vehicle classification method may require a calibrated camera setup. Camera calibration involves estimation of a number of parameters related to the camera location and its field of view [10,11]. The measurement of these parameters is prone to errors but in the same time improves final results of vehicle classification. On the other hand, any change in the camera configuration requires recalibration. Calibration data is used to transform image acquired from a camera prior to feature extraction or to alter features after extraction.

Methods of vehicle classification that do not require camera calibration are much more convenient in terms of installation and usage but require employing more advanced algorithms.

This paper presents experiments regarding vehicle type classification with different image features and different classifiers with a non-calibrated camera setup. Section 2 presents the method for vehicle image acquisition from a road traffic camera. Section 3 describes image features used in experiments. Classifiers employed for vehicle type recognition are presented in Section 4. Section 5 discusses a method for selecting feature vectors constituting training and validation sets. Section 6 contains description of experiments carried out and their results. The last section concludes the paper.

2 Vehicle Image Acquisition

Video frames for the vehicle classification are captured by a traffic camera (Fig. 1). Its field of view is not calibrated i.e. it is not possible to determine real and absolute vehicle dimensions in this way. In order to classify vehicle type, vehicle images needs to be extracted from a video stream. In the first step, all moving objects (e.g. vehicles) are detected in every video frame acquired from a camera. The algorithm based on background modeling method utilizing Gaussian Mixtures is used for this purpose. It proved to be effective in previous experiments [12,13]. The results of background modeling are processed by detecting and removing shadow pixels (basing on the color and luminance of pixels) and by performing morphological operations on the detected objects in order to remove small areas and to fill holes inside objects.

Next, movements of the detected objects (blobs) are tracked in successive image frames using a method based on Kalman filters. Kalman filters allow predicting object position in the current frame. Comparing results of background

subtraction with predicted vehicle positions it is possible to correlate each object with its blob (including partial occlusions), so the movement of each object is tracked continuously [13,14]. All extracted images of every vehicle present in the analyzed video are used for vehicle type classification.

Fig. 1. Sample frame captured by a traffic camera used in experiments (left) and moving vehicle masks (right)

3 Vehicle Image Descriptors

Numerous vehicle image descriptors have been implemented for vehicle type classification. They may be divided into two groups. The first group includes feature based on vehicle mask only. They form *mask* feature set that contains:

- mask aspect ratio (the height of the mask bounding box divided by the width of the mask bounding box)
- eccentricity of the ellipse fitted to the mask
- extent, defined as the proportion of the mask bounding box area to the mask area
- solidity, defined as the proportion of the mask convex hull area to the mask area
- proportion of the square of the mask perimeter to the mask area 24 raw, central and normalized moments of the mask up to the third order (without trivial ones)
- a set of seven Hu invariant moments of the mask [15]; the moments are invariant under translation, changes in scale and rotation

Because the field of view of a camera is not calibrated and no homography image transformation is performed, vehicle image size and dimensions change during its presence in a video stream. Therefore, no feature containing direct, absolute values regarding vehicle size or dimensions are used. However vehicle size is an important factor in vehicle classification (e.g. trucks are larger than cars). Thus vehicle dimensions are included implicitly in statistical moments.

The second group of vehicle descriptors is based on image content. Because there is no correlation between vehicle type and its color, only luminance images are used. All image pixels outside of a vehicle mask are ignored. Three sets of vehicle image descriptors are computed; two of them are based on SURF (Speeded Up Robust Features) and the last set is derived from gradient images using Gabor filters.

Speed Up Robust Features (SURF) [16] is a scale- and rotation-invariant local image descriptor around a selected interest point. Its main advantages are repeatability, distinctiveness, and robustness as well as short computation time. Interest points (their location and sizes) may be chosen automatically using Fast-Hessian detector that is based on the determinant of the Hessian matrix [16]:

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \tag{1}$$

where $L(x,\sigma)$ are convolutions of the Gaussian second order derivatives with an image in point x at scale σ.

SURF descriptors ale calculated in the square regions centered around each interest point. The region is divided into 4×4 equal subregions. In each subregion, the Haar wavelet responses in horizontal d_x and vertical d_y directions (in relation to the interest point orientation) are calculated. Sums and absolute sums of wavelet responses form a four-element feature vector v for each subregion [16]:

$$v_4 = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|\right) \tag{2}$$

This results in a vector of length 64 that describes each interest point vicinity (SURF 64). The wavelet responses are invariant to illumination offset. Invariance to contrast is achieved by turning the descriptor into a unit vector. An alternative version of SURF descriptor is also used. A feature vector for each 4×4 subregion contains 8 elements as each sum and absolute sum is calculated separately depending on the sign of the wavelet response [16]:

$$v_8 = \left(\sum_{d_k<0} d_k, \sum_{d_k\geq0} d_k, \sum_{d_k<0} |d_k|, \sum_{d_k\geq0} |d_k|\right), \text{ where } k \text{ denotes } x \text{ or } y \tag{3}$$

This result in a SURF descriptor vector containing 128 elements for each interest point (SURF-128).

The first set of vehicle image descriptors is based on SURF-64. Interest points are selected automatically using Fast-Hessian interest point detector. For each interest point, SURF-64 vectors are calculated. A number of interest points found vary a lot and can exceed 100. Therefore it is reduced by clustering interest points with k-means algorithm [17]. This algorithm aims to partition n observation vectors into k sets ($k < n$) in order to minimize within-cluster dispersion (defined as a sum of squared Euclidean distances of observation vectors from the cluster center). Interest points are divided into eight sets based on their location in the vehicle image only and the mean vector of all SURF-64 descriptors for interest points from the same cluster is derived. The mean vector is augmented with the cluster center (vehicle image coordinates x and y normalized by the

vehicle mask height). All vectors are sorted according to the location of cluster centers. Final vehicle feature vector based on SURF-64 descriptors *surf-8-kmeans* contains $(64 + 2) \times 8 = 528$ elements.

The second set of vehicle image features is based on SURF-128 descriptors. They are obtained for four interest points that are set manually in the centers of four rectangular, non-overlapping areas the vehicle image is divided into; the areas are located symmetrically around a center of gravity of the vehicle mask. The size of each interest point is equal to the height or width of the area, depending on which value is greater. Final vehicle feature vector *surf-4* based on SURF-128 descriptors contains $128 \times 4 = 512$ elements.

The last set of vehicle image descriptors is based on filtering a gradient image with a bank of Gabor filters. Image gradients are calculated in vertical and horizontal directions independently using Sobel operator with an aperture size equal to 3. The final gradient image is obtained by adding squared vertical and horizontal gradients. Images are scaled to the fixed resolution 100×80 pixels.

Gabor filter kernels are similar to the 2D receptive field profiles of the mammalian cortical simple cells. Therefore they reveal desirable characteristics of spatial locality and orientation selectivity [18]. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave, according to the equation [19]:

$$g_{\lambda,\Theta,\varphi,\sigma,\gamma}(x, y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cdot \cos\left(2\pi\frac{x'}{\lambda} + \varphi\right) \tag{4}$$

$$x' = x\cos\Theta + y\sin\Theta, \qquad y' = -x\sin\Theta + y\cos\Theta \tag{5}$$

where λ denotes the wavelength of the cosine factor, Θ represents the orientation of the normal to the parallel stripes of a Gabor function, φ is the phase offset, σ is the sigma of the Gaussian envelope and γ is the spatial aspect ratio that specifies the ellipticity of the support of the Gabor function. In the experiments $\gamma = 1$ and $\sigma = 0.56\lambda$. The size $g^x \times g^y$ of a Gabor filter is set according to the equation:

$$g^x = 2 \cdot \max\left(\left|s \cdot \sigma\cos\Theta\right|, \left|s \cdot \frac{\sigma}{\gamma}\sin\Theta\right|\right) + 1 \tag{6}$$

$$g^y = 2 \cdot \max\left(\left|s \cdot \sigma\sin\Theta\right|, \left|s \cdot \frac{\sigma}{\gamma}\cos\Theta\right|\right) + 1 \tag{7}$$

where $s = 3$ determines a kernel size.

A bank of eight Gabor filters based on two different wavelengths λ (2.5 and 4) and four different orientations Θ (0°, 45°, 90° and 135°) is used. A scaled gradient image I is convolved with each Gabor filter g with two variants of phase offset φ, according to the equation:

$$I_G = \sqrt{\left(g_{\varphi=0} * I\right)^2 + \left(g_{\varphi=\pi/2} * I\right)^2} \tag{8}$$

This results in eight filtered vehicle images. For each image, seven Hu invariant moments are derived [15]. Final vehicle feature vector *gabor* contains $7 \times 8 = 56$ elements.

4 Classifiers

For the purpose of vehicle type recognition, four different classifiers have been examined: K Nearest Neighbors algorithm (KNN), Artificial Neural Network (ANN), Decision Tree (DT) and Random Forest (RF). All classifiers are trained with selected feature vectors and validated with a different set of vectors (Section 5).

KNN algorithm is the most simple of all classifiers used. It stores all training samples and predicts the response for a new feature vector by analyzing a certain number (K) of the nearest neighbors of the vector in the training set using Euclidean distance measure. In the experiments $K = 2$.

A feed-forward ANN with one hidden layer is used in experiments. The number of ANN inputs i_{ANN} corresponds with the number of vehicle features. The number of outputs o_{ANN} is equal to the number of vehicle types recognized. An expected output consists of a maximum value on one output and minimal values on other outputs. Therefore a vehicle type corresponding with the maximum output value is returned as the classification result. The number of neurons in the hidden layer h_{ANN} is set according to the equation:

$$h_{ANN} = \sqrt{i_{ANN} \cdot (o_{ANN} - 1)} \tag{9}$$

ANN is trained with a resilient backpropagation algorithm (RPROP). Sigmoid activation functions are used in all neurons. Due to the stochastic nature of ANN weights initialization, ANN is trained and validated five times and the average results are returned.

Decision Tree [20] is a model of computation in which an algorithm is considered to be a sequence of branching operations based on comparisons of some quantities.

Random Forest [21] is the last classifier used in experiments. RF is an ensemble classifier that consists of many decision trees. An input feature vector is classified with every tree and the final class label is decided based on individual classification results. Due to the stochastic nature of RF, it is trained and validated five times and the average results are returned.

5 Selection of Training and Validation Feature Vectors

In the result of moving vehicle detection and tracking (Section 2), every vehicle is represented by many images acquired during its presence in a camera field of view. Only vehicle images without occlusions are used in experiments; image validity is determined automatically based on moving object tracking data. Additionally all vehicles with mask bounding box area less than 2000 pixels are discarded due to low amount of details. All remaining images are used to calculate image descriptors and form feature vectors.

Training and validation vectors are selected in two stages. First, vehicles are divided into these two groups. Next, images are chosen for each vehicle. Each

vehicle is represented by T vehicles in the training set. Number of vehicles T is determined with an equation:

$$T = 0.5 \cdot \min_i (N_i) \tag{10}$$

where N_i denotes number of vehicles of type i. All remaining vehicles are assigned to the validation set. Therefore, numbers of vehicles of each type in a validation set are different. Vehicles are divided randomly between validation and training set.

Images for every vehicle are selected according to the same scheme for training and validation sets, separately. Each vehicle type is represented by the same amount of images chosen randomly from images of all vehicles of the type in the training or validation set. Number of images is equal to the size of the smallest image group. Therefore validation and training sets contain, independently, the same amount of images for every vehicle type.

6 Experiments and Results

For the purpose of experiments, a 30-minute video recording from a traffic camera (Fig. 1) has been selected. All moving vehicle images have been automatically extracted using vehicle detection and tracking algorithms (Section 2), validated and hand-labeled with an appropriate vehicle class. Classifiers were evaluated independently for vehicles from the lower and upper lane.

Three vehicle types are classified: sedans (include all small and medium cars), vans (include minibuses) and trucks (includes various medium and large one, also with semi-trailers). Sample vehicle images are presented in Fig 2. The database for the upper lane contains images of 525 different vehicles (367 sedans, 80 vans and 78 trucks) and the lower lane database contains 685 vehicles (569 sedans, 74 vans and 42 trucks). Total number of images is equal 48624 which means that each vehicle is represented by 40 images on average. All image processing operations and classifiers have been implemented in C++ with OpenCV library.

Table 1 presents results of vehicle type recognition using different feature sets and various classifiers. Every vehicle image is classified independently. It may be noticed, that shape descriptors alone provide high accuracy, but similar (or better in some cases) results may be achieved using image descriptors. In all cases, *surf-4* descriptors perform better than *surf-8-kmeans*. The final feature vector consists of *mask*, *surf-4* and *gabor* parameters; *surf-8-kmeans* set is omitted because of its correlation with *surf-4* and to reduce final vector dimensionality.

In majority of cases, ANN and RF classifiers perform better than the KNN and DT. Thus only the former ones are used in further experiments.

Each vehicle is represented by many images in the training and validation sets. Therefore results of classifications of images of the same vehicle can be aggregated in order to increase total effectiveness. The final class assigned to a vehicle is equal to the most frequently labeled class for all images of the vehicle; if there is a draw, the classification fails.

Fig. 2. Sample vehicle images from lower (left) and upper (right) lane for each vehicle type: first row - sedans, second row - vans, third row - trucks; images from the lower; vehicle images are rescaled individually to the same vertical size

Table 1. Summary results of vehicle type classification effectiveness by different feature sets and different classifiers

Feature set	Classifiers							
	Upper lane				Lower lane			
	ANN	KNN	DT	RF	ANN	KNN	DT	RF
mask	83.2%	80.6%	76.8%	82.1%	78.9%	81.1%	72.7%	78.8%
surf-4	84.4%	79.7%	70.0%	84.9%	82.0%	74.0%	61.1%	84.1%
surf-8-kmeans	74.3%	57.4%	63.4%	80.1%	74.8%	54.6%	60.3%	76.2%
gabor	81.7%	76.6%	70.6%	76.7%	73.7%	68.4%	64.0%	64.9%

Tables 2 and 3 present detailed results of vehicle classification for the lower and upper lane, accordingly. It is seen that aggregation of individual image classification results improves recognition rate by approx. 5 percent point. ANN and RF classifiers achieved similar results (within 2 percent points). Classification effectiveness of vehicles on the upper lane is 5 to 10 percent point better than for the lower lane. This behavior is probably caused by the fact that vehicles on the lower lane are located closer to the camera, therefore their physical dimensions and pose vary more during their presence in a camera field of view.

Table 2. Detailed results of vehicle type classification for the lower lane

Vehicle type	Without result aggregation			With result aggregation		
	No. of vehicle images	Classification effectiveness		No. of vehicles	Classification effectiveness	
		ANN	RF		ANN	RF
sedan	474	90.4%	90.3%	344	85.4%	88.0%
van	474	68.9%	70.3%	53	79.6%	82.6%
truck	474	90.2%	85.3%	21	96.2%	87.6%
all types	1422	83.1%	81.9%	418	85.2%	87.3%

Table 3. Detailed results of vehicle type classification for the upper lane

Vehicle type	Without result aggregation			With result aggregation		
	No. of vehicle images	Classification effectiveness		No. of vehicles	Classification effectiveness	
		ANN	RF		ANN	RF
sedan	2272	96.3%	92.0%	315	97.4%	95.0%
van	2272	78.8%	87.4%	41	94.4%	95.1%
truck	2272	89.3%	90.0%	39	84.6%	82.6%
all types	6816	88.1%	89.8%	395	95.8%	93.8%

7 Conclusions

Vehicle type classification is a highly complex task because of large variety of vehicles belonging to each class. Nevertheless, the system presented in the paper can classify up to 95% of vehicles correctly. Experiments presented in the paper prove that feature vector consisting of vehicle mask statistical parameters and image features based on SURF and Gabor filters is sufficiently universal to characterize vehicles with different pose, size and resolution. The best results are achieved with classifiers based on Artificial Neural Networks and Random Trees.

The search for optimal feature vector content for vehicle classification will continue. Other image descriptors will be examined in order to increase effectiveness of classification in a non-calibrated traffic camera system.

Acknowledgements. Research is subsidized by the European Commission within FP7 project "INDECT" (Grant Agreement No. 218086) and by the European regional development fund within the project POIG.01.01.02-00-062/09 "INSIGMA" ("Intelligent Information System for Detection and Recognition...").

References

1. Morris, B., Trivedi, M.: Improved Vehicle Classification in Long Traffic Video by Cooperating Tracker and Classifier Modules. In: IEEE Int. Conf. on Video and Signal Based Surveillance, AVSS (2006)
2. Petrovic, V.S., Cootes, T.F.: Analysis of Features for Rigid Structure Vehicle Type Recognition. In: British Machine Vision Conference, pp. 587–596 (2004)
3. Saito, M., Kitaguchi, K.: Appearance Modeling for Object Pose Recognition using Canonical Correlation Analysis. In: Int. Joint Conf. SICE-ICASE, pp. 2818–2821 (2006)
4. Chung-Lin, H., Wen-Chieh, L.: A Vision-Based Vehicle Identification System. In: Proc. 17th Int. Conf. on Pattern Recognition, vol. 4, pp. 364–367 (2004)
5. Gupte, S., Masoud, O., Papanikolopoulos, P.: Vision-Based Vehicle Classification. In: Proc. of Int. Transportation Systems, pp. 46–51 (2000)

6. Jun-Wei, H., Shih-Hao, Y., Yung-Sheng, C., Wen-Fong, H.: Automatic Traffic Surveillance System for Vehicle Tracking and Classification. IEEE Trans. on Int. Transportation Systems 7(2), 175–187 (2006)
7. Zhong, Q.: Method of Vehicle Classification Based on Video. In: IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics, AIM, pp. 162–164 (2008)
8. Sullivan, G.D., Baker, K.D., Worrall, A.D., Attwood, C.I., Remagnino, P.R.: Model-Based Vehicle Detection and Classification Using Orthographic Approximations. Image and Vision Computing 15(8), 649–654 (1996)
9. Buch, N., Orwell, J., Velastin, S.A.: Detection and Classification of Vehicles for Urban Traffic Scenes. In: 5th Int. Conf. on Visual Information Engineering, VIE (2008)
10. Worrall, A.D., Sullivan, G.D., Baker, K.D.: A simple intuitive camera calibration tool for natural images. In: BMVC, pp. 781–790 (1994)
11. Lai, A.H.S., Fung, G.S.K., Yung, N.H.C.: Vehicle Type Classification from Visual-Based Dimension Estimation. In: Proc. of Int. Transportation Systems, pp. 201–206 (2001)
12. Dalka, P.: Detection and Segmentation of Moving Vehicles and Trains Using Gaussian Mixtures, Shadow Detection and Morphological Processing. Machine Graphics and Vision 15(3/4), 339–348 (2006)
13. Czyzewski, A., Dalka, P.: Moving Object Detection and Tracking for the Purpose of Multimodal Surveillance System in Urban Areas. In: Proc. 1st Int. Symp. on Intell. Interactive Multim. Syst. and Services, Piraeus (2008)
14. Czyzewski, A., Dalka, P.: Examining Kalman filters applied to tracking objects in motion. In: Proc. of 9th Int. Workshop on Image Analysis for Multimedia Interactive Services, pp. 175–178 (2008)
15. Flusser, J., Suk, T.: Rotation Moment Invariants for Recognition of Symmetric Objects. IEEE Trans. Image Proc. 15, 3784–3790 (2006)
16. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
17. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An Efficient k-means Clustering Algorithm: Analysis and implementation. IEEE Trans. Pattern Analysis and Machine Intelligence 24, 881–892 (2002)
18. Daugman, J.G.: Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression. IEEE Trans. Acoustic, Speech and Signal Processing 36, 1169–1179 (1988)
19. Grigorescu, S.E., Petkov, N., Kruizinga, P.: Comparison of texture features based on Gabor filters. IEEE Trans. on Image Processing 11(10), 1160–1167 (2002)
20. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth, Belmont (1998)
21. Breiman, L.: Random Forests. Machine Learning 45(1), 5–32 (2001)

Controlling Computer by Lip Gestures Employing Neural Networks

Piotr Dalka and Andrzej Czyżewski

Gdansk University of Technology, Multimedia Systems Department,
Narutowicza 11/12, 80-233, Gdansk, Poland
{dalken,andcz}@sound.eti.pg.gda.pl

Abstract. Results of experiments regarding lip gesture recognition with an artificial neural network are discussed. The neural network module forms the core element of a multimodal human-computer interface called LipMouse. This solution allows a user to work on a computer using lip movements and gestures. A user face is detected in a video stream from a standard web camera using a cascade of boosted classifiers working with Haar-like features. Lip region extraction is based on a lip shape approximation calculated by the means of lip image segmentation using fuzzy clustering. ANN is fed with a feature vector describing lip region appearance. The descriptors used include a luminance histogram, statistical moments and co-occurrence matrices statistical parameters. ANN is able to recognize with a good accuracy three lip gestures: mouth opening, sticking out the tongue and forming puckered lips.

Keywords: human-computer interface, image processing, lip gestures, artificial neural network, Haar classifiers.

1 Introduction

Human-computer interfaces are designed to make working with a computer as natural, intuitive and effective [1,2]. Traditional interfaces, like keyboard and mouse, cannot be used by everyone (i.e. people with impaired hand movements) or in every situations (i.e. harsh environment conditions). Therefore there is an increasing need for development of new interfaces that would facilitate our everyday coexistence with machines.

One of the main areas of applications of new human-computer interfaces is to enable people with permanent or temporal disabilities to use computers in an efficient way. There are two main types of such solutions [3]. The first group utilizes devices mounted directly on the user's body. Applications in the second group are contactless and they use remote sensors only, therefore they are much more comfortable for a user. Amongst contactless solutions, vision-based human-computer interfaces are the most promising ones. They utilize cameras and image processing algorithms to detect signs and gestures made by a user and execute configured actions. The most common vision-based application employ eye and hand tracking [4,5].

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 80–89, 2010.

Intelligent decision systems are especially useful in the field of recognition of user gestures in a video stream [6,7]. They are able to solve complicated dependencies between input variables that are impossible to define manually and therefore make efficient gesture recognition possible. Furthermore, a decision system may be trained for a particular user or specific environmental conditions that would further improve the results. Training data can be acquired during short, initial calibration.

This paper presents a vision-based human-computer interface called LipMouse. It tracks user's lip movements and detects lip gestures using an artificial neural network. (ANN). This interface is described shortly in Section 2. Section 3 contains details regarding lip gesture recognition, including feature vector composition and ANN design. Results of experiments are presented in section 4. Section 5 concludes the paper.

2 Human-Computer Interface Description

LipMouse is the name of the patent-pending, contactless, human-computer interface that allows a user to work on a computer using lip movements and gestures. LipMouse is an application running on a standard PC computer. It requires only one hardware component: a display-mounted, standard web camera that captures images of the user face. The main task of the LipMouse is to detect and analyze images of user's mouth region in a video stream acquired from a web-camera. All movements of mouth (head) are converted to movements of the screen cursor. Various parameters regarding threshold, speed and acceleration of the cursor movement may be set according to user preferences. LipMouse detects three mouth gestures: opening the mouth, sticking out the tongue and forming puckered lips (as for kissing). Each gesture may be associated with an action, which may be freely chosen by a user. Possible actions include clicking or double-clicking various mouse buttons and moving mouse wheel – both horizontally and vertically.

Before a user starts working with LipMouse, a short calibration lasting about 30 seconds needs to be executed. During the calibration, the user is asked to perform some head movement and gestures according to the instructions seen on the screen. The purpose of the calibration is to tune LipMouse to detect gestures made by the user in the current lighting conditions.

The target users for the tool are people who, for any reason, cannot or do not want to use traditional input devices. Therefore LipMouse is a solution enabling severely disabled and paralyzed people to use a computer and communicate with the surrounding world. No user adaptation, such as placing marks on the face, is required in order to successfully work with LipMouse.

Fig. 1 presents a scheme of the algorithm used in LipMouse. First, a user's face is detected in every image frame captured by a web camera. A cascade of boosted classifiers working with Haar-like features is used for this purpose [8,9]. Further stages of the algorithm are restricted to the ROI containing the user's face. Then, mouth region is localized in the lower part of the face ROI and its

shift from the reference mouth position is calculated. This shift is directly used to move a screen cursor; the greater the shift is, the faster the cursor moves in a given direction. The reference mouth position may be altered at any time on user request. Simultaneously, a small region (blob) placed on user lips is found in the mouth region. This blob is used as a starting condition for an iterative method for lip shape extraction. Lip shape and lip region image features are used by an intelligent decision system utilizing an artificial neural network to classify gestures made by a user.

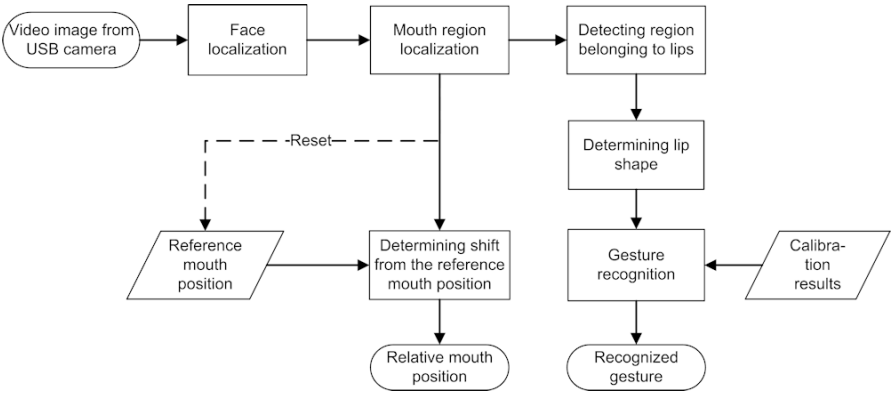


Fig. 1. Scheme of the LipMouse human-computer interface algorithm

3 Lip Gesture Recognition

It turned out during initial experiments that due to a large anatomical variety of faces and lips and hard to predict lighting conditions, defining strict, deterministic rules for lip gesture classification do not provide satisfactory results. Therefore a soft computing algorithm employing an artificial neural network (ANN) is used for lip gesture recognition. A feature vector for the ANN contains parameters describing image region containing lips only. Therefore the region needs to be found first.

3.1 Lip Region Extraction

In order to facilitate lip gesture recognition by ANN, an algorithm for determining region of the image containing lips only must be very precise and has to be robust against head movements in the vertical and horizontal directions. In order to locate lips, a series of face image transformations is performed [10]. They include converting a color spaces into CIE LUV space [11] and with DHT (Discrete Hartley Transform) [12], morphological closing and opening, spatial filtering and binary thresholding. In the result, one or more blobs placed on image lips are obtained. They are used as a starting point for an iterative process that approximates lip shape with an ellipse using fuzzy clustering [11]. In the

method, a dissimilarity measure that integrates the color dissimilarity and the spatial distance in terms of an elliptic shape function is used. Because of the presence of the elliptic shape function, the measure is able to differentiate the pixels having similar color information but located in different regions. Fig. 2 shows optimal results of lip shape approximation with an ellipse for all types of gestures recognized by ANN.

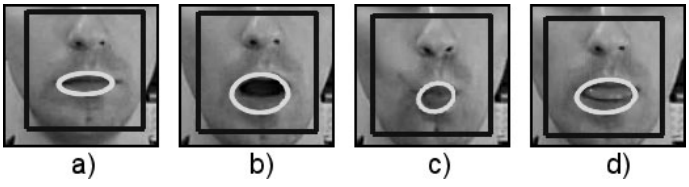


Fig. 2. Sample results of mouth region (red rectangle) and lip shape detection (green ellipse) for the lack of the lip gesture (a) and for three gestures recognized: opening the mouth (b), forming puckered lips (c) and sticking out the tongue (d)

Three different variants of lip region extracting are available. In the first variant (V_1), the lip region is based directly on the ellipse approximating the lip shape and is constituted by the rectangle containing the ellipse. It means that the lip region size is not constant and the region moves and tilts according to the results of lip shape approximation. In the second variant (V_2), horizontal, constant-size square is used as the lip region. Its center is always anchored at the center of the ellipse and the length of its sides is fixed and determined at the beginning of the calibration process by the width of the ellipse. In the third variant (V_3), influence of the ellipse on the lip region extracting is minimal. The lip region is formed by the square which size and position is fixed and determined at the beginning of the calibration phase. The center of the square is located at the center of the ellipse, and the length of sides is equal to the half of the width of the whole mouth region. The third variant is especially useful when the algorithm of lip shape approximation fails.

3.2 Lip Image Features

All lip image features used for lip gesture recognition can be divided into 7 groups. The first one, denoted as G_1, is used only when the first variant V_1 of lip region extracting is chosen and it contains three parameters: the width and the height of the ellipse approximating the lip shape and the angular eccentricity of the ellipse.

The second group of parameters G_2 is formed by the normalized, 20-point luminance histogram of the lip region.

The third group G_3 contains Hu sets of invariant image moments [13,14]. The moments are invariant under translation, changes in scale and rotation. Four sets of Hu moments are calculated based on four equal-sized, non overlapping luminance images the lip region is divided into. Each Hu set contains 7 parameters giving total number of 28 features in the third group.

The last three groups of parameters are based on co-occurrence matrices. A co-occurrence matrix, also referred to as a co-occurrence distribution, is defined over an image to be the distribution of co-occurring values at a given offset [15]. It is commonly used as a texture description. A set of co-occurrence matrices is calculated for three lip image representations: the luminance L and chrominance U of the CIE LUV color space and the first vertical derivative of the luminance image calculated with a Sobel operator [16]. Each set contains 8 normalized, symmetrical, 25×25 co-occurrence matrices (four possible directions: 0°, 45°, 90°, 135° and two distances: 1 and 2). Five statistical parameters are calculated for every co-occurrence matrix: contrast, energy, mean, standard deviation and correlation [15]. This brings total number of 120 parameters (3 image representations $\times$ 8 co-occurrence matrices $\times$ 5 parameters) based on co-occurrence matrix in the ANN feature vector. Statistical parameters of co-occurrence matrices for each of the three image representations form the last three groups of parameters: G_5 – luminance L, G_6 – chrominance U, G_7 – luminance vertical derivative.

3.3 Artificial Neural Network Description

A feed-forward ANN with one hidden layer is used to detect lip gestures. Each image frame is classified independently. The number of ANN inputs corresponds to the number of lip image features and is equal 168 or 171, depending on the chosen variant of lip region extraction. There are 4 outputs from ANN, each one is related with one type of gestures recognized by ANN. Three of them are: opening the mouth, sticking out the tongue and forming puckered lips. A natural, neutral facial expression is the fourth gesture and means that no real lip gesture is present. Based on initial experiments, number of neurons in the hidden layer was set to 8 (see experiments in section 5). It is the minimum number of neurons sufficient for good effectiveness of lip gesture recognition. Sigmoid activation functions are used in all neurons.

A type of the gesture is determined by the maximum value of the ANN outputs. However, in the HCI application it is crucial to minimize false-positive rate of detection of all three, real gestures in order to prevent execution of actions not meant by a user. False-negative rate is less important – if a gesture is not recognized in some frame, it will be recognized in succeeding frames when a user moves his head a little or change face expression.

In order to minimize number of false-positives, post processing of ANN output vector o is performed in order to determine reliability of classification. In the first step, ANN output values are scaled from [-1, 1] range to [0, 1] range with the formula:

$$o'(i) = 0.5 \cdot o(i) + 0.5 \quad \text{for } i \in \{0\ 1\ 2\ 3\} \tag{1}$$

Next, output values are converted according to the equation:

$$o''(i) = \frac{o'(i)}{\sum_{i=0}^{3} o'(i)} \cdot \max_i (o'(i)) \tag{2}$$

If the maximum value of o'' vector is greater or equal to the threshold T, a gesture connected with the maximum output value is returned as the recognized gesture; otherwise, the neutral gesture is returned which means that no real gesture is detected. This method assures that if the neural network output is not firm, no gesture is detected in order to minimize false-positives ratio. It can be noticed that $T = 0$ turns off ANN output post-processing.

ANN is trained with a resilient backpropagation algorithm (RPROP). Training data are acquired during calibration phase which is required at the beginning of every session employing LipMouse. The calibration consists of 4 stages. During each stage, a user is asked to move his head left, right, up and down while making one of the four gestures: neutral one in the stage 1, mouth opening in the stage 2, sticking out the tongue in the stage 3 and forming puckered lips in the stage 4. Each stage lasts 4 second, with 2 second break between stages when a user is asked to change the lip gesture. During each stage, 60 frames containing gesture images are gathered (video rate is 15 fps). Feature vectors obtained from these frames form training vectors (80% of all vectors) and validation vectors (every fifth vector). This means that total 192 feature vectors (48 for every gesture) are used for ANN training and 48 vectors are used to validate ANN after training (12 for every gesture). Five neural networks are trained based on the same data and the one with the smallest error rate of validation vector classification (with post-processing threshold $T = 0.5$) is used for lip gesture recognition.

Lip gesture is classified for each video frame independently. In the human-computer interface application, the results are time-averaged in order to eliminate single detection errors. The gesture that has been detected prevalently during past n milliseconds is reported as final result of classification. This introduces a short delay in gesture change detection but in the same time significantly reduce classification errors that are possible during gesture transients (e.g. the transition from no gesture to sticking out the tongue might result in forming puckered lips gesture detection). The default averaging time frame duration n is equal to 350 ms.

4 Experiments and Results

For the purpose of lip gesture recognition experiments, face recordings of 102 persons were collected (Fig. 3). Each person was asked to carry out typical calibration procedure twice. The first iteration was used to train ANN and the second iteration was used to obtain the effectiveness of lip gesture classification. All face images gathered during the second iteration were used for testing, therefore the testing set of vectors contained 25% more elements than the training set of vectors (20% of vectors gathered during the first iteration is used for instant ANN validation). Each image frame is classified independently.

In the initial experiments, the proper number of neurons in the hidden layer of ANN has been chosen. Table 1 shows results of lip gesture classification for four, eight and sixteen neurons in the hidden layer. Because the effectiveness of neutral gesture recognition is crucial in HCI applications, an ANN with $h = 4$

Fig. 3. Sample frames from test recordings

neurons in the hidden layer is rejected in the first place. Results of ANNs with $h = 8$ and $h = 16$ are similar. Eventually, the ANN with eight neurons has been chosen as an optimal one because of the training phase duration; training of five ANNs lasts less than 2 seconds on a computer equipped with 2 GHZ CPU and therefore is almost transparent to the HCI application user while the training period for ANN $h = 16$ is unacceptably long.

Table 1. Results of lip gesture classification for different number of neurons in ANN hidden layer h (ANN post-processing is turned off and the optimal variant of lip gesture extraction is used for every test recording)

Number of neurons in the ANN hidden layer h	Effectiveness of lip gesture classification [%]				
	Neutral gesture	Mouth opening	Forming puckered lips	Sticking out the tongue	All gestures
4	89.2	95.3	89.9	90.1	91.1
8	92.6	95.3	90.4	91.3	92.4
16	92.3	94.8	92.9	91.1	92.8

In order to determine the proper content of a feature vector, experiments with different combination of parameter groups have been conducted (Table 2). It may be noticed that each parameter group separately provides a high lip gesture detection accuracy, however combination of all parameters allows achieving results better by approx. 10 percent points. Although other parameter combinations provide better results for some test recordings or lip region extracting variants, the combination of all parameters is chosen as the final composition of the feature vector, because of its universality and good results in all conditions.

Table 3 presents a summary of lip gestures recognition for different lip region extracting variants. It is seen that the best results were achieved for the third variant V_3 that relies at least on the results of lip shape approximation with an

Table 2. Results of lip gesture classification for different groups of parameters (Section 3.2) in the feature vector (ANN post-processing is turned off and the optimal variant of lip gesture extraction is used for every test recording)

Group of image features	Effectiveness of lip gesture classification [%]				
	Neutral gesture	Mouth opening	Forming puckered lips	Sticking out the tongue	All gestures
G_1	88.3	67.9	83.2	71.2	77.7
G_2	80.0	91.7	77.0	82.0	82.7
G_3	81.8	81.1	69.7	72.1	76.2
G_4	82.0	92.9	85.3	83.0	85.8
G_5	81.6	85.0	80.0	82.5	82.3
G_6	82.9	86.2	84.2	85.8	84.8
All parameters	92.9	95.4	92.5	94.1	93.7

ellipse. This means that although the ellipse usually fits the real lip shape, its inter-frame variances, especially during head movements, might interfere with the ANN classification effectiveness. It was discovered that V_3 variant provides the best results for 82% of test recordings. Other variants were optimal (V_1 for 14% and V_2 for 4% of test recordings) for those recordings where the lip shape is approximated perfectly with an ellipse in all movie frames. This allows to conclude that V_2 variant is not needed; in case of any problems with lip shape approximations V_3 wins, in other cases V_1 variant is usually sufficient.

Detailed results of lip gesture classification are shown in Table 4. Increasing ANN post-processing threshold T improves effectiveness of neutral gesture recognition and worsens results of other three gestures classification. It is assumed that an optimal value of the T threshold is 0.5 which provides a compromise between the effectiveness and the false-positive ratio of real gesture recognition.

Achieved results of lip gesture classification are satisfactory. Total effectiveness of recognition over 90% means that on average three recognition errors appear every two seconds of algorithm working. Furthermore, due to ANN output post-processing, the majority of the errors emerge when the neutral gesture is recognized instead of other three gestures. These errors do not pose much

Table 3. Summary results of lip gesture recognition for different lip region extracting variants (ANN post-processing is turned off)

Lip region extracting variant	Effectiveness of lip gesture classification [%]				
	Neutral gesture	Mouth opening	Forming puckered lips	Sticking out the tongue	All gestures
V_1	86.1	85.3	85.4	84.8	85.4
V_2	80.2	83.0	75.0	78.8	79.3
V_3	91.3	95.3	92.0	94.1	93.2

Table 4. Results of lip gesture classification for different ANN post-processing threshold T values (optimal variant of lip gesture extraction is used for every test recording)

Gesture	Image frames	$T = 0$		$T = 0.25$		$T = 0.5$		$T = 0.75$	
		Errors /	Accuracy[%]	Errors /	Accuracy[%]	Errors /	Accuracy[%]	Errors /	Accuracy[%]
Neutral (no gesture)	6120	436	92.9	380	93.8	310	94.9	236	96.1
Mouth opening	6120	282	95.4	321	94.8	463	92.4	659	89.2
Forming puckered lips	6120	462	92.5	504	91.8	723	88.2	1004	83.6
Sticking out the tongue	6120	362	94.1	419	93.2	530	91.3	883	85.6
All gestures	24480	1542	93.7	1624	93.4	2026	91.7	2782	88.6

inconvenience to a user and may be attenuated further by the means of simple time-averaging of lip gesture detection results.

5 Conclusions

Development of new HCI solutions and improving existing ones is necessary to facilitate our everyday interactions with computers.

A practical implementation of an artificial neural network in the field of vision-based gesture recognition was presented in the paper. ANN enabled classifying lip gestured made by various people, in different lighting conditions. Results of experiments carried out show that the effectiveness of the algorithm is sufficient for comfortable and efficient usage of a computer by anyone who does not want or cannot use traditional keyboard and mouse.

Future work will be focused on feature vector content optimization and on finding new lip region descriptors in order to increase the number of lip gestures recognized.

Acknowledgements. Research funded within the project No. POIG.01.03.01-22-017/08, entitled "Elaboration of a series of multimodal interfaces and their implementation to educational, medical, security and industrial applications". The project is subsidized by the European regional development fund and by the Polish State budget.

References

1. Baecker, R.M., Grudin, J., Buxton, W.A.S., Greenberg, S. (eds.): Readings in human-computer interaction. Toward the Year 2000, 2nd edn. Morgan Kaufmann, San Francisco (1995)
2. Sears, A., Jacko, J.A. (eds.): Handbook for Human Computer Interaction, 2nd edn. CRC Press, Boca Raton (2007)

3. Aggarwal, J.K., Caim, Q.: Human Motion Analysis: A Review. CVIU (73) 3, 428–440 (1999)
4. Duchowski, A.T.: A Breadth-First Survey of Eye Tracking Applications. Behavior Research Methods, Instruments & Computers (BRMIC) 34(4), 455–470 (2002)
5. Shin, G., Chun, J.: Vision-based Multimodal Human Computer Interface based on Parallel Tracking of Eye and Hand Motion. In: Int. Conf. on Convergence Information Technology, pp. 2443–2448 (2007)
6. Moon-Jin, J., Seung-Eun, Y., Zeungnam, B.: User adaptive hand gesture recognition using multivariate fuzzy decision tree and fuzzy garbage model. In: IEEE Int. Conf. on Fuzzy Systems, pp. 474–479 (2009)
7. Rashid, O., Al-Hamadi, A., Michaelis, B.: A framework for the integration of gesture and posture recognition using HMM and SVM. In: IEEE Int. Conf. on Intelligent Computing and Intelligent Systems, vol. 4, pp. 572–577 (2009)
8. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: IEEE CVPR (2001)
9. Lienhart, R., Maydt, J.: An Extended Set of Haar-like Features for Rapid Object Detection. In: IEEE ICIP, vol. 1, pp. 900–903 (2002)
10. Dalka, P., Czyzewski, A.: Lip movement and gesture recognition for a multimodal human-computer interface. In: Proc. of the Int. Multiconf. on Computer Science and Information Technology, pp. 451–455 (2009)
11. Leung, S., Wang, S., Lau, W.: Lip image segmentation using fuzzy clustering incorporating an elliptic shape function. IEEE Transactions on Image Processing 13(1), 51–62 (2004)
12. Moran, L., Luis, E., Pinto, R.: Automatic Extraction of the Lips Shape Via Statistical Lips Modelling and Chromatic Feature. In: Electronics, Robotics and Automotive Mechanics Conference, CERMA, pp. 241–246 (2007)
13. Flusser, J.: On the Independence of Rotation Moment Invariants. Pattern Recognition 33, 1405–1410 (2000)
14. Flusser, J., Suk, T.: Rotation Moment Invariants for Recognition of Symmetric Objects. IEEE Trans. Image Proc. 15, 3784–3790 (2006)
15. Clausi, D.A.: An analysis of co-occurrence texture statistics as a function of grey-level quantization. Canadian Journal of Remote Sensing 28(1), 45–62 (2002)
16. Young, I., Gerbrands, J., Vliet, L.: Fundamentals of Image Processing. Delft University of Technology, The Netherlands (1998)

Computer Animation System Based on Rough Sets and Fuzzy Logic

Piotr Szczuko

Multimedia Systems Department, Gdansk University of Technology
szczuko@sound.eti.pg.gda.pl
http://sound.eti.pg.gda.pl

Abstract. A fuzzy logic inference system was created, based on the analysis of animated motion features. The objective of the system is to facilitate the creation of high quality animation by analyzing personalized styles contained in numerous animations. Sequences portraying a virtual character acting with a differentiating personalized style (natural or exaggerated) and various levels of fluidity were prepared and subjectively evaluated. Knowledge gathered in subjective evaluation tests was processed utilizing variable precision rough set (VPRS) approach for defining non-ambiguous inverse relation between subjective features of the result animation and objective parameters of the animated motion. Once the mapping is known then the user can define own requirements on animation, and the input motion is processed accordingly to produce the desired result. The paper focuses on employing variable precision rough set methodology for selection of representative parameter values.

Keywords: variable precision rough set, subjective features processing, computer animation.

1 Introduction

Currently two major techniques for animation production are used. Animated motion can be obtained utilizing motion capture technology [1]. The result animation is realistic and natural, but this technology is expensive, requires experienced actors, motion data are hard to edit and practically cannot provide an exaggerated, cartoon motion. Another method is keyframe animation [2]. In this method the representation of animation data is intuitive, clear, and easy to edit. Moreover, the method doesn't require expensive hardware, the motion can be either natural or exaggerated, but that technique is very time-consuming and the result greatly depends on animator skills. To facilitate the creation and development of high quality animated motion we proposed new improvement to keyframe animation: fuzzy processing that considers subjective requirements for the action's style and fluidity. The method is outlined in Sec. 2. The engineered fuzzy logic inference system utilizes knowledge obtained from subjective evaluation tests. In Sec. 3 processing of test results for defining the mapping between subjective parameters and objective features of motion are described, then results are discussed and some conclusions are provided.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 90–99, 2010.

2 Animation Processing Method

In our work it is assumed that any animation can be segmented into parts containing first a still pose then one or more transitions and then the second still pose (Fig. 1). Each animation segment is parameterized and processed separately utilizing the described methodology. Typically animation containing those discernable segments is called pose-to-pose animation.

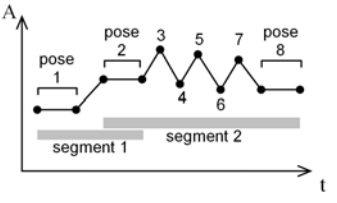


Fig. 1. Animation segmentation. Changes in time of one animation parameter (location of moving hand) are shown. Segment 1 contains one transition, segment 2 contains six transitions. Boundaries of segments are determined by poses being held in time (pose 1, 2, and 8).

For pose-to-pose animation traditional animation rules created by animators of Walt Disney Studio [3] suggest to place the anticipation phase before the starting pose, preparing the viewer for the action. Anticipation is often represented as a slight motion in a direction opposite to the following main action. Then, after the ending pose the other phase should be inserted, called overshoot, reflecting the way the character stops his motion. Decisions on amplitudes and lengths of these phases come from experience, and it is the animator's task to portray the character personality and the animation style by these phases. To help the animator in calculation of these parameters the ANIMATOR system was proposed and created. It was observed and described in literature [4], that fast motion with large amplitude should be preceded by a large anticipation effect; fast and long motion should be preceded by a long anticipation. Therefore we assumed that some proportionality occurs between motion speed, amplitude and length, i.e. respectively parameters V_m, A_m, t_m, and anticipation amplitude and length, i.e. parameters A_a, t_a[1]. The proportionality between the motion and additional phases, can be represented as:

$$A_a = V_m A_m alpha \quad \text{and} \quad t_a = V_m t_m beta \tag{1}$$

where $(alpha, beta)$ are proportionality coefficients being analyzed in our work. During initial subjective evaluation tests relations (1) were employed for

[1] Overshoot phase is processed utilizing the same methodology. For clarity this part is omitted.

generation of animations. Discrete values of coefficients were used: $alpha = \{0.3, 0.4, \ldots, 1.2, 1.3\}$, and $beta = \{1, 3, 5, 7\}$[2]. Animations were then rated for $style = \{natural, medium, exaggerated\}$, $fluidity = \{abrupt, medium, fluid\}$, and $quality = \{1, 2, 3, 4, 5\}$. Animations of a simple character were presented on the computer screen, alongside the rating graphical interface (Fig. 2). A condition was made that each animation must be rated at least by one viewer. Some tendencies were discovered, e.g. *style* increases with the increase of *alpha* coefficient value, and *fluidity* increases with the increase of *beta* coefficient value. Moreover changing *style* does not cause *quality* changes, and strong positive correlation exists between *fluidity* and *quality*. These observations are reflected in values of particular correlations (Table 1).

Table 1. Correlations between $(alpha, beta)$ coefficients and $(style, fluidity)$ ratings in the subjective test

	beta −*style*	*beta* −*fluidity*	*beta* −*quality*	*alpha* −*style*	*alpha* −*fluidity*	*alpha* −*quality*	*style* −*fluidity*	*style* −*quality*	*fluidity* −*quality*
R	-0.14	0.86	0.81	0.82	0.16	0.09	-0.21	-0.27	0.94

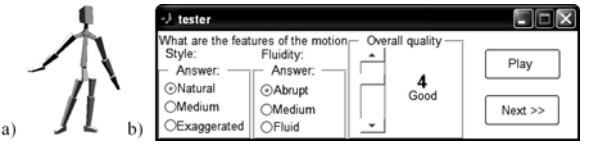


Fig. 2. (a) The animated character utilized in motion evaluation tests; (b) graphical user interface dedicated to e subjective tests

Overall subjective quality rated in a test is averaged giving a mean opinion score value (MOS) for a single animation, represented as a pair of $(alpha, beta)$ values. Results of MOS per each animation are presented in Table 2.

It is clear that the relation between $(fluidity, style)$ motion attributes and $(alpha, beta)$ coefficients can be defined based on these results. For utilizing the above observations in the ANIMATOR system the mapping between subjective requirements $(fluidity, style)$ and animation parameters $(alpha, beta)$ should be determined. Once the mapping is discovered, then the system can process any animation described with (V_m, A_m, t_m), and based on $(fluidity, style)$ requirements given by the user can calculate $(alpha, beta)$. This finally results in (A_a, t_a) parameters of motion phases that must be added to the animation. The processing methodology of the evaluation test results is described in Sec. 3.

[2] It was first verified which ranges of values return subjectively acceptable motions, and then how large changes (discretization steps) return significant changes of animation features. Results are $\langle 0.3; 1.3 \rangle$ with step 0.1 for *alpha* and $\langle 1; 7 \rangle$ with step 2 for *beta*.

Table 2. Preferences of test participants, MOS Q for animations described with pairs of coefficients $(alpha, beta)$

	$alfa$										
	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3
1	3.21	3.16	2.68	2.91	2.92	3.3	2.8	3.11	3.08	3.06	3.07
beta 3	3.22	3.19	3.2	3.14	2.92	3.41	2.41	3.42	3.23	3.04	3.28
5	3.7	4.03	3.8	3.5	3.69	3.8	3.82	3.35	3.41	3.47	3.53
7	4.17	4.04	3.92	3.7	3.8	4.1	3.9	3.5	3.5	3.85	3.69

3 Application of VPRS to Subjective Results Processing

In our work rough sets are employed for processing of ambiguous results of subjective evaluation tests. In the test inconsistent decisions are made for animations described with the attributes analyzed. The aim of processing the subjective evaluation results is to select representative values of parameters $(alpha, beta)$ for each combination of subjective features $(fluidity, style)$. Non-ambiguous inverse relation has to be found, for given decisions pointing at values of attributes. Therefore we have to choose from all objects only one candidate that was the most frequently rated as having particular decisions $(fluidity, style)$.

For further discussion let us assume that we deal with two decision systems: the objects in both decision systems are animations described with attributes: $(alpha, beta)$, and the first decision is the rating of fluidity, and the second one is the rating of the animation style. Each time the animation is assigned $(fluidity, style)$ ratings during the subjective test, then it is added to the decisions systems with those decision values. Because generally the viewers' answers are not consistent, then a single animation can appear repeatedly in decision systems, each time with different decisions, and relation $R : (alpha, beta) \rightarrow (fluidity, style)$ is not objective. Therefore non-ambiguous selection of representative objects $(alpha, beta)$ for required $(fluidity, style)$ ratings is made by means of rough sets, namely calculation of lower approximation of sets of animations that yield required results in the subjective evaluation test. Frequency of ratings is analyzed by means of rough sets. Three ratings are available for attributes: $fluidity = \{abrupt, medium, fluid\}$ and $style = \{natural, medium, exaggerated\}$. Therefore in fluidity and style domains three rough sets are analyzed, each having its accuracy of approximation, being the result of consistent or inconsistent ratings. The accuracy of the approximation of set X (e.g. $style = medium$) with objects described with attributes $B = (alpha, beta)$ is defined as [5]:

$$\alpha_B(X) = \frac{|\underline{B}X|}{|\overline{B}X|} \tag{2}$$

where:

$$\underline{B}X = \{x | [x]_B \subseteq X\} \quad \text{and} \quad \overline{B}X = \{x | [x]_B \cap X \neq \emptyset\} \tag{3}$$

$\alpha_B(X)$ describes effectiveness of the approximation by comparing number of objects in the lower approximation to the number of objects of the higher approximation. If the approximation is exact then $\alpha_B(X) = 1$. If $\underline{B}X$ is empty, the rough set is non-deterministic and $\alpha_B(X) = 0$. B-indiscernible objects noted as $[x]_B$, are all instances of a single animation x described with particular attributes B. Therefore the accuracy of the approximation accounts to the number of animations obtaining consistent decisions, compared to the number of all objects, that were at least once rated as belonging to the set X (e.g. were given the rating *medium* for *style*). Changing the precision of approximation with parameter π [6], requires redefining the lower and upper approximations (3) to:

$$\underline{B}_\pi X = \{x | \mu_B^X(x) \geq \pi\} \quad \text{and} \quad \overline{B}_\pi X = \{x | \mu_B^X(x) > 1 - \pi\} \tag{4}$$

where:

$$\mu_B^X : U \rightarrow [0; 1] \quad \text{and} \quad \mu_B^X(x) = \frac{|[x]_B \cap X|}{|[x]_B|} \tag{5}$$

$\mu_B^X(x)$ can be interpreted as a measurement of intersection of $[x]_B$ and X. For example if 3 of 4 B-indiscernible objects in a set $[x]_B$ (animations similar by means of attributes B) are rated as X, then $\mu_B^X(x) = \frac{3}{4} = 0.75$. If $\pi = 1$ then the set is classically rough. Lower π leads to assigning more objects to the lower approximation. In the example above, if $\pi = 0.75$ then x is in the $\underline{B}_\pi X$, if $\pi = 1$ then x is only in $\overline{B}_\pi X$, and $\underline{B}_\pi X$ is empty. Another interpretation may be that if an object was rated with regard to the particular decision X in more than π cases (e.g. more than 0.75 cases), then it belongs to the lower approximation of X. In the following sections we will refer to it as "frequency" of ratings.

In our application, if no animations are in the lower approximation of analyzed set X, then the precision is gradually lowered until $\underline{B}_\pi X$ will contain at least one animation. Eventually any combination of decisions $(fluidity, style)$ will have at least one representative x described by $B = \{alpha, beta\}$. If there are more equivalent candidates, then an additional criterion in considered - the value of MOS of overall quality Q, to choose the best between them. Moreover, values of π should be at the level for which any object is representative for no more than one rough set. It is fulfilled when lower approximations of all sets are disjoint (Fig. 3).

Particular rating frequencies for animations made with all combinations of $(alpha, beta)$ are presented in Tables 3. Value 0.0 means no particular answer (e.g. $style = natural$) was given for an animation with the corresponding $(alpha, beta)$; 1.0 means that all participants' answers have been the same (e.g. $style = natural$) for animations with the corresponding $(alpha, beta)$. Bold numbers mark answers more frequent than 0.5.

It was shown that the rating frequency is connected to the precision of the rough set, and provides decision if the object is in the lower approximation of a set X or not. For classical rough sets with precisions $\pi_{style} = 1$ and $\pi_{fluidity} = 1$ accuracies of approximations equal:

$$\begin{aligned} &\alpha_B(style\_natural) = \alpha_B(style\_exaggerated) = \alpha_B(style\_medium) = 0 \\ &\alpha_B(fluidity\_fluid) = \alpha_B(fluidity\_abrupt) = \alpha_B(fluidity\_medium) = 0 \end{aligned} \tag{6}$$

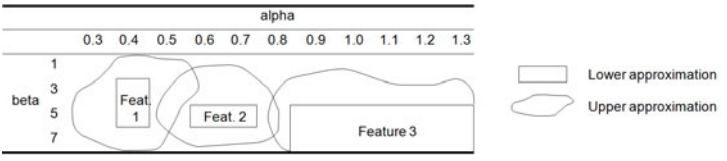


Fig. 3. Graphical illustration of defining precisions of set approximation utilizing the rough set methodology. We decrease the precision aiming at disjoint lower approximations of all sets, and no intersection with the higher approximation of any other set.

Table 3. Frequency of ratings *style* = *natural*, *style* = *medium* and *style* = *exaggerated*

	Natural										
						alpha					
	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3
beta 1	**0.69**	0.43	0.23	0.27	0.22	0.23	0.21	0.16	0.18	0.15	0.19
3	**0.69**	0.45	0.31	0.42	0.36	0.39	0.23	0.30	0.25	0.23	0.23
5	**0.54**	**0.63**	0.32	0.31	0.29	0.36	0.25	0.38	0.34	0.30	0.29
7	**0.56**	**0.71**	0.37	0.30	0.28	0.33	0.25	0.22	0.27	0.29	0.29

	Medium										
						alpha					
	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3
beta 1	0.21	0.36	**0.53**	0.21	0.19	0.19	0.22	0.27	0.20	0.21	0.21
3	0.26	0.16	0.23	0.09	0.11	0.06	0.18	0.16	0.13	0.12	0.12
5	0.35	0.14	0.17	0.14	0.13	0.08	0.06	0.14	0.13	0.13	0.07
7	0.29	0.12	0.15	0.21	0.06	0.06	0.16	0.21	0.18	0.17	0.13

	Exaggerated										
						alpha					
	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3
beta 1	0.10	0.21	0.24	**0.52**	**0.59**	**0.58**	**0.57**	**0.57**	**0.62**	**0.64**	**0.60**
3	0.05	0.38	0.46	0.48	**0.53**	**0.54**	**0.60**	**0.55**	**0.62**	**0.65**	**0.65**
5	0.11	0.22	**0.51**	**0.55**	**0.58**	**0.56**	**0.69**	0.48	**0.53**	**0.57**	**0.64**
7	0.15	0.17	0.48	0.49	**0.66**	**0.61**	**0.59**	**0.57**	**0.55**	**0.54**	**0.58**

because the lower approximations of the rough sets are empty. For lowered precisions of rough sets: $\pi_{style} = 0,57$ and $\pi_{fluidity} = 0,53$ accuracies are:

$$\begin{aligned} &\alpha_B(style\_natural) = 0,5 \\ &\alpha_B(style\_exaggerated) = 0,457 \\ &\alpha_B(style\_medium) = 0 \\ &\alpha_B(fluidity\_fluid) = 1 \\ &\alpha_B(fluidity\_abrupt) = 1 \\ &\alpha_B(fluidity\_medium) = 0,714 \end{aligned} \tag{7}$$

Rough sets obtained with parameters π_{style} and $\pi_{fluidity}$ given above are taken as a global approximations, and as the starting point for calculation of local solutions, i.e. subsets of objects representing particular grades of $(style, fluidity)$. Result memberships of objects to the lower approximation and boundary regions for all analyzed sets are presented in Fig. 4 5.

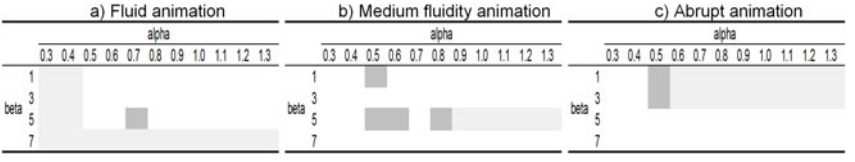


Fig. 4. Membership of objects described with particular $(alpha, beta)$ assigned to the rough set of a given fluidity rating. Light gray areas - objects of the lower approximation; dark gray - objects of the boundary region; white areas - the complement of the set, i.e. objects outside the rough set.

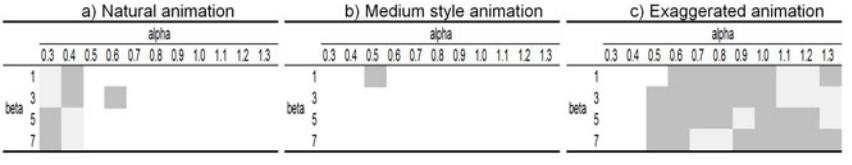


Fig. 5. Membership of objects described with particular $(alpha, beta)$ assigned to the rough set of a given style degree

Once relations of $(alpha, beta)$ with each subjective $(fluidity, style)$ features are known, the combined relation must be defined. It is performed by finding an object being simultaneously a representative for both features. The object is positioned in non-empty intersections of lower approximations of *style* and $fluidity$ sets. The algorithm and the graphical representation of the method developed are presented in Fig. 6.

Utilizing presented methodology for all possible combination of requirements $(fluidity, style)$ solutions represented as pairs of $(alpha, beta)$ are found (Table 4). Finally this knowledge is employed to enhance animated sequences. The animation described with (V_m, A_m, t_m) is processed by ANIMATOR system, and based on $(fluidity, style)$ requirements given by the user the $(alpha, beta)$ are calculated (Table 4. This finally results in amplitudes A_a and lengths t_a of additional phases of motion (1) that are intended to be introduced into animation, changing its subjective quality. Mappings obtained are modeled employing fuzzy logic processing [7] [8]. Support points for interpolation surfaces are placed in all values present in Table 4, i.e. triangle fuzzy membership functions are created for *alpha* and *beta*, each having the core point (range of membership value 1) at given values, and 3 triangle functions for $fluidity$ levels and *style* levels (Fig. 7).

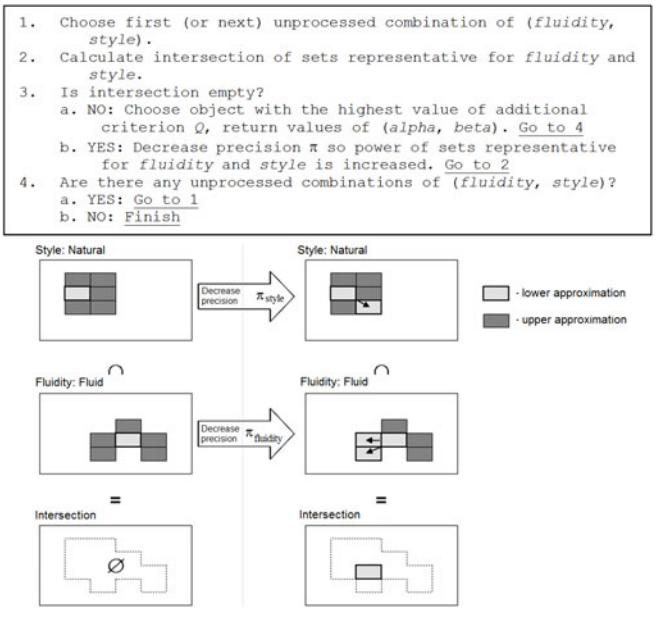


Fig. 6. Pseudocode and the graphical representation of the algorithm searching for non-empty intersections of lower approximations of style and fluidity sets, utilizing variable precision. On the left side the intersection of the lower approximation is empty, therefore precisions are gradually decreased to the levels that result in broadening of the lower approximation (arrows on the right side), and finally non-empty intersection is found.

Table 4. Result mapping between required *fluidity* and *style* of animations and *alpha* and *beta*

alpha		fluidity			beta		fluidity		
		abrupt	medium	fluid			abrupt	medium	fluid
style	natural	0.7	0.5	0.3	style	natural	3	5	7
	medium	0.9	0.7	0.5		medium	1	5	5
	exaggerated	1.3	1.1	0.9		exaggerated	3	5	7

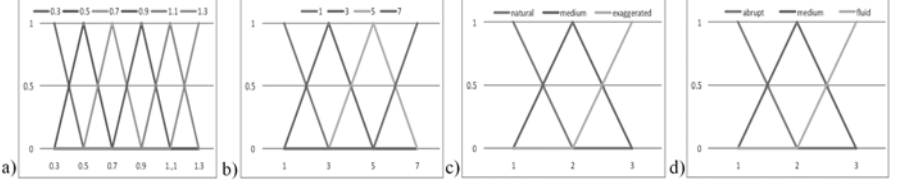


Fig. 7. Membership functions for: (a) *alpha*, (b) *beta*, (c) *style*, (d) *fluidity*

Utilizing fuzzy modeling instead of simple rules presented in Table 4, gives an important possibility to use continuous input values (any degree of *fluidity* and *style*), and then output values could be interpolated accordingly. Moreover fuzzy processing and knowledge representation as fuzzy rules are simple and intuitive, and susceptible for editing by hand (if needed by the user) [8].

4 Results

Created animation assisting ANIMATOR system was tested utilizing simple animations of 5 actions meant for processing with different requirements. Result animations were evaluated by a group of viewers. Obtained results are presented in Fig. 8. Style variations do not influence perceived quality of motion, and the processed animations are always rated higher than original, therefore the ANIMATOR system can be successfully utilized for creation of the high quality animation of virtual characters, featuring different styles of motion.

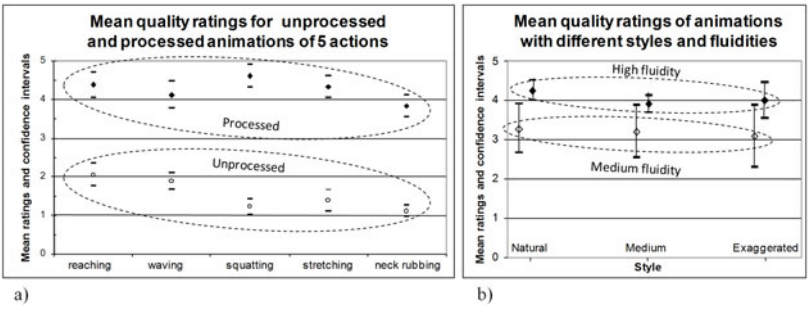


Fig. 8. Ratings of result animation quality: unprocessed and processed animations of five actions: (a) processed animations have significantly higher quality; (b) style variations do not influence quality

The meaning of the result action is generally unchanged, while its quality and style are improved. The method can be used for rapid animation development, and for avatar animation, where predefined animations can be used, but with varying style and fluidity, best matching the avatar's personality [9].

5 Conclusions

The method for processing ambiguous data by utilization of the variable precision rough set was proposed. It was employed for the creation of mapping between two decisions and attributes of objects best representing particular classes: animated motion characterized by subjective ratings of style and fluidity. In future work more decisions can be used simultaneously, for example age and gender of the animated character. Also the list of animation attributes can be extended to other than pose-to-pose approach, eventually leading to the higher accuracy of set approximations, and result quality.

References

1. Menache, A.: Understanding Motion Capture for Computer Animation and Video Games. Morgan Kaufmann, San Francisco (1999)
2. Williams, R.: The Animator's Survival Kit: A Manual of Methods, Principles, and Formulas for Classical, Computer, Games, Stop Motion, and Internet Animators. Faber & Faber, London (2002)
3. Thomas, F., Johnston, O.: Disney Animation The Illusion of Life. Abbeville Press, New York (1981)
4. Whitaker, H., Halas, J.: Timing for animation. Focal Press, Oxford (2002)
5. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough Sets: A Tutorial. In: Pal, S.K., Skowron, A. (eds.) Rough Fuzzy Hybridization: A New Trend in Decision-Making, pp. 3–98. Springer, Singapore (1999)
6. Ziarko, W.: Variable precision Rough Set Model. J. of Computer and System Sciences 46, 39–59 (1993)
7. Pedrycz, W.: Fuzzy control and fuzzy systems. Wiley, New York (1993)
8. Zadeh, L.A.: Fuzzy Logic = Computing with Words. IEEE Transactions on Fuzzy Systems 4, 103–111 (1996)
9. Szczuko, P., Kostek, B., Czyewski, A.: New method for personalization of avatar animation. In: Cyran, K.A. (ed.) Man-Machine Interactions. ASIC, vol. 59, pp. 435–443. Springer, Heidelberg (2009)

Adaptive Phoneme Alignment Based on Rough Set Theory

Konstantinos Avdelidis[1], Charalampos Dimoulas[1], George Kalliris[2], and George Papanikolaou[1]

[1] Aristotle University of Thessaloniki, School of Engineering, Dept. of Electrical and Computer Engineering, University Campus, 54124 Thessaloniki, Greece
[2] Aristotle University of Thessaloniki, Dept. of Journalism and Mass Communication, University Campus, 54124 Thessaloniki, Greece
avdel@auth.gr, babis@eng.auth.gr, gkal@jour.auth.gr, pap@eng.auth.gr

Abstract. The current work describes a phoneme matching algorithm based on rough set concepts. The objective of this type of algorithms is focused on the localization of the phonemic content of a specific spoken occurrence. According to the proposed algorithm, a number of rough sets containing the multiple expected phonemic instances in a sequence are created, each defined by a set of short term frames of the voice signal. The properties of the corresponding information system are derived from a features set calculated from the speech signal upon initiation. Given the above, an iterative procedure is applied by updating the phoneme instances versus the optimization of the accuracy metric. The main advantage of this algorithm is the absence of a training phase allowing for wider speaker adaptability and independency. The current paper focuses on the feasibility of the task as this work is still in early research stage.

Keywords: phoneme alignment, phoneme matching, phoneme segmentation, rough sets, audio features, phonemic sequence.

1 Introduction

Speech recognition (SR) is an active research field whereas significant research has been conducted over the last decades. Nowadays, computer technology continuous evolution in combination with the availability of broadly-used signal processing software tools made possible the implementation of efficient SR algorithms, with remarkable accuracy and speed. We may distinguish a broad area of application that are related to SR, including but not limited to human-machine interaction, adaptive speech to text, lip-sync application, phoneme to viseme conversion for realistic digital characters speech emulation, audio and audiovisual content management and summarization, and others [1], [2]. Among SR- sub-fields we may point out speech and voicing detection-segmentation, phoneme recognition and matching, the latter one being also the main topic of the current work.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 100–109, 2010.

A specific category of applications in the field of speech processing which responds to the matching of an expected with a spoken content is referred to as speech alignment. The primary targeting of this kind of applications is focused on the matching of the fundamental speech units described as phonemes. The task of the so called Phoneme Alignment (PhA) (referred to also as segmentation or matching), is the proper positioning of a sequence of phonemes in relation to a corresponding continuous speech signal. The outcome of this procedure is particularly useful in a variety of applications such as ASR and text-to-speech applications [3] as well as health-related treatment of speech [4], [5], [6]. The majority of the proposed approaches are based on Hidden Markov Modeling (HMM) [7], [8], [9] whereas a few implementations involving Support Vector Machine (SVM) based discriminative learning [10] and hybrid approaches (HMM- SVM) [11] have also been proposed.

Rough set, introduced by Pawlak in 1982 [12], is an alternative Knowledge Discovery and Data mining (KDD) approach, that has been utilized in many scientific fields. Among them methodologies for speech processing [13], speech recognition [14], recognition of isolated words [15] and singing voice recognition [16] have been proposed during the last years. Among the RS advantages, the unsupervised training capability is very important in SR /PhA applications, considering that a representative ground-truth training set is rather difficult to be obtained, incorporating all the potentially encountered conditions (i.e. background noise variations, speaker differentiation, multi-language large vocabulary, etc).

In the current paper, a theoretical analysis towards a speaker-independent PhA implementation based on rough-set (RS) theory is presented. The primary objectives of this approach is the absence of a training phase, in contrast to the proposed HMM/SVM methods, as well as the automated adaptability in personalized speech patterns.

1.1 Problem Definition

As already mentioned, there are various conditions / parameters that have to be considered during phoneme recognition (vocabulary size, speaker dependence, SNR conditions, read / spontaneous speech, language-specific characteristics and multilingual aspects, etc) [17], [18]. The primary concern of the current work is to develop an accurate RS-based phoneme matching system to be used in real-world studio recording condition (with acceptable SNR), both for spontaneous and read speech (i.e. news casting, presentation speeches, radio production, etc). In addition, the main focus of the implemented system is to support Greek language phoneme recognition, taking advantage of the available phonetic-sequence information that has been extracted via text processing [2]. It is obvious that the text-acquired phonetic information provides only an expectation of the real phonemes-sequence. In fact, certain difficulties arise due to the different speech tempo of the involved speakers, the potential phoneme miss-articulation, the appearance of different-length pauses in real world conversation, and others [18]. Hence, the goal of the proposed RS-based module is to estimate the missing phonetic-timing, aligning the textual phonetic information with the corresponding sound signal.

The advantage of the proposed approach is due to the fact that no training data is required. In fact, it is quite demanding to prepare phoneme-tagged audio sequences in

order to train / validate an expert Greek-phoneme matching algorithm. Instead of that, proper windowed-audio extracted-features can easily form an attributes-table to facilitate RS-based phoneme alignment, given the predefined phonetic sequence that is extracted from the text processing module. Based on the above analysis, the proposed architecture involves: a) text-processing phonetic information extraction, b) audio pre-processing (i.e. windowing and salient feature extraction and quantization), c) objects-attributes-table formation and RS reasoning, d) iterative phoneme range adaptation until the termination criterion is met. Additionally, the proposed framework considers single-speaker audio sequences are processed. In real world dialogue application and conversation systems this can be faced employing speaker-related audio segmentation prior to PhA [19].

The PhA task can in fact be formulated using the following definition. If P is the set containing the phonemes of a given language and X is a set of short term frame features (STF) of the signal then:

$$\begin{gathered} \bar{x} = (x_1, x_2, ...x_M), x_i \in X \\ \bar{p} = (p_1, p_2, ...p_N), p_i \in P \\ f_{PhA}(\bar{x}, \bar{p}) \to \bar{t} = (t_i | t_0 = 0, i \in [1, N], t_i \in [1, M]) \end{gathered} \tag{1}$$

Notations $\bar{x}$ and $\bar{p}$ characterize the STF and phonemic sequences which via the f_{PhA} produce a sequence of markers $\bar{t}$ indicating the start point of each phoneme, with N the phonemic and M the STF sequence lengths respectively. Obviously, each expected phoneme i in the sequence is implied in the discrete short term time region $[t_i, t_{i+1} - 1]$ [10].

The definition of the estimator function f_{PhA} which in fact indicates the actual alignment is not trivial due to the variations of the phonetically identical patterns among different speakers. Moreover, speech disorders as well as environmental factors (such as presence of noise) may render the above task more difficult to accomplish [18]. Based on these observations, a local version of the estimator able to establish relativity constrains among the different versions of the same phoneme in the utterance under investigation could eliminate some of the deteriorating factors.

Before continuing further, a formulation must be set in order to treat the above defined sequences as sets. Each sequence can be considered that it contains ordered instances of a certain class of objects. Thus two set views of a sequence are defined along with respective operators which for the case of sequence $\bar{p}$ are:

$$\begin{gathered} \bar{p} = (p_1, p_2, ...p_N) \\ p = instance(\bar{p}) = \{p_i | p_i \in \bar{p}, i \in [1, N]\} \\ \hat{p} = class(\bar{p}) = \{\hat{p}_i | \hat{p}_i \in P, i \in [1, N']\}, N' \le N \end{gathered} \tag{2}$$

In pursue of this goal, the following can be defined according to (2):

$$\begin{gathered} X_i = \{x_t | x_t \in x, t \in [t_i, t_{i+1} - 1], i \in [0, N]\} \\ T_i = \{t | t \in [t_{i-1}, t_i - 1], i \in [0, N]\} \\ \mathrm{T}[X_i] \to T_i \\ P_j = \{p | \hat{p}_j \in \hat{p}\} \\ R_j = \bigcup_{p_i \in P_j} X_i \end{gathered} \tag{3}$$

In brief, the sets X_i contain the SFT instances occurring in the region of appearance of each expected phoneme instance i in the sequence and T_i the respective STF discrete timing indexes which can be mapped to a range in the original signal. It is obvious that the accuracy of the mapping is depended on the length and overlapping of the processing window used for the calculation of the STFs. Moreover, the mapping function $\mathrm{T}[X_i]$ is defined as a transformation of the SFT instance set X_i to the respective timing index set T_i, which has a rather straightforward implementation since each instance in sequence $\bar{x}$ is considered unique.

On the other hand, the sets P_j contain all the instances of the phoneme class j appearing in the sequence. Therefore the set R_j contains all the STF instances that are assigned to the phoneme class j. According to the above formalization an alignment quality criterion can the degree of separation among the R_j classes which is obviously affected by the selection of $\bar{t}$.

2 The PHAROS Algorithm

In this section are provided details about the analysis procedure in the Phoneme Alignment using ROugh-Set (PHAROS) algorithm. The analysis is performed considering each STF of the signal as an object with a predefined number of discrete properties. This fact has severe impact in the feature extraction procedure which was especially designed to fit the needs of the current task. These considerations are discussed in perspective of the requirements of rough set analysis, which is used for the subsequent optimization of R_j.

2.1 Feature Extraction

Given the properties of the alignment task as well as the rough sets theory, the input features set must contain an expected sequence of phonemes as well as a number of discrete valued properties.

In order to increase applicability the phonemic sequence is derived from a text input which is automatically converted to SAMPA transcription via an improved version of the rule-based converter described in [2]. The properties set was selected to be based on the Mel-frequency cepstrum coefficients (MFCC). To deal with their continuous nature it is needed to create a derived properties set by means of quantization. In fact, we are seeking for a transform from the original STF space X to a discrete properties space q according to the following relation

$$x=(x_1,x_2,...x_C),x\in X,x_i\in\mathbb{R}\rightarrow q_x=(q_1,q_1,...q_C),q_x\in q,q_i\in[1,K],q_i\in\mathbb{N} \quad (4)$$

In an effort to describe the properties of the quantization, the linear range quantization is not the best choice due to the acute non-linear behavior of speech [15]. Additionally, the fact that the proposed algorithm is designed to operate locally (not performed versus a globally trained system), does not pose any constrains regarding global normalization. Consequently, the quantization procedure of each coefficient was based on the equal range entropy criterion in order to focus the descriptive accuracy on the appropriate value ranges. Since the values of each coefficient are known throughout the input, this criterion is expressed by the following properties:

$$\begin{gathered} x_i=\left\{x_{ij}\,\middle|\, j\in[1,M]\right\},i\in[1,C] \\ \left|\tilde{x}_{i(k)}\right|=\frac{|x_i|}{K},x_i=\bigcup\tilde{x}_{i(k)},\varnothing=\bigcap\tilde{x}_{i(k)} \\ \min\left\{\tilde{x}_{i(0)}\right\}=\min\{x_i\},\max\left\{\tilde{x}_{i(k)}\right\}<\min\left\{\tilde{x}_{i(k+1)}\right\},\max\left\{\tilde{x}_{i(K)}\right\}=\max\{x_i\} \end{gathered} \quad (5)$$

In short, this criterion is met if the feature set x_i of M coefficient values x_{ij} is split in K incompatible sets $\tilde{x}_{i(k)}$ of equal cardinality and incompatible [max, min] ranges, which define the quantization thresholds. This provides a quantization region distribution of equal range probability $1/K$ for the random variable x_i in the context of the specific signal input.

Therefore, the STF sequence was defined to be the MFCCs and the system properties were provided according to the above quantization procedure. The values $K=4$ and $C_{MFCC}=8$ were proved to be an acceptable choice.

2.2 Rough-Set Analysis

The formulation of the rough set analysis was based on the definition of an information system (IS) having M objects (STFs) with C_{MFCC} properties (quantized MFCC coefficients) each represented by K discrete values.

On the above mentioned IS, N' is the number of the phoneme classes appeared in the sequence. Since, in any given occurrence, only a certain number of phoneme classes appear, it is obvious that $N'\leq|P|$. This fact leads us to the conclusion that only the phoneme classes appearing in the sequence will be iterated for instances.

Fig. 1. Information system of the phoneme alignment algorithm. The per phoneme class R_j, per phoneme instance X_i and universal x sets of STF instances are shown. The properties set $q = \{q_i | i \in [1, C_{MFCC}]\}$ has a value assignment for each member of x.

R_j	X_i	x	q_1	q_2	q_3	...	$q_{C_{MFCC}}$
Phoneme class 1	Phoneme instance 1	STF 1				...	
		STF 2				...	
		STF 3				...	
Phoneme class 2	Phoneme instance 2	STF 4				...	
		STF 5				...	
		STF 6				...	
Phoneme class 1	Phoneme instance 3	STF 7				...	
		STF 8				...	
		STF 9				...	
...	...	...	...	...	...	...	...
Phoneme class N'	Phoneme instance N	STF $M-2$				...	
		STF $M-1$				...	
		STF M				...	

Therefore, the R_j sets can be considered in the context of the objects x having the properties set q. Since, $R_j \supset x$ we can define the accuracy of each R_j in the approximation space $A = (x, q)$ as:

$$a_q(R_j) = \frac{\left|\underline{A}(R_j)\right|}{\left|\overline{A}(R_j)\right|} \tag{6}$$

where $\underline{A}(R_j)$ is the lower and $\overline{A}(R_j)$ the upper approximation of R_j in $A = (x, q)$ [20]. By interpretation of the result, this value is inversely representing a quantity of resemblance of the STFs included in a certain phoneme class with the ones excluded from it. In other words, this metric indicates the need to include certain STFs in a phonemic class, resulting to the change of a STF set region of X_i.

In terms of local investigation, conclusions can be drawn by local processing of $\{X_i\}$ partitions. For each member of each P_j, local phoneme instances X_i can be processed separately in a subset of x considering a local approximation space $A' = (x', q)$ consisting of the members of $X_i, i = k$ neighbors. Using (3) for a neighborhood of X_k the following equations apply:

$$\begin{aligned} x' &= \bigcup_{i \in [k-1, k+1]} X_i \\ R_j' &= \bigcup_{\substack{p_i \in P_j \\ i \in [k-1, k+1]}} X_i = X_k \end{aligned} \tag{7}$$

In this case, a phoneme instance is compared to its neighbors for resemblance. The upper approximation $\overline{A}'(X_k)$, provides an indication of the STFs that should be included in the set X_k (or claimed from its neighbors) according to its internal equivalent classes structure. The final update decision must comply with the sequential nature of the STFs contained in X_i by omitting sequential gaps, as shown in Fig. 2.

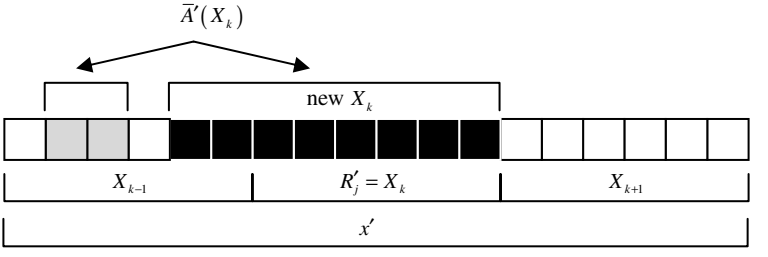


Fig. 2. Local estimation of range change based on the $\overline{A}'(X_k)$, iterated for each member of each P_j

Finally, the properties that optimally represent each phoneme class can be inferred using the R_j reducts set. More specifically the following apply:

$$\begin{gathered} \mathrm{RED}(R_j) = \{q'_1, q'_2, \ldots q'_Q\} \\ R_{(q')j} = [R_j]_{q'_1} = [R_j]_{q'_2} = \ldots = [R_j]_{q'_Q} = \{r_1, r_2, \ldots r_e\} \\ A'_r = (x, q - q'_r) \end{gathered} \tag{8}$$

$$\gamma_{q-q'_r}(R_{(q')j}) = \frac{\sum_{k=1}^{e} |\underline{A}'_i(r_k)|}{|x|} \tag{9}$$

where $\mathrm{RED}(R_j)$ is the reducts set of R_j over the approximation space A, $R_{(q')j}$ the equivalence classes set of R_j (same by definition for every reduct in A), A'_i the approximation space omitting the properties q'_i, $\gamma_{q-q'_i}$ the dependency measure of attribute set q'_i on the $q - q'_i$, $\underline{A}'_i(r_i)$ the lower approximation of the equivalence class r_i on the approximation space A'_i and x the set of all STFs.

This provides a descriptive quality metric of the reducts set versus the remaining of properties of the original attribute set. In fact, the attribute set $q_j = q'_r$ over $\mathrm{RED}(R_j)$ with minimal $\gamma_{q-q'_r}$ is selected as the more appropriate to identify R_j in the approximation space A. The combination of appropriate X_i and q_j result in the minimization of the intersection among R_j which is the requirement posed by (3) as phoneme alignment sanity criterion. Therefore, (8) can be considered as form of objective function.

2.3 Iterative Algorithm

Using the above mentioned theory, an iterative procedure is proposed in order to minimize γ_j of a given $(\overline{x}, \overline{p})$ alignment task. The following optimization procedure is inspired by the alternating optimization (AO), primarily performed on fuzzy clustering [21]. The analogy between fuzzy clustering can be summarized by the fact that an input sequence ($\overline{x}, \overline{p}$) needs to be categorized in a certain amount of prototypes (R_j, q_j) according to a distance metric (δ). The latter can be derived according to (6) and the local neighborhood as defined by (7), using the following equations:

$$A' = \left(x', q_j\right), q_j = \left\{q_j \middle| X_i \cap R_j \neq \varnothing\right\}$$
$$0 \leq a_{q_j}\left(X_i\right) = \frac{\left|\underline{A'}\left(X_i\right)\right|}{\left|\overline{A'}\left(X_i\right)\right|} \leq 1 \tag{10}$$
$$\delta_i = -\log\left(a_{q_j}\left(X_i\right)\right) \Rightarrow \delta = \left[\delta_1, \delta_2, ...\delta_M\right]^T$$

Table 1. The modified AO procedure applied in PHAROS

Let data sequences $\overline{x}, \overline{p}$ be given.
Let each phoneme class be uniquely identified by $\left(R_j, q_j\right)$
Quantize the features set x according to (5)
Define P_j and M' from the phonemic sequence
Initialize $\delta^{(0)}$, $u = 0$, $q_j = q$ and X_i, R_j such as $\left|X_i\right| = 1/M$
Choose a precision for termination ε

REPEAT
  Increase u by 1
  Determine $a_j = \left\{a_q\left(R_j^{(u-1)}\right) \middle| q = q_j^{(u-1)}\right\}$
  FOR each j in ascending order of $\left\{a_j\right\}$
    Determine $X_i^{(u)}$ and $R_j^{(u)}$ according to implications of (7) for fixed $q_j^{(u-1)}$
  NEXT
  FOR each j
    Determine $q_j^{(u)}$ according to implications of (8) and (9)
  NEXT
  FOR each i
    Determine $\delta_i^{(u)}$ according to implications of (10)
  NEXT
UNTIL $\left\|\delta^{(u-1)} - \delta^{(u)}\right\| \leq \varepsilon$

This calculation is performed locally as due to the sequential nature of speech we are only interested in local instance discrimination as well as for complexity reduction reasons. The negative logarithm is used in order for δ to obtain distance-like monotonicity attributes. Therefore, the derivation of the optimization modification is relatively straightforward and is described in Table 1.

The final phonemic alignment is provided according to (1) by the inferred from the final X_i STF index values T_i, which in turn refer to actual signal timing ranges according to the following:

$$f_{PhA}(\bar{x},\bar{p}) \to \bar{t} = \left(\max\left\{\mathrm{T}\left[X_1\right]\right\}, \max\left\{\mathrm{T}\left[X_2\right]\right\}, ..., \max\left\{\mathrm{T}\left[X_M\right]\right\}\right) \Rightarrow$$
$$\bar{t} = \left(\max\left\{T_1\right\}, \max\left\{T_2\right\}, ..., \max\left\{T_M\right\}\right) \tag{11}$$

3 Results and Discussion

The current paper was focused on the theoretical substantiation of the PHAROS algorithm. It should be noted that the above methodology is still in early research stage. Although analytical tests are not yet available, the preliminary development conclusions show that this approach is able provide valuable results. Moreover, for the sake of simplicity, certain aspects of PHAROS such as the phoneme transition states management and the features quantization error provision are not covered in the present work.

It is obvious that the above methodology can be easily extended on speech of different languages by merely modifying the corresponding text-processing phoneme-extraction system functionality. Thus, the proposed strategy can be applied to general phoneme recognition / alignment under various conditions. Potential advantage of the current methodology is the easy deployment for audio phoneme tagging that could be utilized as ground truth for supervised training and performance evaluation of alternative fully automated (not-text assisted) phoneme recognition algorithms.

References

1. Avdelidis, K., Dimoulas, C., Kalliris, G., Bliatsiou, C., Passias, T., Stoitsis, J., Papanikolaou, G.: Multilingual automated digital talking character. In: IBC Convention, Amsterdam, Netherlands (2003)
2. Kalliris, G., Dimoulas, C., Papanikolaou, G., Avdelidis, K., Passias, T., Stoitsis, J.: Phoneme recognition for 3d modeled digital character talking emulation. In: 112th AES Convention, Munich, Germany (2002)
3. Rabiner, L., Juang, B.: Fundamentals of Speech Recognition. Prentice Hall PTR, Englewood Cliffs (1993)
4. Gordon-Salant, S., Yeni-Komshian, G.H., Fitzgibbons, P.J., Barrett, J.: Age-related differences in identification and discrimination of temporal cues in speech segments. The Journal of the Acoustical Society of America 119, 2455 (2006)
5. Kain, A.B., Hosom, J., Niu, X., van Santen, J.P., Fried-Oken, M., Staehely, J.: Improving the intelligibility of dysarthric speech. Speech Communication 49, 743–759 (2007)

6. Shriberg, L.D., Green, J.R., Campbell, T.F., Mcsweeny, J.L., Scheer, A.R.: A diagnostic marker for childhood apraxia of speech: the coefficient of variation ratio. Clinical Linguistics & Phonetics 17, 575–595 (2003)
7. Brugnara, F., Falavigna, D., Omologo, M.: Automatic segmentation and labeling of speech based on Hidden Markov Models. Speech Communication 12, 357–370 (1993)
8. Hosom, J.: Speaker-independent phoneme alignment using transition-dependent states. Speech Communication 51, 352–368 (2009)
9. Mporas, I., Ganchev, T., Fakotakis, N.: Speech segmentation using regression fusion of boundary predictions. Computer Speech & Language 24, 273–288 (2010)
10. Keshet, J., Shalev-shwartz, S.: Phoneme Alignment Based on Discriminative Learning Abstract, Lisbon (2005)
11. Lo, H., Wang, H.: Phonetic boundary refinement using support vector machine. In: Proc. ICASSP, pp. 933–936 (2007)
12. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Science 11, 341–356 (1982)
13. Czyzewski, A., Kaczmarek, A., Kostek, B.: Intelligent Processing of Stuttered Speech. Journal of Intelligent Information Systems 21, 143–171 (2003)
14. Brindle, D., Ziarko, W.: Experiments with rough sets approach to speech recognition. In: Raś, Z.W., Skowron, A. (eds.) ISMIS 1999. LNCS, vol. 1609. Springer, Heidelberg (1999)
15. Czyzewski, A.: Speaker-independent recognition of isolated words using rough sets. Information Sciences 104, 3–14 (1998)
16. Żwan, P., Szczuko, P., Kostek, B., Czyżewski, A.: Automatic Singing Voice Recognition Employing Neural Networks and Rough Sets. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 793–802. Springer, Heidelberg (2007)
17. Aubert, X.L.: An overview of decoding techniques for large vocabulary continuous speech recognition. Computer Speech & Language 16, 89–114 (2002)
18. Deng, L.: Acoustic Modeling in Automatic Speech Recognition Overview of Current State and Research Challenges. In: Human and Machine Learning, Irvine, California (2009)
19. Kinnunen, T., Li, H.: An overview of text-independent speaker recognition: From features to supervectors. Speech Communication 52, 12–40 (2010)
20. Pawlak, Z.: Rough sets: theoretical aspects of reasoning about data. Kluwer Academic Publishers, Dordrecht (1991)
21. Bezdek, J.C., Hathaway, R.J., Pal, N.R.: Norm-induced shell-prototypes (NISP) clustering. Neural, Parallel Sci. Comput. 3, 431–449 (1995)

Monitoring Parkinson's Disease Patients Employing Biometric Sensors and Rule-Based Data Processing

Paweł Żwan, Katarzyna Kaszuba, and Bożena Kostek

Gdansk University of Technology, Narutowicza 11/12, 80-233 Gdansk, Poland
{zwan,katkasz,bozenka}@sound.eti.pg.gda.pl

Abstract. The paper presents how rule-based processing can be applied to automatically evaluate the motor state of Parkinson's Disease patients. Automatic monitoring of patients by using biometric sensors can provide assessment of the Parkinson's Disease symptoms. All data on PD patients' state are compared to historical data stored in the database and then a rule-based decision is applied to assess the overall illness state. The training procedure based on doctors' questionnaires is presented. These data constitute the input of several rule-based classifiers. It has been proved that the rough-set-based algorithm can be very suitable for automatic assessment of the PD patient's stability/worsening state.

Keywords: Rough sets, Rule-based processing, Parkinson's Disease, decision systems, automatic assessment of the patient's motor state.

1 Introduction

Parkinson's disease (PD) is a common neurodegenerative disease which belongs to the group of conditions called movement disorders. In advanced stages this disease is related to the symptoms such as motor disability, dyskinesias, freezing, falls and on-off states which disable normal patients' lives. Since most of these symptoms are related to movements or fluctuations of normal movements they can be objectively measured. One of the ways of measuring the patient's motor activity is to use accelerometers similarly to other applications in the health domain [1], [2], [3], [4], [5]. This possibility can solve one of the important problems of neurogenerative illnesses – the lack of objective measuring methods in the examination and monitoring of PD patients. This is especially important because the number of PD disease patients is continuously increasing and the number of specialists is still insufficient. Moreover, in most cases, the patients are elderly people who are not able to frequently visit doctors in their clinics. A continuous monitoring of a PD patient by using biometric sensors (i.e. accelerometers) can solve these problems. The PERFORM (A so**P**histicat**E**d multi-pa**R**ametric system **FOR** the continuous effective assessment and **M**onitoring of motor status in Parkinson's disease (PD) and other neurodegenerative diseases progression and optimizing patients' quality of life) is the FP7 European project aimed at

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 110–119, 2010.

automatic assessment of the PD symptoms [6], [7]. The objective of this project is to build a Personal Health System which enables a continuous patient's monitoring in home. In this solution symptoms of the disease can be automatically analyzed and results of the analysis can be sent to doctors. Additionally, some intelligent processing methods can analyze the changes in the symptoms and assess the overall progression of the disease in the Unified Parkinson Disease Rating Scale (UPDRS).

In the paper the application of rough sets and other selected rule-based decision systems in this domain is presented. The algorithms are trained with the use of patient's historical data which are assessed by medical doctors. The training examples are processed by algorithms and sets of rules are generated. Results obtained for all the algorithms examined are compared and conclusions are derived.

2 Methodology

The PERFORM system uses accelerometers that are attached to the body of a patient, as presented in Fig. 1. Signals from the accelerometers (3 axes, 6 accelerometers) are processed and automatically analyzed in order to calculate UPDRS descriptions. These descriptions are integer numbers: $\{0, 1, 2, 3, 4\}$ where 0 is related to the lack of a given PD symptom and 4 is related to its most severe state. A detailed description of these UPDRS rates is presented in literature related to PD patients' assessment [7], [8]. The UPDRS ratings that can be automatically calculated in the PERFORM are listed in Table 1.

In the next step, these UPDRS scores are analyzed by the learning algorithm which can support doctors in their decisions. This additional processing can be particularly important in the case when a medical doctor is monitoring many patients at the same time. He can be notified in the case of the worsening of the state of one of the patients. Moreover, this support seems especially important when a doctor who is not specialist in the neurology is treating a PD patient.

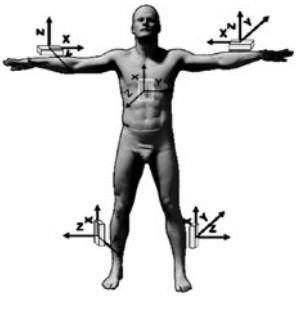


Fig. 1. Placement of the accelerometer sensors on the patient's body [9]

Table 1. UPDRS symptoms in PERFORM

UPDRS	Description
$UPDRS_{13}$	Falling (unrelated to freezing)
$UPDRS_{14}$	Freezing when walking
$UPDRS_{20RH,LH,RL,LL}$	Tremor at rest calculated separately for both hands and legs
$UPDRS_{21RH,LH}$	Action or postural tremor of right and left hand
$UPDRS_{23RH,LH}$	Finger tapping test calculated for right and left hand
$UPDRS_{24RH,LH}$	Movement test calculated for right and left hand
$UPDRS_{25RH,LH}$	Alternating movement test calculated for right and left hand
$UPDRS_{28}$	Posture test
$UPDRS_{29}$	Gait
$UPDRS_{31}$	Body Bradykinesia and Hypokinesia
$UPDRS_{32}$	Duration of Dyskinesias
$UPDRS_{33}$	Disability of Dyskinesias
$UPDRS_{39}$	On/Off Proportion

The evaluation of the stability/worsening of a PD patient's state is based on processing current and historical UPDRS values of symptoms with the use of a learning algorithm. In order to perform training of these algorithms a doctor's decision is needed to contain the knowledge how to translate changes of the UPDRS scores into worsening/stability assessment. The scheme of the methodology used is presented in Fig. 2. The training of the system is performed utilizing PD subjects' historical UPDRS data. These patients are examined by the doctors and current UPDRS results are acquired. Current and historical UPDRS data are then assessed by the doctors who evaluate the PD progress in each of cases. As a result a set of rules is generated. These rules are then used in the PERFORM system, where current UPDRS symptoms (automatically assessed) can be compared with historical ones stored in the PERFORM database. As a result an automatic assessment on worsening/stability can be generated for the monitored PD subject. This type of a decision system mimics the way of the

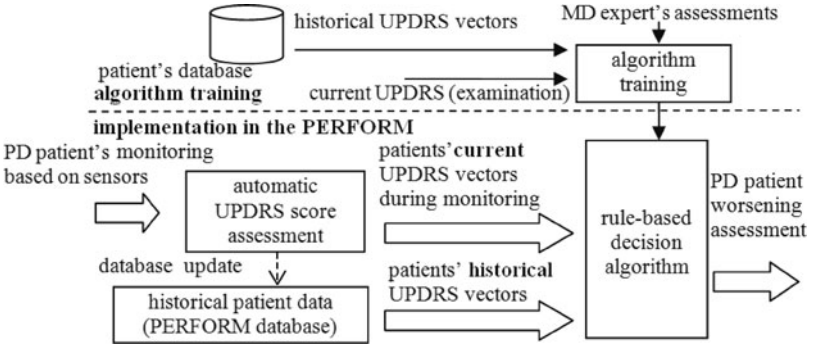


Fig. 2. General scheme of the automatic assessment of the worsening of PD disease

doctors' thinking while assessing the progress of a Parkinson Disease by repeatedly performing the UPDRS test.

2.1 Algorithms

The presented problem can be described as a search for an optimal function projecting the set of input attributes into a decision. Input attributes A (composed of differences in UPDRS rates for all the symptoms) along with the decision attribute d (experts' evaluations) are given for the training (Eq. 1). U in Eq. 1 is the Universe – possible values of the difference in UPDRS, its domain are integer values $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$. These values are positive if the value of a given UPDRS rate increased, negative if its value decreased.

$$A = (U,\ A \cup \{d\}) \tag{1}$$

This set of attributes is processed by the algorithms presented below.

There exist several methods of automatic decision systems in medical applications that are presented in the literature ([10], [11]). In the automatic assessment of the PD state worsening/stability the following algorithms are compared: Rough Sets (RS) [12], [13], [14], Rough Sets with rules generalization algorithms (RS-g), Repeated Incremental Pruning method (RIPPER) [15], Nearest Neighbour algorithm (NN) [16], PART Decision List with two sets of coefficients (PART) [17] and Ripple Down Rule method (RDR) [18], [19]. Since the algorithms mentioned here are widely utilized in data mining fields, only their usefulness in the context of data gathered and rules generated will be shortly discussed.

The usefulness of the rough sets in this application needs to be stressed since this technique is widely used in data mining [12], [13], [20], [21]. Moreover, in the case of the automatic assessment of the stability of a PD patient, this approach seems very appropriate because of the inconsistent character of the training data [14], [19] (doctors may assess a given medical case differently). However, initially the number of rules generated on the basis of the training objects is very high therefore the expert knowledge enclosed in them can be difficult to interpret. This drawback can be minimized if discretization methods are introduced in rough sets. These methods cause that training objects are automatically gathered with regard to the values of the attributes. As a result the domain of each attribute is divided by using the minimal set of cut points. In the global discretization methods the Universe U is divided equally for each of attributes, in the local discretization the Universe is divided independently for each of attributes. As a result a number of attributes is diminished and the cut-points are calculated. This information is used for the formation of decision rules.

Rules obtained in the processing cover all training data. This capacity has a very significant drawback – a loss of generalization capacities (similarly as the over training in case of neural networks [22]). To avoid this problem algorithms of generalization are introduced in this study. The strength of the generalization is controlled by a coefficient g which is number from $(0 \leq g \leq 1)$. This coefficient

determines how aggressive the procedure should be. The coefficient equals 1.0 means that the generalization must preserve the precision level of the original rules. For coefficients closer to zero, generalization may cause rules to lose precision for the sake of greater generalization capabilities [8]. The rough set using this type of processing will be denoted as RS-g classifier to distinguish between rough sets without additional generalization post processing (RS classifier).

As the PERFORM system needs to process huge amount of noisy data it is crucial to investigate algorithms that can compute them in a relatively short time. The Repeated Incremental Pruning to Produce Error Reduction (RIPER) is chosen to be examined as it is often dedicated for such tasks [15]. The RIPPER algorithm has a capability of dealing with missing data, which may be essential in case of UPDRS patients' diagnosing. It also computes quickly large amount of noisy data and usually produces small error rate. Still if there are contradictory data in the training set the RIPPER fails to produce classification.

The Nearest Neighbor with Non-Nested Generalization (NNge) algorithm using a lazy learning method (k-nearest-neighbor) was chosen from a wide range of WEKA system methods. Also this is the only solution examined that produces generalization schemes – non-nested hyperrectangle. The main disadvantage of this algorithm is however that to produce a hyperractangle all features from the set must be used. This results with a very high number of rules, which are hard to be validated by doctors.

The PART decision list (PART) was chosen to be examined as a method known for its high efficiency. It is also a method which deals with the problem of over pruning the rule set, as it does not execute the rule set simplification. The PART algorithm has been proved to give high classification efficiency together with a great flexibility and satisfying computation speed. Still using separate-and-conquer principle does not guarantee an ability to execute many rules at the same time, therefore no contradictory data can be classified. In cases of the UPDRS classification where often even slight change in one attribute can cause significant change in a diagnosis, thus this method is not very suitable for doctors.

The Ripple Down Rule classifier (RDR) represents an expert system (also in medical domain [20]), thus it is crucial that its structure is regularly validated by an independent expert. Such approach allows updating knowledge and may result in a high accuracy of classification. The RDR system is a high quality method to store big amount of knowledge. It is also immune to missing values or missing classes. However it fails when multiple outputs must be given. This method also requires data preprocessing – raw data must be divided into sets of certain ranges – no raw data processing can be performed.

3 Knowledge Building and Rule Generation

In order to obtain the medical knowledge historical UPDRS data of 47 patients (24 males and 23 females, the average age of the patients was 68.2 y.) from St. Adalbert Hospital in Gdansk, Poland were used. The average illness time was

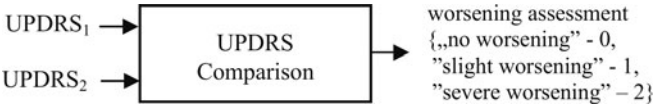


Fig. 3. The scheme of the module comparing two UPDRS scores

9 years with the standard deviation of 5 years. Time periods for the historical UPDRS examination as compared to current examination were 8 months in average with the variance of 7 months. The patients have been assessed by 4 doctors experienced in the UPDRS.

Since for some of these patients an additional historical examination has been available, overall 71 pairs of UPDRS evaluations between 'current UPDRS' and 'historical UPDRS' have been used. These pairs of the UPDRS vectors (containing only 21 items that are monitored in the PERFORM) were presented to four medical doctors and each of them assessed the stability/worsening state of the patient by using of following decision scale: "0" – no worsening, "1" – slight worsening, "2" – severe worsening (Fig. 3). Definitions of "slight" and "severe" have been discussed with the doctors and related to the alert level that should be raised in the PERFORM. Slight worsening is related to a low priority alert (warning in the system), severe worsening is when a high priority alert (alarm) should be risen. In this way each of these UPDRS pairs has been assigned to three output classes: 0, 1, 2, therefore the obtained data could be used directly to train decision systems. The doctors were also instructed not only to sum up the UPDRS points but to consider the importance of each of the symptoms in the assessment of an alert. In this sense rules acquired by the intelligent system reflect the importance of each of UPDRS inputs in the overall evaluation.

3.1 Rule Generation

As mentioned before, pairs of UPDRS (current and historical) have been assessed by 4 experts, therefore the training set consisted of 284 training objects. All samples were divided into training and testing sets and classifiers based on rules extracted for the training set have been used. The testing set was used to verify generalization qualities of these classifiers. The training and testing sets were firstly divided in proportion 50 : 50, 142 training objects contained in each set.

Examples of rules calculated by the rough set algorithm are presented below:

If $(\Delta\text{UPDRS}_{13} < 1)$ **&** $(-1 < \Delta\text{UPDRS}_{14} < 2)$ **&** $(\Delta\text{UPDRS}_{23\text{RH}} < 1)$ **&** $(\Delta\text{UPDRS}_{29} < 1) \Rightarrow (\text{output} = \{\mathbf{1} - \text{'warning'}\})$.

If $(\Delta\text{UPDRS}_{20\text{LH}} > 2)$ **&** $(\Delta\text{UPDRS}_{23\text{RH}} > 3) \Rightarrow (\text{output} = \{\mathbf{2} - \text{'alarm'}\})$.

The rule antecedent prepositions are the changes in UPDRS ratings for a given symptom. This rule gives the doctor information which UPDRS values have been used along with critical differences in the UPDRS values. In this context rules can be directly interpreted by a doctor. The experiments have been carried out in the RSES environment for rough sets [8] and in WEKA tool for other methods [23].

Table 2. Results of classification on training and testing data

	training data						testing data					
[%]	RS	RS-g	RIPPER	NN	PART1	RDR	RS	RS-g	RIPPER	NN	PART1	RDR
case 1:	**80**	74	73	78	77	74	54	**62**	52	54	51	51
case 2:	**89**	80	82	86	86	82	73	77	63	72	71	70
case 3:	88	**92**	89	89	89	91	75	77	**82**	80	71	72

The algorithms utilized generated the following number of rules, i.e. RS – 240 rules, RS-g – 180 rules, PART – 17 rules, RDR – 13 rules, NN – 34 rules, RIPPER – 5 rules. The efficiency of each classifier was tested first on training set and then using testing data. In the first case the ability to cover the training data efficiently has been investigated, in the second case the generalization qualities of the classifiers have been tested. The efficiency of the classification has been calculated using a confusion matrix, i.e. presenting separately the recognition results for each of the recognized categories. Three types of the efficiency of the classification has been defined for the analysis: **case 1:** when the efficiency is calculated for distinguishing between all **3 classes: {0, 1, 2}**, **case 2:** when the efficiency is calculated in relation to discrimination of a high priority alert **{{0}∪{1}, {2}}** and **case 3:** in relation to recognition of the patient's stability **{{0}, {1}∪{2}}**. The results for the classification of training and testing data for these three cases are presented in Table 2.

Since the training data are inconsistent or even conflicting (it is a subjective evaluation performed by doctors), the accuracy of 100% cannot be achieved. The maximum accuracy that can be achieved by this system in this case is related to the coverage of the training examples (accuracy of recognition of the training data). The analysis of the results presented in Table 2 can lead to the following observations:

1. The best accuracies for the training set are: **80%** for the case 1; **89%** for the case 2; and **92%** for the case 3. In all cases the best accuracy has been achieved by the RS classifier. It proves that rough sets have the best coverage of training data in the group of presented classifiers.
2. The RS classifier with generalization algorithms obtains a better accuracy on the testing data and a worse accuracy on the testing data as compared to the RS classifier without generalization implemented.
3. The accuracy achieved for the testing data (not used in the training) is **62%** in the first case, **77%** in the second case, and **84%** in the third case. In cases 1 and 2 the best results are achieved for the RS-g classifier. Only in case 3 slightly better results have been achieved by RIPPER and NN algorithms.
4. The remaining algorithms (RIPPER, NN, PART, RDR) show in most cases significantly worse results as compared to RS-g.
5. The set of rules for the PART consists of only 17 rules, their analysis can be easier for doctors but the classifier is performing well only for the training data with low generalization capacities. This excludes this classifier from the application in the presented system.

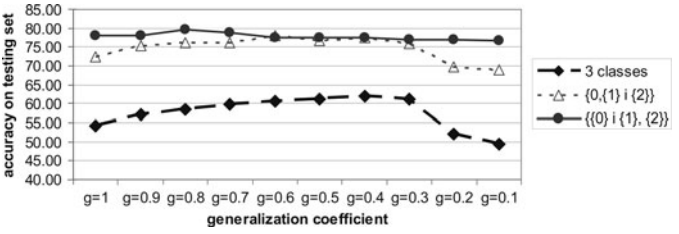


Fig. 4. Recognition accuracy on testing data for different values of generalization coefficient

The observations presented justify the choice of rough sets with the generalization algorithm (RS-g) for the application in the medical system which automatically assesses a progression of the Parkinson Disease patient's state. Since this algorithm proved itself to be the best for such tasks, its settings have been further analyzed. The optimal value of the generalization coefficient has been selected on the basis of the detailed examination of the accuracy for various parameter values. It was changed from values 0.1 to 1 and the accuracy has been calculated in each case. The results are presented in Fig. 4.

4 Data Processing

The set of rules generated by the rough set algorithm has been directly used in the PERFORM. The algorithm compares the current UPDRS value with a closest value in the past by calculating differences of the UPDRS values (Fig. 5). If rule k is activated with a given output $o = \{0, 1, 2\}$ and is covered with s_k^o training examples (strength) than the overall output d_{out} value based on all activated rules can be defined as follows:

$$d_{out} = \frac{\sum_{k=1}^{K} \left(s_k^0 \cdot 1 + s_k^1 \cdot 2 + s_k^2 \cdot 3\right)}{\sum_{k=1}^{K} \left(s_k^0 + s_k^1 + s_k^2\right)} - 1 \tag{2}$$

where K is the total number of the rules activated, d_{out} is the output decision given as float number (0,1).

The final decision is based on the following principles:

out = "no warning" if $d_{out} \subset (0, 2/3)$;

out = "low priority alert" if $d_{out} \subset \langle 2/3, 4/3)$;

out = "high priority alert" if $d_{out} \subset \langle 4/3, 2)$.

If the comparison doesn't trigger off any alert, the UPDRS vectors generated in the past are compared with the current UPDRS value till an alert is found or till the *dayLimit* value is reached (Fig. 5). This procedure is used separately for a low and high priority alerts. As resulted from the processing presented the doctors are given the alert along with information regarding UPRDRS symptom for which the alert has been generated.

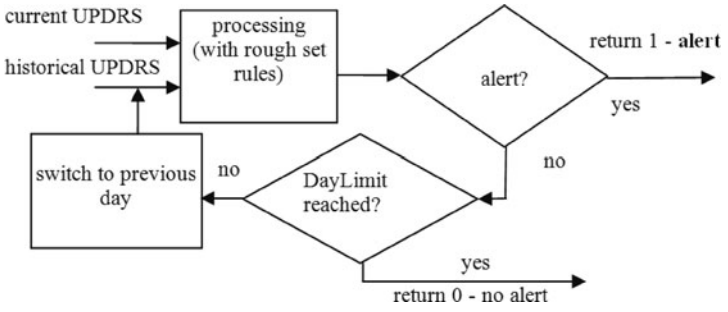


Fig. 5. Generation of alerts in PERFORM

5 Conclusions

The results presented show that an automatic system assessing the progression of the PD patient's state can be implemented. The accelerometers attached to the patient's body can record motor symptoms of the illness. These symptoms can be automatically transformed to the UPDRS values. The aim of the paper was to present the evaluation and functioning of a decision system assessing the overall stability/worsening of the patients. Its functioning is based on analyzing changes of these UPDRS symptoms with the use of a rule-based system. Comparison of several rule-based systems has been performed in this study and it has been proved that rough sets are very suitable for this application. Rough sets with generalization achieved the highest recognition results within a group of classifier analyzed. The accuracy achieved by this classifier on a testing set seems not very high, but the corresponding confusion matrix has been calculated with classifiers trained on inconsistent subjective data.

The presented solution can directly be used in the PERFORM system for monitoring PD patients and in the clinics as a tool to help the doctors with the diagnosis of the overall state of the PD patient.

Acknowledgments. This research is subsidized by the European Commission within FP7 "PERFORM" project, No. 215952. All trials and investigations have been approved by the Ethical Committee of Gdansk Medical University, PL.

References

1. Aminian, K., Najafi, B.: Capturing human motion using body-fixed sensors: outdoor measurement and clinical applications. Computer Animation and Virtual Worlds 15, 79–94 (2004)
2. Veltink, P.H., Bussmann, H.B.J., de Vries, W., Martens, W.L.J., van Lummel, R.C.: Detection of static and dynamic activities using uniaxial accelerometers. IEEE Trans. Rehab. Eng. 4(4), 375–386 (1996)
3. Wetzler, M., Borderies, J.R., Bigaignon, O., Guillo, P., Gosse, P.: Validation of a two-axis accelerometer for monitoring patient activity during blood pressure or ecg holter monitoring. Clinical and Pathological Studies (2003)

4. Mantyjarvi, J., Himberg, J., Seppanen, T.: Recognizing human motion with multiple acceleration sensors. IEEE International Conf. on Sys. Man and Cybernetics 2, 747–752 (2001)
5. Randell, C., Muller, H.: Context awareness by analysing accelerometer data. In: IEEE International Symposium on Wearable Comp., pp. 175–176 (2000)
6. Baga, D., Fotiadis, D.I., Konitsiotis, S., Maziewski, P., Greenlaw, R., Chaloglou, D., Arrendondo, M.T., Robledo, M.G., Pastor, M.A.: PERFORM: Personalised Disease Management for Chronic Neurodegenerative Diseases: The Parkinson's Disease and Amyotrophic lateral Sclerosis Cases. In: eChallenges e-2009 Conference, Istanbul, Turkey, October 21-23 (2009)
7. The Unified Parkinson's Disease Rating Scale (UPDRS): Status and Recommendations. State of the Art Review, Movement Disorders 18(7), 738–750 (2003)
8. Goetz, C.G., et al.: Movement Disorder Society-sponsored revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS): Process, format, and clinimetric testing plan. Movement Disorders 22(1), 41–47 (2007)
9. Maziewski, P., Kupryjanow, A., Kaszuba, K., Czyżewski, A.: Accelerometer Signal Pre-processing Influence on Human Activity Recognition. In: 13th IEEE NTAV/SPA Conference, Poznan, Poland, September 24-26, pp. 95–99 (2009)
10. Tsumoto, S.: Mining diagnostic rules from clinical databases using rough sets and medical diagnostic model. Information Sciences: International J. 162(1), 65–80 (2004)
11. Ryutaro, I., Masayuki, N.: Knowledge Discovery from Medical Database with Multi-Strategy Approach. SIG-FAI J. 51, 31–36 (2003)
12. Leondes, C.T. (ed.): Knowledge-based systems. Academic Press, London (2000)
13. Rough Set Exploration System, Skowron A. – Project Supervisor, http://logic.mimuw.edu.pl/~rses/
14. Wong, S.K.M., Ziarko, W., Li Ye, R.: Comparison of rough-set and statistical methods in inductive learning. International J. Man-Machine Studies 24, 53–72 (1986)
15. Cohen, W.W.: Fast Effective Rule Induction. In: Machine Learning: Proceedings of the Twelfth International Conference (1996)
16. Brenth, M.: Instance-Based Learning Nearest Neighbour with Generalization. Working Paper Series (1995)
17. Sokolova, M., Marchand, M., Japkowicz, N., Shawe-Taylor, J.: The Decision List Machine. University of Ottawa, Canada, University of London Egham, UK (2002)
18. Compton, P., Edwards, G., Kang, B., Lazarus, L., Malor, R., Menzies, T., Preston, P., Srinivasan, A., Sammut, S.: Ripple down rules: possibilities and limitations. University of New South Wales, PO Box 1, Kensington NSW, Australia 2033, Department of Chemical Pathology, St. Vincent's Hospital Darlinghurst NSW, Australia (2010)
19. Richards, D., Compton, P.: Combining Formal Concept Analysis and Ripple Down Rules to Support the Reuse of Knowledge. School of Computer Science and Engineering, Sydney, Australia (1997)
20. Żwan, P., Szczuko, P., Kostek, B., Czyżewski, A.: Automatic Singing Voice Recognition Employing Neural Networks and Rough Sets. In: Peters, J.F., Skowron, A., Rybiński, H. (eds.) Transactions on Rough Sets IX. LNCS, vol. 5390, pp. 455–473. Springer, Heidelberg (2008)
21. Cyran, K.A., Mrozek, A.: Rough sets in hybrid methods for pattern recognition. International J. of Intelligent Systems 16(1), 149–168 (2001)
22. Wasserman, P.D.: Neural computing theory and practice. Van Nostrand Reinhold Co., New York (1989)
23. Weka Tool, http://www.cs.waikato.ac.nz/ml/weka/

Content-Based Scene Detection and Analysis Method for Automatic Classification of TV Sports News

Kazimierz Choroś and Piotr Pawlaczyk

Institute of Informatics, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
kazimierz.choros@pwr.wroc.pl

Abstract. A large amount of digital video data is stored in local or network visual retrieval systems. The new technology advances in multimedia information processing as well as in network transmission have made video data publicly and relatively easy available. Users need the adequate tools to locate their desired video or video segments quickly and efficiently, for example in Internet video collections, TV shows archives, video-on-demand systems, personal video archives offered by many public Internet services, etc. Detection of scenes in TV videos is difficult because the diversity of effects used in video editing puts up a barrier to construct an appropriate model. The framework of automatic recognition and classification of scenes reporting the sport events in a given discipline in TV sports news have been proposed. Experimental results show good performance of the proposed scheme on detecting scenes on a given sport discipline in TV sports news. In the tests a special software called AVI – the Automatic Video Indexer has been used to detect shots and then scenes in tested TV news videos.

Keywords: digital video segmentation, scene detection, scene classification, content-based video indexing, sport videos, TV sport news.

1 Introduction

New technology advances in visual retrieval systems allow the storage of a large amount of digital video data. Video data have become publicly and relatively easy available. However, without appropriate indexing and retrieval methods all these video data are hardly usable. Textual retrieval approaches are not efficient solutions in this case because users want to query not only technical data of videos such as length, format, type of compression, etc., but also the content of video clips. Manual indexing is unfeasible for large video collections. But the content-based automatic indexing and retrieval of video data are still processes difficult to be effectively performed. The content is very subjective to be characterized completely. It is usually concerned about main objects, second plan, background, domain, context, etc. This is one of the main reasons why the problem of content-based access is still largely unsolved. A user analyzing TV news would like to ask for specific events or the most attractive highlights presented in the news sequence. It is necessary to develop a method to

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 120–129, 2010.

organize, index, browse, and retrieve video archives in view of semantic level. Therefore, we are looking for effective tools to identify the video segments with a specific content, for example news on weather, sports, science, finances, technology, world travel, national economy, or entertainment news. On the other hand we would also want to detect and to remove commercial in TV news program. Content-based indexing of videos has become a research topic of increasing importance, difficult but at the same time fascinating, theoretical, scientific problem as well as practical task.

Digital video is hierarchically structured [1-3]. Video is composed of acts, episodes (sequences), scenes, shots, and finally of single frames. The most general unit is an act. So, a film is composed of one or more acts. Then, acts include one or more sequences, sequences comprise one or more scenes, and finally, scenes are built out of camera shots. A shot is a basic unit. A shot is usually defined as a continuous video acquisition with the same camera, so, it is a sequence of interrelated consecutive frames recorded contiguously and representing a continuous action in time or space. The length of shots affects a film. Shots with a longer duration make a scene seem more slower paced whereas shots with a shorter duration can make a scene seem dynamic and faster paced. The average shot length of a film is generally several seconds or more. The length of shots in TV sports news will be one of the crucial criteria in our experiments presented later.

The paper is organized as follows. The next section describes the main related works in the area of automatic scene detection and selection in sport videos. Moreover, some recent related research works are cited. The temporal segmentation process leading to the partition of a given video into a set of meaningful and individually manageable segments will be discussed in third section. This process is relatively well managed. The ASD module, i.e. the Automatic Shot Detection module of the Automatic Video Indexer [4] performs this task with adequate efficiency. The forth section presents a automatic detection of the sequence of shots making a scene. In the fifth section the experimental results for the classification of scenes in the tested TV sports news videos are reported. The tests performed using the ASA module – Automatic Shot Analyzer will show that it is possible to detect the soccer news in the sequence of TV sport news. The final conclusions and the future research work areas are discussed in the last 6th section.

2 Related Works

There are many recent investigations towards automatic recognition of a content of a video clip [5], many of the detection methods have been tested on sport videos. In the field of an automatic video processing research, a sport videos summarization has become a popular application because of its popularity to users and its simplicity due to repeated patterns. Also due to their huge commercial appeal sports videos represent an important application area for video automatic indexing and retrieval.

In [6] a two-level framework has been proposed to automatically detect goals in soccer video using audio/visual keywords. The first level extracts low-level features such as motion, colour, texture, pitch, etc. to detect video segments boundaries and label segments as audio and visual keywords. Then, two Hidden Markov Models have been used to model the exciting break portions with and without goal event,

respectively. The proposed approach has been applied to the detection of goal event in six half matches of soccer videos (270 minutes, 14 goals) from FIFA 2002 and UEFA 2002 and achieve 90% precision and 100% recall, respectively.

Another approach and another kind of analyses of sports news were implemented in a system [7] that performs automatic annotation of soccer videos. That approach has resulted in detecting principal highlights, and recognizing identity of players based on face detection, and on the analysis of contextual information such as jersey's numbers and superimposed text captions.

Furthermore, many experiments have been also performed on sports news classification and many approaches and schemes have been proposed. One of the first scene classification algorithm has been based on a DCT (Discrete Cosine Transformation) components extracted from the whole image and used it as the classification features [8]. An other technique described in [9] relies upon the concept of "cues" which attach semantic meaning to low-level features computed on the video. A cue detector was defined as a supervised specifically trained classifier. Examples of cue detectors included: grass, swimming pool lanes, ocean, or audio elements like referee whistle, crowd cheer. It has been also shown [10] that the weighting of individual classifier according to their estimated performance gives better results in automatic classifications. In [11] a unified framework for semantic shot classification in sports videos has been defined. The proposed scheme makes use of domain knowledge of specific sport to perform a top-down video shot classification, including identification of video shots classes for each sport. The method has been tested over 3 types of sports videos: tennis, basketball, and soccer. The results ranging from 80~95% have been achieved. In a new research work [12] it has been demonstrated that combining the tiny images and tiny videos datasets improves categorization precision in a wider range of categories.

There are also many promising experiments in which the specific features of sport courts are used to classify sport events in videos [13-15].

Other experiments have been carried out for example with baseball videos [16], with tennis videos [17], as well as with other sports.

3 Video Indexing Process

The process of automatic analysis and video indexing is composed of several stages. Generally, temporal segmentation is the first step of a video indexing. Such an approach is applied in our special software AVI - Automatic Video Indexer [4].

Automatic video file segmentation in the AVI includes five important steps (Fig. 1). The first step is a temporal segmentation process leading to the shot boundary detection. The second step is the key frame extraction, the best for depicting the content of corresponding shot or scene. And the third step is an analysis of the content in the shots detected in the video during the temporal segmentation. The detection of studio shots will lead in the fourth step to shot clustering, and in consequence scene segmentation. Finally, the content of a scene is recognized after shot content identification and shot clustering processes.

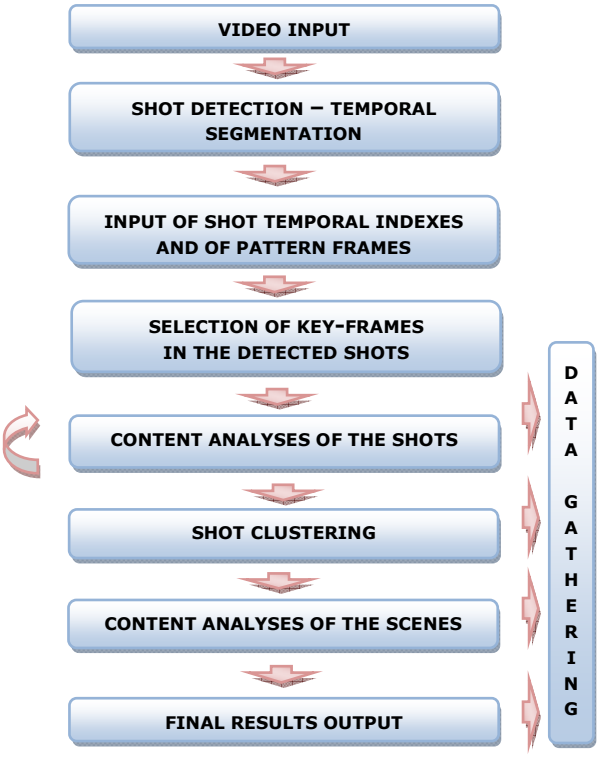


Fig. 1. Indexing scheme in the Automatic Video Indexer AVI

4 Temporal Segmentation Process

As it was already stated a video clip is structured into a strict hierarchy and is composed of different structural units: acts, episodes (sequences), scenes, and finally, camera shots. Depending on the style of video editing shots in a scene are content related but can be temporally separated and/or even spatially disconnected.

A shot change occurs when a video acquisition is done with another camera. The cut is the simplest and the most frequent way to perform a change between two shots, and at the same time cuts are probably the easiest shot changes to be detected. Cut takes place when the last frame of the first video sequence is directly followed by the first frame of the second video sequence. But mainly due to the application of digital movie editing software shot changes become more and more complex and more and more attractive for movie audience.

The other basic shot changes are fades and dissolves. A dissolve is a transition where all the images inserted between the two video sequences contain pixels whose values are computed as linear combination of the final frame of the first video sequence and the initial frame of the second video sequence. Cross dissolve describes the cross fading of two scenes. Over a certain period of time (usually several frames

or several seconds) the images of two scenes overlay, and then the current scene dissolves into a new one.

Cross dissolve in digital environment – contrarily to analogue systems – is relatively easily realized. Dissolve can be modelled as luminance value operations. It is performed according to the following mathematical formula

$$Y = \alpha Y^1 + (1-\alpha)Y^2 . \quad (1)$$

The luminance Y of the pixels of dissolved image is the sum of the luminance values Y^1 and Y^2 of the adequate pixels of two frames from two mixed shots. The parameter α decreases from one to zero. The number of its values determines the number of frames (duration) for a given dissolve effect [4].

Fades are special cases of dissolve effects, where a black frame most frequently replaces the last frame of the first shot (fade in) or the first frame of the second shot (fade out). Whereas, a wipe effect is obtained by progressively replacing the old image by the new one, using a spatial basis.

Many tests, experimental works have been undertaken, many papers have been written on temporal segmentation processes. The efficacy of many methods have been evaluated and it can be said that we can state we can effectively detect shots in digital videos. Also our specially designed software the Automatic Video Indexer [4] have achieved sufficient level of quality and reliability (Tab. 1) to be applied in practice for further research investigations. A more detailed description of the segmentation methods applied in the Automatic Video Indexer as well as a more detailed presentation of testing results will be found in [4].

The Automatic Video Indexer has been used to perform a temporal segmentation and to detect shots in tested movies. Then, the detected shots have been clustered.

Table 1. The best results of recall R and the best results of precision P of temporal segmentation methods received (but not necessarily simultaneously) for several categories of video when testing the effectiveness of the Automatic Video Indexer

Results with the best recall and the best precision	Pixel pair differences		Likelihood ratio method		Histogram differences		Twin threshold comparison	
	R	P	R	P	R	P	R	P
TV Talk-Show	1.00	1.00	1.00	0.98	1.00	1.00	1.00	0.89
Documentary Video	0.87	1.00	0.98	1.00	0.89	1.00	1.00	0.86
Animal Video	0.88	1.00	1.00	0.89	0.96	1.00	1.00	0.76
Adventure Video	1.00	0.80	1.00	0.76	0.92	1.00	0.97	0.75
POP Music Video	0.95	1.00	0.85	0.90	0.65	1.00	0.88	0.85

5 Automatic Scene Detection

TV sports news program has a specific structure. The analyses of TV sports news broadcasted in the first national Polish TV channel show that the program has its individual standard structure. It is composed of several highlights which are introduced and commented by anchorperson or numerical results presented in tables. These shots will be called as studio shots. Shots that belong to the same scene are visually similar and they are also located closely along the time axis. Two different classes are identified: studio shots and news report shots. A single scene is formed by all successive news report shots until the next studio shot.

If a scene is a set of shots it would be interesting to find which frame from which shot of a scene is the best for a classification of a whole scene [18], in our experiments which frame is the most adequate for a sport discipline identification. 333 shots detected in the tested ten TV sports news videos belong to 26 scenes which are composed of minimum two shots and, furthermore, there are 76 single shot scenes. There are 21 scenes which have at least five shots (Tab. 3). These 21 scenes were analyzed.

The usefulness evaluation of the frames taken from the first, middle, and finally the last shot of a scene has been evaluated. A certainty reflects the tester conviction of a sport discipline identification. The results obtained are presented in the Table 2.

The tests have indicated that the most adequate frame is not the frame from the beginning of the shot, what is frequently practiced, but rather from the middle part or from the end of a shot. Further investigations lead us to the conclusion that if the automatic process of the identification of a sport discipline is based on a single, still frame, such a frame should be chosen from the middle part of the first shot in a scene.

The next important observation of studio shots is that such a kind of shots significantly differs from other shots. First of all, the commentary from the TV studio as well as the presentation of sport scores in the form of different kinds of tables last much

Table 2. Usefulness of the frames from different time positions in the shots detected in ten TV sports news for the classification of sport video shots

	Sport discipline	Number of shots	Average certainty [0-10]	Time position of the key frame in a shot		
				10%	50%	90%
				Certainty level [0-10]		
1.	Alpinism	2	5.00	5.00	5.00	5.00
2.	Basketball	24	5.64	5.75	5.42	5.75
3.	Closing credits	4	5.83	5.00	7.50	5.00
4.	Car racing	24	3.61	2.33	3.96	4.54
5.	Commentary/Studio	63	5.34	5.40	5.56	5.08
6.	Golf	5	5.07	5.60	5.60	4.00
7.	Opening credits	4	7.00	2.50	9.75	8.75
8.	Preview	35	4.58	3.31	5.23	5.20
9.	Ranking table	7	9.10	7.57	10.00	9.71
10.	Ski jumping	11	6.24	5.82	6.45	6.45
11.	Soccer	143	5.28	5.18	5.34	5.34
12.	Speedway	11	5.64	5.64	5.64	5.64
	Arithmetic averages:		5.29	4.93	6.29	5.87

Table 3. Average length of a shot for a given sport discipline in the training set of TV sports news programs

Category/Discipline	Average shot length [in frames]
Soccer	91
Golf	125
Speed way	145
Cross-country skiing	189
Basketball	239
Commentary/Studio	413
Table	438

longer than other shots. The analyses of the length of shots in TV sports news presented in the Table 3 show that studio shots last twice longer than any other and that the soccer shots are generally the shortest ones. In most cases all shots between two studio or table shots are from the same semantic video scene. A scene is detected as a series of shots separated by these two kinds of relatively easily detected shots: studio and table.

6 TV Sports News Categorization

The collections of 20 TV sports news programs have been used in the tests. These videos were broadcasted on different days in the first national Polish TV channel and after the digitisation process they constructed TV sports news database in the DV format (720 x 576 pixels, full colour) for training and evaluation. Seven videos have been used as a training set, the next 13 have been used in an automatic content scene analysis leading to the scene categorization, i.e. to the recognition of a sport category (discipline). The training set has also been used to choose the most representative still frames for a given sport discipline.

Content analysis of shots is performed by three algorithms. In the first step the longest shots are selected as the most probable studio or table shots and are treated as scene boundaries. Then, the well-known technique of measuring the distance of histograms of pattern frames set and the key-frames, that is the most representatives frames indicated by the strategy described in the previous section has been applied.

Histograms are commonly used to classify images in content-based image retrieval systems. Two images are compared by measuring the distance or similarity of their histograms. Various distance/similarity measures are applicable to compare two histograms [19, 20]. The standard distance D of quantized colour histograms H of two images I_A and I_B has been applied in the AVI experiment and measured as follows:

$$D_H(I_A, I_B) = \sum_{k=1}^{n} \left| H^k(I_A) - H^k(I_B) \right|, \tag{2}$$

where n is the number of colours in the colour space.

To improve the results obtained using the histogram distance measure in the next step of shot content analysis a colour coherence vector has been measured.

A colour coherence vector measure [19] improves global histogram matching and takes into account spatial information in colour images. Colour coherence vector

indicates if pixels belong to a large region of similar colour. Each pixel is classified as either coherent or incoherent to a given colour. If every pixel in the set has at least one pixel of the same colour among its eight closest neighbours, such a set is called a maximal set. The size of a maximal set must exceed a given threshold, then a whole region is classified as coherent. The total number α of coherent and the total number β of incoherent pixels are computed for each colour of n colours in a discretized set of colours. The colour coherence vector V_C of an image is defined as:

$$V_C = [(\alpha_1, \beta_1), (\alpha_2, \beta_2), ..., (\alpha_n, \beta_n)]. \tag{3}$$

Two colour coherence vectors of two images A and B can be compared according to the following distance formula:

$$D_C(I_A, I_B) = \sum_{k=1}^{n} (\left|\alpha_{A_k} - \alpha_{B_k}\right| + \left|\beta_{A_k} - \beta_{B_k}\right|). \tag{4}$$

These three techniques have been used during the content analysis of TV sports news classification process.

The techniques used during the content analysis enable us to receive very promising results for soccer scene retrieving, presented in Table 5, of standard measures of recall (from 0.51 to 1) and precision (from 0.60 to 0.82).

Table 4. The example of analysis of a video reflecting the structure of a video as well as the performance of the Automatic Video Indexer

Shot position in video	Shot length [in frames]	Sport discipline	Discipline identified	Correctness
1	126	opening credits	-	no
2	3	soccer	soccer	yes
3	72	soccer	soccer	yes
4	76	studio	studio	yes
5	121	basketball	studio	no
6	74	speed way	studio	no
7	675	studio	studio	yes
8	33	soccer	soccer	yes
9	78	soccer	soccer	yes
10	35	soccer	soccer	yes
11	78	soccer	soccer	yes
12	387	soccer	soccer	yes
13	93	soccer	soccer	yes
14	10	soccer	soccer	yes
15	29	soccer	soccer	yes
16	65	soccer	soccer	yes
17	76	soccer	soccer	yes
18	92	soccer	soccer	yes
19	299	soccer	soccer	yes
20	1128	studio	studio	yes
21	25	basketball	soccer	no
22	73	basketball	soccer	no
23	23	basketball	studio	no
24	5	basketball	studio	no

Table 4. (*Continued*)

Shot position in video	Shot length [in frames]	Sport discipline	Discipline identified	Correctness
25	19	basketball	soccer	no
26	45	basketball	soccer	no
27	44	basketball	soccer	no
28	67	basketball	-	no
29	61	basketball	soccer	no
30	2502	studio	studio	yes
31	151	speed way	soccer	no
32	236	speed way	studio	no
33	86	speed way	soccer	no
34	35	speed way	table	no
35	57	speed way	-	no
36	50	speed way	studio	no
37	243	speed way	studio	no
38	234	speed way	studio	no
39	198	speed way	soccer	no
40	1057	studio	studio	yes

Table 5. The extreme recall values and precision for two videos of the automatic content-based recognition of soccer scenes

Discipline	Recall	Precision
Video 1		
Commentary/studio	1	0.38
Soccer	1	0.60
Video 2		
Commentary/studio	1	0.26
Soccer	0.51	0.82

7 Final Conclusion and Further Studies

The framework of a sport classification of TV sport news broadcasted from the first Polish national TV channel has been proposed. Furthermore, a special software the Automatic Video Indexing has been designed and implemented. The main tasks of the AVI software are: temporal segmentation of videos, key-frames selection, shot analysis and clustering, soccer shot retrieving. The first results are satisfactory.

In further research the pattern frames for other sport disciplines will be selected. Then, the most frequent number of shots in a scene for a given sport discipline and specific structures of scenes for a given sport discipline will be analyzed. Finally, new computing techniques are being developed leading to new functions provided to implement in the Automatic Video Indexer. We want to extend its functionality by introducing an automatic extraction of video features and objects like faces, lines, texts, etc., as well as to extend its application to other kinds of TV shows.

References

1. Zhang, Y.J., Lu, H.B.: A hierarchical organization scheme for video data. Pattern Recognition 35, 2381–2387 (2002)
2. Money, A.G., Agius, H.: Video summarisation: a conceptual framework and survey of the state of the art. Journal of Visual Communication and Image Representation, 121–143 (2008)
3. Choroś, K.: Digital video segmentation techniques for indexing and retrieval on the Web. In: Advanced Problems of Internet Technologies, pp. 7–21. Academy of Business, Dąbrowa Górnicza (2008)
4. Choroś, K., Gonet, M.: Effectiveness of video segmentation techniques for different categories of videos. In: New Trends in Multimedia and Network Information Systems, pp. 34–45. IOS Press, Amsterdam (2008)
5. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval: State of the art and challenges. ACM Transactions on Multimedia Computing, Communications, and Applications, 1–19 (2006)
6. Kang, Y.-L., Lim, J.-H., Kankanhalli, M.S., Xu, C., Tian, Q.: Goal detection in soccer video using audio/visual. In: Proceedings of the ICIP, pp. 1629–1632 (2004)
7. Bertini, M., Del Bimbo, A., Nunziati, W.: Automatic annotation of sport video content. In: Sanfeliu, A., Cortés, M.L. (eds.) CIARP 2005. LNCS, vol. 3773, pp. 1066–1078. Springer, Heidelberg (2005)
8. Ariki, Y., Sugiyama, Y.: Classification of TV sports news by DCT features using multiple subspace method. In: Proceedings of Fourteenth International Conference on Pattern Recognition, vol. 2, pp. 1488–1491 (1998)
9. Messer, K., Christmas, W., Kittler, J.: Automatic sports classification. In: Proceedings of the 16th International Conference on Pattern Recognition, pp. 1005–1008 (2002)
10. Vakkalanka, S., Krishna Mohan, C., Kumaraswamy, R., Yegnanarayana, B.: Combining multiple evidence for video classification. In: Proceedings of the International Conference on Intelligent Sensing and Information Processing, pp. 187–192 (2005)
11. Ling-Yu, D., Min, X., Qi, T., Chang-Sheng, X., Jin, J.S.: A unified framework for semantic shot classification in sports video. IEEE Transactions on Multimedia, 1066–1083 (2005)
12. Karpenko, A., Aarabi, P.: Tiny videos: a large dataset for image and video frame categorization. In: Proceedings of the 11th IEEE International Symposium on Multimedia, ISM 2009, pp. 281–289 (2009)
13. Wang, D.-H., Tian, Q., Gao, S., Sung, W.-K.: News sports video shot classification with sports play field and motion features. In: ICIP 2004 International Conference on Image Processing, pp. 2247–2250 (2004)
14. Zhong, D., Chang, S.-F.: Real-time view recognition and event detection for sports video. Journal of Visual Communication and Image Representation, 330–347 (2004)
15. Chena, L.-H., Laib, Y.-C., Liaoc, H.-Y.M.: Movie scene segmentation using background information. Pattern Recognition, 1056–1065 (2008)
16. Lien, C.-C., Chiang, C.-L., Lee, C.-H.: Scene-based event detection for baseball videos. Journal of Visual Communication and Image Representation, 1–14 (2007)
17. Delakis, M., Gravier, G., Gros, P.: Audiovisual integration with segment models for tennis video parsing. Computer Vision and Image Understanding, 142–154 (2008)
18. Choroś, K.: Video shot selection and content-based scene detection for automatic classification of TV sports news. In: Internet – Technical Development and Applications. Advances in Soft Computing, vol. 64, pp. 73–80. Springer, Heidelberg (2009)
19. Pass, G., Zabih, R., Miller, J.: Comparing images using color coherence vectors. In: Proceedings of ACM Multimedia, pp. 65–73 (1996)
20. Cha, S.: Taxonomy of nominal type histogram distance measures. In: Proceedings of the American Conference on Applied Mathematics, pp. 325–330 (2008)

Combining Multiple Classification or Regression Models Using Genetic Algorithms

Andrzej Janusz

Faculty of Mathematics, Informatics, and Mechanics, The University of Warsaw,
Banacha 2, 02-097 Warszawa, Poland
andrzejanusz@gmail.com

Abstract. Blending is a well-established technique, commonly used to increase performance of predictive models. Its effectiveness has been confirmed in practice as most of the latest international data-mining contest winners were using some kind of a committee of classifiers to produce their final entry. This paper is a technical report presenting a method of using a genetic algorithm to optimize an ensemble of multiple classification or regression models. An implementation of this method in *R*, called Genetic Meta-Blender, was tested during the Australian Data Mining 2009 Analytic Challenge competition and it was awarded with the Grand Champion prize for achieving the best overall result. In the report, the purpose of the challenge is described and details of the winning approach are given. The results of Genetic Meta-Blender are also discussed and compared to several baseline scores.

1 Introduction

An ensemble can be defined as a set of separately trained classifiers whose predictions are combined in order to achieve better accuracy ([1]). Many researchers have investigated the problem of constructing successful ensembles ([1], [2], [3]).

Numerous experiments on real-life datasets confirmed that the most accurate ensembles are characterized by diversity and high performance of individual classifiers. The most commonly used methods of constructing such sets of predictive models are *bagging*[1] and *boosting* ([2], [4], [5]). In both of these methods a single learning algorithm is used to create multiple classifiers. In the classical bagging algorithm, the training dataset is resampled multiple times and the models are trained on each of the bootstrap samples. Tested instances are then assessed by each of the models and the final prediction is made by voting or by averaging the output of individual predictors. In the boosting approach, an ensemble is built incrementally. In each step of the algorithm, a new classifier is constructed and new weights are assigned to training instances so that the examples misclassified by previous classifier become more important in the next step. The resulting model aggregates individual models based on their predictive power.

The desired diversification of models in the ensemble can be also achieved by including different learning algorithms, using different parameter settings or

[1] Bootstrap aggregating.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 130–137, 2010.

features selection techniques ([3], [6]). This approach can be used in combination with bagging and boosting and it was proved to be successful in practice ([6], [7]).

Another factor that influences the effectiveness of classifier committees is a voting strategy or an aggregation function, which may be seen as methods of combining decisions of individual models in the ensemble. Many voting and aggregation methods were investigated in the literature ([8]). The most popular methods include majority voting and weighted voting, in which the weights are usually dependent on accuracies of the base predictors. A major drawback of such approaches is that they do not take into account correlation between decisions of aggregated models.

The Genetic Meta-Blender (GMB) presented in this paper is an algorithm for computing weights of models in the ensemble which is not based on the performance of individuals. Instead, it uses the notion of genetic programming ([9]) to find the global optimum of a given scoring function (which could be, for example, an area under the ROC curve). This approach was used during the Australian Data Mining 2009 Analytic Challenge competition to construct successful second level ensembles[2] In the next section the challenge is described and in the later parts of this paper the way in which the GMB was utilized is discussed.

2 AusDM 2009 Analytic Challenge

Australian Data Mining 2009 Analytic Challenge was a special event of the AusDM 2009 Conference that took place on 1–4 December 2009 in Melbourne, Australia. The challenge was related to the problem of ensembling and it was divided into two tasks. Three datasets were made available for both tasks, each consisting of a different number of expert models that made predictions of movie ratings from the Netflix database. The models were provided by *The Ensemble* and *BellKor's Pragmatic Chaos* – the two teams that placed first and second in the Netflix competition ([10]).

For the first set of tables the task was to minimize the root mean squared error of made predictions and for the second set the task was to maximize the AUC score[3] The datasets from the first task (later on called RMSE) were labeled with the actual movie ratings – integers from the set $\{1000, 2000, 3000, 4000, 5000\}$. In the second task (later on called AUC), the data tables were labeled with binary decision attributes whose meaning was slightly different for each table. In both tasks the available datasets had three sizes. The *small* datasets contained 30000 movie ratings described by 200 expert models. Results achieved on those datasets were posted on a leaderboard which was publicly available during the competition and they were not taken into account in the final ranking. The *medium* and the *large* datasets had 40000 and 100000 movie ratings respectively and were described by 250 and 1151 predictors. All the datasets were divided

[2] Ensembles of ensembles.

[3] The quality measure used in this task was a Gini coefficient which can be computed as $Gini = 2 * \|AUC - 0.5\|$.

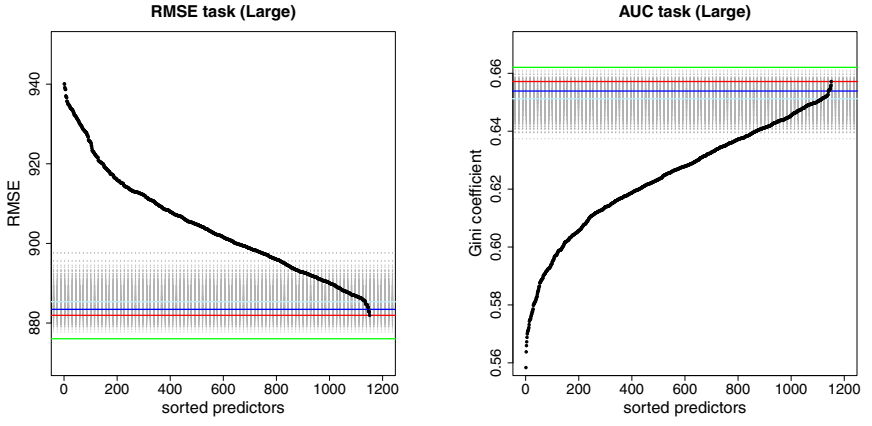


Fig. 1. Performance of individual models from large RMSE and AUC tasks. The red horizontal line marks the best models from the sets. The blue line indicates the results of the ensemble made by averaging all 1151 predictors and the green line points out the scores achieved by averaging the best 10 models. The gray dotted lines correspond to scores of random ensembles and the light blue line is their empirical median.

into a training and a test sets in proportion of 50%/50% ratings each. The true label values were available only for the ratings from the training sets.

Figure 1 shows distributions of scores achieved by individual models from the large data tables of the RMSE and AUC tasks. They were computed on the training sets. It is interesting to notice that only 5 models from the RMSE task ($\approx 0.4\%$) and 10 models from the AUC task ($\approx 0.9\%$) performed better than the ensemble made by simply averaging all the available models (the blue lines on the plots). Additionally, on average[4], only 17 models from the RMSE task ($\approx 1.5\%$) and 36 models from the AUC task ($\approx 3.1\%$) were superior to a random ensamble which was constructed by averaging 10 randomly chosen predictors. In comparison to the scores achieved by the ensembles made of the best 10 models from each task (the green lines), the median scores of the random ensembles (the light blue lines) were lower only by $\approx 1.05\%$ and $\approx 1.64\%$, respectively.
The question arises: what is the best way of combining models in the ensemble?

3 Genetic Meta-Blender – A General Idea

The main idea of the Genetic Meta-Blender is simple: instead of averaging or assigning weights based on performances of individual predictors, GMB utilizes the genetic algorithm to optimize proportions between models in the final blend. This optimization is done by searching for a set of weights, which is an approximation of a global maximum of a predefined scoring function. The selection of

[4] A median from 1000 experiments is taken.

the proper scoring function should be dictated by the quality measure which will be used to evaluate the performance of the final ensemble. For example, if the task is to minimize the root mean squared error of the predictions then the following scoring function should be maximized:

$$Score(w_1, \ldots, w_k) = -\sum_{i=1}^{n} \left(trueValues^{(i)} - \frac{\sum_{j=1}^{k} w_j * predValues_j^{(i)}}{\sum_{j=1}^{k} w_j} \right)^2 \quad (1)$$

In this function, $trueValues^{(i)}$ is a true target value of the i-th training example and $predValues_j^{(i)}$ is a predicted target value of the i-th training example made by the j-th model in the ensemble.

To learn the optimal set of weights of models it is necessary to prepare a sufficient number of training data. In order to avoid overfitting, it is recommended to compute predictions of all learning algorithms which are being utilized for the whole training data using the cross-validation technique. Additionally, models which were constructed for each cross-validation fold can also make predictions for the test data. Those predictions may be combined in the final ensemble to make it even less vulnerable to overfit. An exemplary scheme of the GMB method is given in Appendix.

4 GMB in Practice

In this section the approach which were used in the AusDM 2009 Analytic Challenge is discussed. As it was described in Section 2, the data in this competition consisted of predictions made by numerous models. This fact made a good opportunity to verify how well a multi-level ensemble combined with the GMB optimization method will perform.

In the challenge, the assessment of samples from the score data sets was conducted in two steps. First, for each training set, a wide range of predictive models was constructed. The GMB scheme showed in Appendix was implemented in the *R System* ([11]). For the AUC task, popular classification models available in standard *R* libraries were tried: linear and logistic regression models (library *stats*), neural networks (library *nnet*), recursive partitioning trees (library *rpart*), k-NN (library *class*), the random forest (library *randomForest*) and the generalized boosting models (library *gbm*). The linear, logistic and neural network models were additionally averaged over multiple runs on different attribute subsets. The neural networks had one hidden layer which contained 1 to 5 neurons. The recursive partitioning trees were bagged and a few values of the complexity parameter were tried. The k-NN algorithm was used as a scoring model, several k values between 50 and 150 were used. The generalized boosting models were fitted with the *bernoulli*, *gaussian* and the *adaboost* loss functions.

For the RMSE task, due to lack of time for experiments with the parameters settings, only linear regression models and simple neural networks were used.

Each model's prediction values for samples from the training sets were acquired by the cross-validation test and used in the second step as an input for

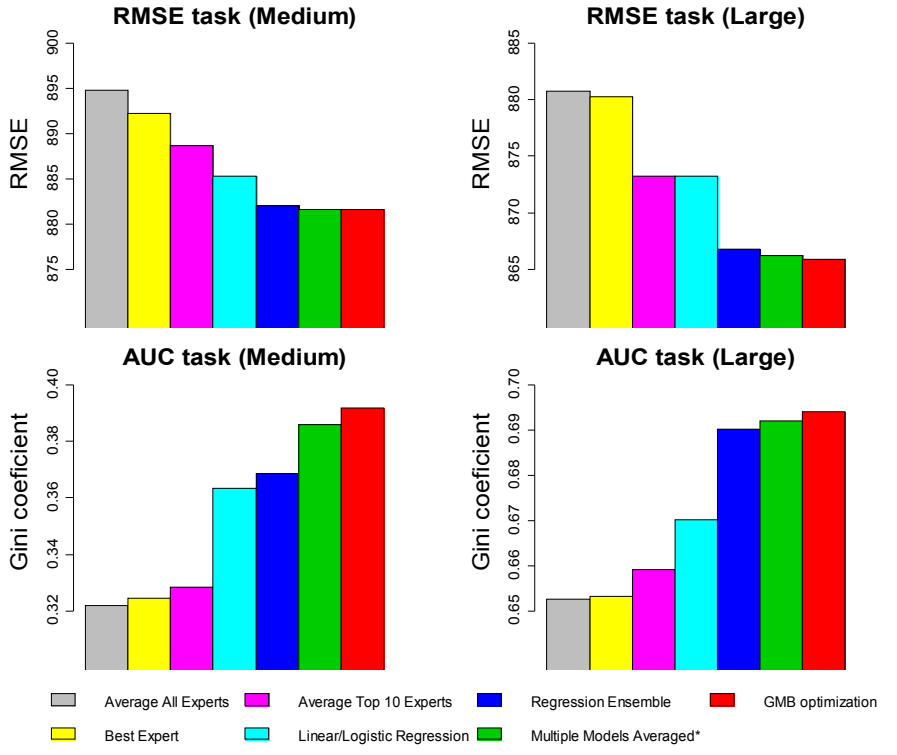


Fig. 2. Results of the GMB compared to the straight average of models from which it was constructed and several others baseline scores provided by the organizers of AusDM 2009 Analytic Challenge.
*A straight average ensemble of the models which were optimized by the GMB.

the blending algorithm. The genetic algorithm, which was implemented in *R* for the purpose of the challenge, used different scoring functions for each of the tasks. It tried to directly maximize the AUC or minimize the RMSE by assigning significance levels (weights) to models in the ensemble. The population was coded as a list of vectors of weights. In the experiments, the population size was set to 500. The total number of models included for the GA optimization in the final submission was dependent on the size of the datasets and the task. It was limited to 20 for the small and medium AUC data, to 25 for the large AUC data and to 10 for all sizes RMSE data. The restriction on number of models was introduced to avoid over-fitting. Some other precautions, such as a restriction on the granularity of the weights of the models were also taken. The algorithm was stopped when the averaged quality of the population members did not change significantly in 5 consecutive generations.

Figure 2 shows the final results of the presented method. The scores achieved by the GMB on the medium and large datasets are compared to several baseline results provided by the organizers after completion of the challenge. Beside the average of all experts, the best expert and the average of top 10 experts, the performance of 3 other meta-models are given. The light blue and dark blue bars indicate the scores of two approaches which recently were particularly popular (e.g. during the Netflix competition). The first one employs a linear regression model (or logistic regression for the AUC task) as a meta-learning algorithm which combines decisions of experts in the ensemble. The second one is similar, in a sense that it also utilizes the linear or logistic regression models but in this method, those predictors are additionally bagged to create a second-level ensemble. The third of the compared baseline approaches, denoted by the green bars in the figure, is a straight average of the meta-models which went into the optimization by the GMB. Those scores were included to verify how much prediction accuracy is gained by the usage of the GMB optimization method.

More details about the AusDM 2009 Analytic Challenge, tables with the scores of all competitors and brief descriptions of the leading methods can be found at the web site with the results of the challenge[5].

5 Conclusions

The main scope of this paper was the Genetic Meta-Blender – a method of optimizing an ensemble of multiple predictive models using a genetic algorithm. The general idea of this meta-model, as well as its results from Australian Data-Mining 2009 Analytic Challenge was presented. The datasets and the tasks of this competition was also briefly described.

The results showed in Figure 2 confirm usefulness of the GMB. The GA optimization led to a better score of the final solution than the straight average in 3 out of 4 datasets. The differences between the optimized and the averaged models are generally less significant for the RMSE task, which is perhaps due to lower diversification of the utilized predictors. It is also noticeable that the multi-level ensembles greatly outperformed the single-level ones. For example, the bagged version of the logistic regression ensemble (the dark blue bars on the charts from the AUC tasks) achieved $\approx 3.0\%$ better score on the large AUC data than the model without bagging. Finally, the standard statistical or machine-learning models proved to be very effective as meta-learning algorithms which can be used to combine predictions in the ensemble.

Acknowledgements. The author would like to thank Michał Grotowski for proofreading this paper. This research was partially supported by the grants N N516 368334 and N N516 077837 from Ministry of Science and Higher Education of the Republic of Poland.

[5] http://www.tiberius.biz/ausdm09/results.html

References

1. Opitz, D., Maclin, R.: Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research 11, 169–198 (1999)
2. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning 36(1-2), 105–139 (1999)
3. Stefanowski, J.: An experimental study of methods combining multiple classifiers - diversified both by feature selection and bootstrap sampling. In: Atanassov, K.T., Kacprzyk, J., Krawczak, M., Szmidt, E. (eds.) Issues in the Representation and Processing of Uncertain and Imprecise Information, pp. 337–354. Akademicka Oficyna Wydawnicza EXIT, Warsaw (2005)
4. Maclin, R., Opitz, D.W.: An empirical evaluation of bagging and boosting. In: AAAI/IAAI, pp. 546–551 (1997)
5. Quinlan, J.R.: Bagging, boosting, and c4.5. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence, pp. 725–730 (1996)
6. Caruana, R., Niculescu-Mizil, A., Crew, G., Ksikes, A.: Ensemble selection from libraries of models. In: Proceedings of the 21st International Conference on Machine Learning, pp. 137–144. ACM Press, New York (2004)
7. Caruana, R., Munson, A., Niculescu-Mizil, A.: Getting the most out of ensemble selection. In: Proceedings of the 6th IEEE International Conference on Data Mining, pp. 828–833 (2006)
8. Tsymbal, A., Puuronen, S., Skrypnyk, I.: Ensemble feature selection with dynamic integration of classifiers. In: Int. ICSC Congress on Computational Intelligence Methods and Applications, pp. 558–564 (2001)
9. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Heidelberg (1996)
10. Bennett, J., Lanning, S., Netflix, N.: The netflix prize. In: KDD Cup and Workshop in conjunction with KDD (2007)
11. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2008)

Appendix

A computation scheme of the GMB ensemble optimization:

Input:
A set of k learning algorithms (LA);
A training set ($TrSet$);
A test set ($TeSet$);
A number of cross-validation folds (n);

Output:
A vector of the final predictions for samples from $TeSet$ ($finalPreds$);

Begin

Divide the a training set $TrSet$ into n disjoint subsets;

for $i = 1$ **to** n **do**

Train k models with the learning algorithms from LA using all but i-th subsets;

With each model, predict target values for the samples from i-th subset;

With each model, predict target values for the samples from $TeSet$;

end

For each of k models, average or decide by the majority voting its final predictions for samples from $TeSet$, creating a set of predictions $PredSet = \{predVealues_i : i = 1, \ldots, k\}$;

Use predictions for the samples from $TrSet$ as an input for a genetic algorithm and compute the values of approximately optimal weights $\{w_1, \ldots, w_k\}$;

Combine prediction vectors from the set $PredSet$ using the formula:

$$finalPreds = \frac{\sum_{i=1}^{k} w_i * predVealues_i}{\sum_{i=1}^{k} w_i}$$

End

Argument Based Generalization of MODLEM Rule Induction Algorithm

Krystyna Napierała and Jerzy Stefanowski

Institute of Computing Science, Poznań University of Technology,
ul. Piotrowo 2, 60–965 Poznań, Poland
{krystyna.napierala,jerzy.stefanowski}@cs.put.poznan.pl

Abstract. Argument based learning allows experts to express their domain, local knowledge about the circumstances of making classification decisions for some learning examples. In this paper we have incorporated this idea in rule induction as a generalization of the MODLEM algorithm. To adjust the algorithm to the redefined task, a new measure for evaluating rule conditions and a new classification strategy with rules had to be introduced. Experimental studies showed that using arguments improved classification accuracy and structure of rules. Moreover the proper argumentation improved recognition of the minority class in imbalanced data without essential decreasing recognition of the majority classes.

Keywords: rule induction, argument-based learning, MODLEM algorithm, learning from imbalanced data.

1 Introduction

Discovering rules from examples is one of the main tasks in machine learning, data mining and also in rough set theory. Up to now several algorithms have been proposed to induce rules – for review see, e.g., [2,10]. However, new researches on improving rule induction are still undertaken. One of the directions include incorporating *domain knowledge* into the learning process. It should direct the learning process to provide the rules more consistent with experts' expectations, leading to better classification abilities and possibly reducing the complexity of learning. For a review of different approaches see, e.g., [5].

This kind of additional knowledge is usually formulated *globally* with respect to the whole domain of the problem. However, it can be difficult for experts to express it. Therefore, recently another paradigm has been introduced as *argument based machine learning* (briefly denoted as ABML) [6,8]. According to it some difficult learning examples can be additionally annotated by the expert's explanations called *arguments*, i.e. an expert gives descriptions of reasons for assigning the example to the given class. This approach uses "*local*" expert's knowledge which concerns specific situations and is valid for limited, chosen examples rather than for the whole domain [6]. This idea has been originally introduced by Bratko *et al.* and implemented for rule induction as an extension

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 138–147, 2010.

of a CN2 algorithm [6]. Moreover, they applied it to problems of justification of cases in law, loan policy and medical treatment, see e.g. [7].

Despite their promising results, we found that there are still some aspects which could be inspected in more detail. The main aim of our paper is to adopt the ABML paradigm into another rule induction schema than CN2. We decided to choose the MODLEM algorithm [9], which is more suited to deal with numerical and imperfect data. Although our generalization, called ABMODLEM, is inspired by the paper [8], we have to consider new methodological aspects. First of all, it is necessary to introduce another measure for evaluating candidate elementary conditions to be added to a rule, which allows us to obtain more general rules, in particular ones covering argumented examples. Secondly, while classifying new objects with rules, a new classification strategy is required which takes into account that rules induced from argumented examples are usually supported by fewer examples than non-argumented rules.

Another contribution of this paper is studying the influence of argumentation on the recognition of particular classes. This aspect has not been considered in [6,8], however our preliminary experiments showed that it is particularly interesting for imbalanced data. We want to provide a more precise experimental study to find out if ABML can be useful for problems where the recognition of minority classes is particularly important.

All above problems will be experimentally evaluated on three data sets coming from UCI repository. We carry out a comparative study of the argument based ABMODLEM vs. basic version of the MODLEM algorithm.

2 Basic Concepts of Argument Explanations

Following ideas of Bratko and Mozina on ABML [6,8], we assume that the domain expert's explanations for some of the learning examples are given in a form of arguments *for* and *against the decision* (called positive and negative arguments, respectively).

An argumented example AE is denoted as a triple (*Attributes, Decision, Arguments*), where *Attributes* and *Decision* are defined using standard attribute-value pairs, and *Arguments* is a set of positive and negative explanations in the following form: *Positive argument* is defined as *Decision because of Reasons* and *Negative argument* is defined as *Decision despite Reasons*. *Reasons* are expressed as conjunctions of *attribute-value* expressions r_i which take a form similar to elementary conditions used in the syntax of a rule. An example of this notation is further given together with the illustrative example of ABML.

The rule R covering argumented examples should have its condition part consistent with the argumentation. Thus, following [8], the new definition of *AB-covering* (*argument-based covering*) is: rule R *AB-covers* an argumented example AE if: (1) all the conditions in R are true for the description of AE, (2) a condition part of R is consistent with at least one positive argument of AE, and (3) it is not consistent with any of the negative arguments of AE, where consistency means that the condition part of R contains elementary expressions

r_i from *reasons* or being their generalized forms. To illustrate it let us consider a simple example of admission decisions for three patients. Their description is given below:

patient	temperature	stomach ache	blood test result	blood pressure	admitted
Johns	high	no	bad	normal	yes
Biggle	normal	no	bad	v.high	yes
Perkins	high	yes	good	normal	no

If one wants to discover a rule explaining a decision on admitting a patient to a hospital, a typical rule induction algorithm will produce the following rule: *if (stomach ache = no) then (admitted = yes)*, which covers all the positive examples of this decision although this rule contradicts common sense.

A physician asked to explain why patient Johns was admitted to hospital could explain it as: (giving a positive argument) *"patient Johns was admitted to hospital because his body temperature = high"*. He could also explain the decision by giving a negative argument: *"Patient Johns was admitted to a hospital despite stomach ache = no"*. Notice that this argumentation is "local" – the physician claims that Johns was admitted to hospital because of temperature, but he does not claim that all patients with high temperature are automatically admitted (as is the case of Mr.Perkins). This argumented example is formally denoted as:

$$AE = (Attributes = \{Johns,\ high,\ no,\ bad,\ normal\},\quad Decision = yes,$$
$$Arguments = \{Decision = yes\ because\ temperature = high,$$
$$Decision = yes\ despite\ stomach\ ache = no\})$$

Including the argumentation in the rule induction process will shift the induction process towards a desired direction: now the rule built around patient Johns will neglect the attribute *stomach ache* (to prevent consistency with the negative argument), and include the attribute *temperature* (to preserve consistency with the positive argument). An algorithm using this argumentation would therefore induce a rule *if (temperature = high) and (blood test results = bad) then (admitted = yes)*, which is consistent with common sense and the expert's knowledge.

3 Description of ABMODLEM Algorithm

3.1 Construction of the Algorithm

The ABML paradigm will be incorporated inside MODLEM algorithm [9]. It is a sequential covering algorithm that induces a minimal set of unordered rules. It iteratively searches for the best rule for a given class, removes all covered positive examples from the learning set and continues the procedure until all the examples from that class are covered. The process is repeated for each decision class. A construction of a single rule starts from finding the best condition, and continues by adding new conditions until a stopping criterion is met. The specific property of MODLEM consists in direct processing numerical values of attributes

(without pre-discretization) and missing values. It can also be adopted to handle inconsistent or noisy examples either by rule pruning or rough approximations. Details of MODLEM can be found in [9,10,11].

```
Procedure ABMODLEMForOneClass (Examples ES,
                               Class T)

 1.Let RULE_LIST be an empty list.
 2.Let AES be the set of examples
          that have arguments.
 3.Evaluate arguments (as if they were rules)
      and sort examples according to the
      evaluation of their best argument.


 4.while AES is not empty do
 5. Let AE1 be the first example in AES.
 6. Let BEST_RULE be
          ABFindBestRule(ES,AE1,T).
 7. Add BEST_RULE to RULELIST.
 8. Remove from AES examples
          AB-covered by BEST_RULE.
 9.end while


10.for all RULE in RULE_LIST do
11. Remove from ES examples covered by RULE.
12.end for


13. Add rules obtained with
        MODLEMForOneClass(ES,T) to RULE_LIST.
```

Procedure 1. ABMODLEMForOneClass

```
Procedure ABFindBestRule (Not_covered_ex ES,
                          Argumented_Example AE,
                          Attributes Attr)

 1.LET BEST_RULE be an empty rule.
 2.foreach Positive_argument Arg for AE
 3. Let RULE be a set of reasons of Arg.
 4. Let S be a set of objects in ES
                          covered by RULE

 5. while (S contains negative examples)
 6.  Let BEST_CONDITION be an empty
                     elementary condition
 7.  foreach attribute A in Attr
 8.   Let NEW_CONDITION be
           ABFindBestCondition(A,S,AE,RULE)
 9.   if (NEW_CONDITION is better then
                          BEST_CONDITION)
10.    BEST_CONDITION = NEW_CONDITION
11.  end foreach
12.  Add BEST_CONDITION to RULE
13.  Update S
14. end while
15. Remove unnecessary conditions from RULE
16. if (RULE is better than BEST_RULE)
17.  BEST_RULE = RULE
18.end foreach
```

Procedure 2. ABFindBestRule

A general framework of our generalized algorithm (called ABMODLEM) is inspired by some solutions from ABCN2 algorithm [8]. The main schema of ABMODLEM is given in Procedure 1. In the first phase, the algorithm induces rules that AB-cover the argumented examples, and preferably other examples - to build as general rules as possible. To achieve it, argumented examples are sorted with respect to evaluation measures (calculated for conjunction reasons) so that the algorithm started from constructing rules that have a chance to cover many positive examples. After generating argumented rules, if there are still some remaining examples not covered by those rules, the remaining set of rules is found by standard MODLEM procedure. Our technique for constructing condition parts of rules is definitely different to ABCN2. To find the best rule that AB-covers an argumented example (Procedure `ABFindBestRule(ES,AE1,T)` listed in Procedure 2.), one rule is built from each positive argument (to assure coherence with at least one positive argument given for the example). All reasons of the argument are added as elementary conditions to a condition part of the rule and if the stopping criterion is not met (e.g. the rule still covers some negative examples from other classes), additional conditions are added iteratively by

`ABFindBestCondition(a,S,AE,RULE)`. Then, the induced rule created for the example AE is added to the rule set.

Procedure `ABFindBestCondition(a,S,AE,RULE)` finds the best condition by comparing candidate conditions for each attribute, assuring the coherence with arguments. For nominal attributes, as the rule must cover the argumented example, the condition must take the form *attribute* = *value*, where values comes from reasons of the argumented example. If adding this condition to the rule will be consistent with any negative arguments for the analysed example, this condition is skipped. For numeric attributes conditions are in form of $X > x_i$, $X < x_i$, $X \geq x_i$ or $X \leq x_i$. For a particular x_i, the direction of the condition is chosen so that the condition covered the argumented example. To choose x_i, candidate thresholds are built between values present for the attribute in the learning set (which discriminate examples from different classes). For each threshold, a temporary condition is built and added to the rule. If it does not violate any negative arguments, new candidate rule is evaluated using an evaluation measure and the best condition is chosen.

3.2 Rule Evaluation Measure

In the original formulation of MODLEM candidates for the condition part of the rule are evaluated either by class entropy (calculated for the set of covered examples) or Laplace Accuracy - for details see [11]. Although these measures worked well for many problems, our experiments with ABMODLEM showed that rules induced from arguments were not able to generalize over too many other examples. After consideration of various rule evaluation measures (see a review [2]), we decided to choose a *Weighted Information Gain* defined (after [1]) as

$$WIG = \frac{|S_1^+|}{|S^+|} * (log_2 p(+|S_1) - log_2 p(+|S))$$

where S denotes a set of learning examples covered by the current condition part of a rule, S_1 is the set of examples covered by the condition rule extended with the evaluated condition; $|S^+|$ (and $|S_1^+|$) is a number of positive examples in S (and S_1); $p(+|S)$ and $p(+|S_1)$ are rule accuracies for examples in S and S_1, respectively. This measure favors candidates that cover many positive examples as well as lead to more accurate rules. According to [2] such a category of measures is inspired by Quinlan's proposal from FOIL algorithm.

3.3 Classification Strategy for ABML

While classifying a new object, conflicts of ambiguous matching of the object's description to many rules may occur. Strategies solving them are usually based on voting rules in the conflict set where rule evaluation measures such as support or accuracy are taken into account. For instance, a set of rules induced by MODLEM was often used with *Strength Method* introduced by Grzymala in [3], where all matched rules vote for a given class according to their strength – support being the number of covered learning examples from the given class

(for other strategies see [10]). The voting strategies may lower the role of argumented rules in classifying new objects as they may be supported by fewer examples (because they are usually built for difficult examples, see discussion in the following section) and could be overvoted by stronger and more general non-argumented rules. However, such rules should receive more attention as they were partly supervised by experts.

Bratko *et al* tried to solve this problem by choosing a single rule according to a new quality measure, which was estimated by their own methods of extreme value correction [8]. However, calculating its parameters is time consuming. Therefore, we looked for simpler methods that would still have a good intuitive meaning. Staying rather with basic support measures, we noticed similar problems in adopting rule classifiers to deal with imbalanced data, where strength of rules from the minority class was additionally amplified [4]. This leads us to proposing the *Average Strengths Method*, which aims to balance the influence of argumented and non-argumented rules while classifying new objects. In this method, the strength of each argumented rule is multiplied by a factor MA defined as:

$$MA = \frac{avg_{r \in R_n} MR(r)}{avg_{r \in R_{arg}} MR(r)}$$

where R_{arg} stands for argumented rules and R_n for non-argumented rules, $MR(r)$ is a strength of rule r being a number of covered learning examples.

4 Identification of Examples for Argumentation

A separate, but crucial problem, is an identification of examples which should be argumented, as it influences the final set of rules. For problems described with a relatively small number of examples, an expert could know them all and determine the necessary examples manually. However, for larger problems or carrying out experiments with many data sets, an automatic selection of examples is necessary. Following some inspiration from earlier machine learning studies on active learning we can assume that focusing on examples difficult or uncertain for standard classifiers may be more promising than taking into account easy examples.

Bratko [8] sketched an idea of an iterative approach based on most frequently misclassified examples in the cross validation procedure. Each step includes identification of only one example misclassified by the rule classifier and the argumentation for this example is used to induce new set of rules. The procedure is repeated until a given stopping criterion is met. However, it is computationally costly, and requires time-consuming cooperation with an expert. We claim that a one-phase solution is neccessary.

We decided to choose a set of examples which were the most frequently misclassified in repeated several times 10-fold cross validation. However, our preliminary experiments showed that for some difficult data sets, still too many examples ranked with the same number of errors were identified. In this case we

decided to co-operate with the expert on selecting a smaller number of examples. For data sets further considered experimentally we established that staying with a small percentage of the data size was sufficient and increasing it did not bring substantial improvements with respect to the classification abilities. While choosing examples for argumentation we also control the distribution of examples among classes (see section 6 for more details).

5 Experiments

The main aim of the experiments is to compare the argumented rule induction algorithm ABMODLEM against its basic, non-argumented origin MODLEM[1]. We want to evaluate how much the argumentation and other modifications (as changing evaluation measure, classification strategies) could influence the structure of the rule set (number of rules, average length of the condition parts) as well as classification abilities. An additional aim of our study is to examine the effect of argumentation on recognition of classes for imbalanced data. In imbalanced data one of the classes (called a *minority class*) contains a much smaller number of examples than the remaining classes.

Table 1. Characteristics of argumented data sets

Data set	No. of examples	Argumented examples (%)	Attributes (Numeric)	Minority class in %	Domain
Zoo	100	6 (6%)	17(1)	4	type of animal
G.Credit	1000	21 (2%)	20(7)	30	admission of credit
Cars	1728	33 (2%)	6(2)	4	car evaluation

For the experiments, data has to be accompanied by the expert's argumentation. Bratko *et al.* in their papers [7] co-operated with real experts in problems from law or medicine. Unfortunately, neither these data sets nor their ABCN2 implementation (including their rule evaluation measure and classification strategy) are publicly available. Thus, we decided to choose data sets from the UCI repository which come from "intuitive" domains for which we were able to provide the argumentation on our own. The following data sets were chosen: *ZOO* – describing species of animals with descriptive attributes, *German Credit* – representing bank credit policy and *Cars* – evaluation of the quality of cars. Moreover, all these data sets contain numerical attributes (for which MODLEM or ABMODLEM is well suited) and are characterized by class imbalance. Their characteristics are given in Table 1. To identify examples for argumentation we used a technique described in section 4. Our argumentation was based on common or encyclopaedic knowledge. However, for some difficult examples we additionally induced rules by various algorithms and analysed the syntax of condition parts for the strongest and accurate patterns covering these examples.

[1] Both algorithms were implemented in Java using WEKA framework.

Table 2. Comparison of MODLEM versus ABMODLEM algorithms

Data set	Algorithm	Classification strategy	Evaluation measure	No. of rules	No. of conditions	Accuracy
ZOO	MODLEM	Strength	Entropy	14	1,31	89,3
	ABMODLEM	Strength	WIG	9	2,19	95,8
	ABMODLEM	AS	Entropy	9	2,19	97,0
	ABMODLEM	**AS**	**WIG**	**9**	**2,19**	**97,0**
Cars	MODLEM	Strength	Entropy	148	4,72	90,6
	ABMODLEM	Strength	WIG	149	4,74	91,5
	ABMODLEM	AS	Entropy	152	4,74	94,8
	ABMODLEM	**AS**	**WIG**	**149**	**4,74**	**95,0**
Credit	MODLEM	Strength	Entropy	172	4,09	71,4
	ABMODLEM	Strength	WIG	123	3,91	74,6,1
	ABMODLEM	AS	Entropy	178	4,06	73,8
	ABMODLEM	**AS**	**WIG**	**123**	**3,91**	**75,6**

We compared the performance of the standard MODLEM versus ABMODLEM with respect to: average rule length, average number of rules in a set and total classification accuracy (see Table 2). We also present in this table results of running algorithm with different rule evaluation measures and classification strategies. *Entropy* denotes the class entropy used in the original formulation of MODLEM, while *WIG* is a new measure described in section 3.1. Then, *Strength* denotes the Grzymala classification strategy, and *AS* stands for our new *Average Strengths Method* – see section 3.3. All evaluations were carried by 10-fold-cross-validation. Argumentation for a given example was used only if the example belonged to a training set. We shift the discussion of these results to the next section, saying only that including argumentation always improved the results.

Table 3. Classification accuracies in classes (%)

ZOO

algorithm	mammal	bird	reptile (M)	fish	amph. (M)	insect	inv.
MODLEM	100,0	100,0	0	100,0	25,0	98,0	80,0
ABMODLEM	100,0	100,0	60,0	100,0	75,0	100,0	100,0

Cars

algorithm	unacc	acc	vgood (M)	good (M)
MODLEM	100,0	79,0	35,0	56,0
ABMODLEM	99,0	95,0	50,0	65,0

Credit

algorithm	bad (M)	good
MODLEM	32,0	90,0
ABMODLEM	38,0	92,0

The last experiments concerned the recognition of particular classes for each data set, where the best configuration of ABMODLEM was compared with MODLEM (see Table 3). As data sets are imbalanced, the minority classes are denoted with *(M)* – however for two data sets two small classes have quite similar cardinalities, so we marked them both.

6 Discussion and Final Remarks

Let us discuss the experimental results. First of all, taking into consideration the argumentation of examples (ABMODLEM), it always led to an improvement of the overall classification accuracy (see the last column of Table 2). Moreover, one can see that ABMODLEM generated a slightly smaller set of rules (except Cars data). We can also say, that the number of argumented examples is not very high, comparing to the size of the data set (less than 10%) - although we did not carry out too many additional experiments on changing this number.

Then, the new measure for evaluating candidates for condition parts of rules (Weighted Information Gain WIG in ABMODLEM) was more useful than class entropy (previously used in non-argumented MODLEM).

Furthermore, the new proposed AS classification strategy always improved the overall classification accuracy compared to the standard strategy where more specific argumented rules could be outvoted by more general non argumented rules being in the same conflict set. We should mention that we also tested yet another classification strategy, which was based on arbitrarily choosing the strongest argumented rule from the conflict set. However, it always decreased the classification accuracy compared to the two above-mentioned strategies, so due to the paper size we have skipped its presentation.

Finally, analysing results from Table 3 we can see that the argumentation influenced the classification in particular classes. For instance, in ZOO data the recognition of the "reptile" minority class increased from 0% to 96%; in Cars for "very good" class that improvement was 15%; and 6% for the "bad credits" class in German Credit. On the one hand, we should add that this could be because arguments were often created just for difficult, misclassified examples that were coming from minority classes in the original set. On the other hand, the observed improvement in the minority classes did not decrease too much recognition of the majority classes (no decrease for ZOO and Credit, a small percentage only for Cars). We should mention that in one additional experiment for German Credit data with changing argumentation we also noticed that choosing arguments only from the minority class worked worse than also argumenting a few majority class examples. In conclusion, the proper argumentation for chosen examples may be seen as a valid method for improving classifier learning from imbalanced data, which according to our best knowledge was not studied before.

We could not directly compare our approach against Bratko *et al.* ABNC2 as it was not available. However, we carried out an additional comparative study of the best variant of ABMODLEM vs. non-argumented RIPPER rule induction and C4.8 tree algorithm (WEKA implementations). For ABMODLEM the

overall accuracy was still better (from 2 to 12% improvement for RIPPER and from 4 to 15% for C4.8, depending on the data) and it worked even better for imbalanced classes (improvement by up to 75%).

Although our research on the generalization of rule induction was inspired by earlier works of Bratko *et al.*, we claim that our experimental results extend the knowledge of usefulness of the ABML paradigm. Moreover, we have proposed new methodological elements improving the construction of rule classifiers and the handling of class imbalance.

Last but not least, the proper choice examples for expert's argumentation is a crucial issue. Although we identified them as the most frequently misclassified examples in repeated cross-validation and it worked well in our experiments, we are not completely satisfied as for two data sets (cars and german credit) we still received too many examples with the same number of errors and had to post-process them. So, it is still an open problem and other methods of selecting examples should be studied. In our on-going research we are currently examining another approach for estimating uncertainty of the classification decision for critical examples. Finally, when the class imbalance is important, this approach should in some way control the distribution of selected examples among classes.

References

1. Flach, P.A.: Invited Tutorial on Rule Induction. Spring School on Intelligent Data Analysis, Palermo (March 2001), http://www.pa.icar.cnr.it/IDAschool
2. Furnkranz, J.: Separate-and-conquer rule learning. Artificial Intelligence Review 13(1), 3–54 (1999)
3. Grzymala-Busse, J.W.: Managing uncertainty in machine learning from examples. In: Proc. of the 3rd Intern. Symposium in Intelligent Systems, pp. 70–94 (1994)
4. Grzymala-Busse, J.W., Stefanowski, J., Wilk, S.z.: A comparison of two approaches to data mining from imbalanced data. Journal of Intelligent Manufacturing 16(6), 565–574 (2005)
5. Liu, B., Hsu, W.: Domain knowledge to support the discovery process - previously discovered knowledge. In: Klosgen, W., Zytkow, J. (eds.) Handbook of Data Mining and Knowledge Discovery, pp. 461–467. Oxford Press, Oxford (2002)
6. Mozina, M., Bratko, I.: Argumentation and machine learning. Research Report: Deliverable 2.1 for the ASPIC project (2004)
7. Mozina, M., Zabkar, J., Bench-Capon, T., Bratko, I.: Argument based machine learning applied to law. Artificial Intelligence and Law 13(1), 53–57 (2005)
8. Mozina, M., Zabkar, J., Bratko, I.: Argument based machine learning. Aritificial Intelligence Journal 171, 922–937 (2007)
9. Stefanowski, J.: Rough set based rule induction techniques for classification problems. In: Proc. 6th European Congress on Intelligent Techniques and Soft Computing, EUFIT 1998, Aachen, September 7-10, vol. 1, pp. 109–113 (1998)
10. Stefanowski, J.: Algorithms of rule induction for knowledge discovery. In: Habilitation Thesis published as Series Rozprawy, vol. 361, pp. 18–21. Poznan University of Technology Press (2001) (in Polish)
11. Stefanowski, J.: On combined classifiers, rule induction and rough sets. In: Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J.W., Orłowska, E., Polkowski, L. (eds.) Transactions on Rough Sets VI. LNCS, vol. 4374, pp. 329–350. Springer, Heidelberg (2007)

Integrating Selective Pre-processing of Imbalanced Data with Ivotes Ensemble

Jerzy Błaszczyński, Magdalena Deckert, Jerzy Stefanowski, and Szymon Wilk

Institute of Computing Science, Poznań University of Technology,
60-965 Poznań, Poland
{jerzy.blaszczynski,magdalena.deckert,jerzy.stefanowski,
szymon.wilk}@cs.put.poznan.pl

Abstract. In the paper we present a new framework for improving classifiers learned from imbalanced data. This framework integrates the SPIDER method for selective data pre-processing with the Ivotes ensemble. The goal of such integration is to obtain improved balance between the sensitivity and specificity for the minority class in comparison to a single classifier combined with SPIDER, and to keep overall accuracy on a similar level. The IIvotes framework was evaluated in a series of experiments, in which we tested its performance with two types of component classifiers (tree- and rule-based). The results show that IIvotes improves evaluation measures. They demonstrated advantages of the abstaining mechanism (i.e., refraining from predictions by component classifiers) in IIvotes rule ensembles.

1 Introduction

Learning classifiers from imbalanced data has received a growing research interest in the last decade. In such data, one of the classes (further called a *minority class*) contains significantly smaller number of objects than the remaining *majority classes*. The imbalanced class distribution causes difficulties for the majority of learning algorithms because they are biased toward the majority classes and objects from the minority class are frequently misclassified, what is not acceptable in many practical applications.

Several methods have been proposed to deal with learning from imbalanced data (see [5,6] for reviews). These methods can be categorized in two groups. The first group includes classifier-independent methods that rely on transforming the original data to change the distribution of classes, e.g., by re-sampling. The other group involves modifications of either learning or classification strategies.

In this paper, we focus on re-sampling techniques. The two well known methods are SMOTE for selective over-sampling of the minority class [3], and NCR for removing objects from the majority classes [8]. Stefanowski and Wilk also proposed a new method to selective pre-processing combining filtering and over-sampling of imbalanced data (called SPIDER) [11]. Experiments showed that it was competitive to SMOTE and NCR [12]. Unfortunately, for some data sets the improvement of the sensitivity for the minority class was associated with

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 148–157, 2010.

too large decrease of specificity for this class (it translated into worse recognition of objects from the majority classes). It affects SPIDER and other methods included in the experiment. In our opinion it is an undesirable property as in many problems it is equally important to improve sensitivity of a classifier induced from imbalanced data and to keep its specificity and overall accuracy at an acceptable level (i.e., both measures should not deteriorate too much comparing to a classifier induced from data without pre-processing). We claim that in general there is a kind of trade off between these measures and too large drop of specificity or accuracy may not be accepted. Thus, our goal is to modify SPIDER in a way that would improve this trade-off.

To achieve it we direct out attention to *adaptive ensemble classifiers* which iteratively construct a set of component classifiers. Such classifiers optimize the overall accuracy, by iteratively learning objects which were difficult to classify in previous iterations. However, as these objects are sampled from the original learning set which is predominated by the majority classes, even misclassified objects may be still biased toward these classes. Our proposition to overcome this problem is using the SPIDER method to transform each sample in succeeding iterations. It should increase the importance of the minority class objects in learning each component classifier. As an ensemble we decided to consider the Ivotes approach introduced by Breiman in [2], as it is already based on a kind of focused sampling of learning objects. Moreover, we have already successfully applied this ensemble with the MODLEM rule induction algorithm [9,10] and we think its classification strategy could be biased toward the minority class with so-called abstaining [1].

A similar idea of using adaptive ensembles was followed in the SMOTEBoost algorithm [4], where the basic SMOTE method was successfully integrated with changing weights of objects inside the AdaBoost procedure. Results reported in the related literature show that Ivotes gives similar classification results as boosting, therefore we hope that our solution will also work efficiently.

The main aim of this paper is to present the new framework for dealing with imbalanced data based on incorporating SPIDER into the Ivotes ensemble. We evaluate its performance experimentally on several imbalanced data sets and we compare it to the performance of single classifiers combined with SPIDER. We consider tree-based and rule-based classifiers induced by the C4.5 and the MODLEM algorithms respectively, as according to previous studies they are sensitive to the class imbalance [11,12].

2 Related Works

We discuss only these re-sampling methods that are most related to our study. For reviews of other approaches see [5,6]. Kubat and Matwin in their paper on one-side sampling claimed that characteristics of mutual positions of objects is a source of difficulty [7]. They focus attention on *noisy* objects located inside the minority class and *borderline* objects. Such objects from the majority classes are removed while keeping the minority class unchanged. The NCR method

introduced in [8], which uses the Edited Nearest Neighbor Rule (ENNR) and removes these objects from the majority classes that are misclassified by its k nearest neighbors. The best representative of focused over-sampling is SMOTE that over-samples the minority class by creating new synthetic objects in the k-nearest neighborhood [3].

However, some properties of these methods are questionable. NCR or one-side-sampling may remove too many objects from the majority classes. As a result improved sensitivity is associated with deteriorated specificity. Random introduction of synthetic objects by SMOTE may be questionable or difficult to justify in some domains. Moreover, SMOTE may blindly "over-generalize" the minority class area without checking positions of the nearest objects from the majority classes, thus increasing overlapping between classes.

Following this criticism Stefanowski and Wilk introduced SPIDER – a new method for selective pre-procesing [11]. It combines removing these objects from the majority classes that may result in misclassification of objects from the minority class, with local over-sampling of these objects from the minority class that are "overwhelmed" by surrounding objects from the majority classes. On the one hand, such filtering is less greedy than the one employed by NCR, and on the other hand, over-sampling is more focused that this used by SMOTE. SPIDER offers three filtering options that impact modification of the minority class and result in changes of increasing degree and scope: *weak amplification*, *weak amplification and relabeling*, and *strong amplification*. More detailed description is given in Section 3.

Finally, let us note that various re-sampling techniques were integrated with ensembles. The reader is referred to a review in [6] that besides SMOTEBoost describes such approaches as DataBoost-IM or special cost-sensitive modifications of AdaBoost.

3 Proposed Framework

Our framework combines selective pre-processing (SPIDER) with an adaptive ensemble of classifiers. We decided to use Ivotes [2] as the ensemble due to reasons given in Section 1. Briefly speaking Ivotes similarly to boosting sequentially adds a new classifier to the current ensemble by analysing its classification performance and partly adapting to objects that are difficult to learn in succeeding iterations. However, unlike boosting it uses different mechanism for focusing learning on these objects (importance sampling) and another final aggregation rule which comes from bagging. We propose to incorporate SPIDER inside this ensemble to obtain a classifier more focused on minority class. However, due to the construction of the ensemble and its general controlling criterion (accuracy) we still expect that it should sufficiently balance the sensitivity and specificity for the minority class.

The resulting Imbalanced Ivotes (shortly called IIvotes) algorithm is presented in Figure 1. In each iteration, IIvotes creates a new training set from LS by *importance sampling*. The rationale for the importance sampling is that the

new training set will contain about equal numbers of incorrectly and correctly classified objects. In this sampling an object is randomly selected with all objects having the same probability of being selected. Then it is classified by an out-of-bag classifier (i.e., ensemble composed of all classifiers which were not learned on the object). If the object is misclassified then it is selected into the new training set S_i. Otherwise, it is sampled into S_i with probability $\frac{e(i)}{1-e(i)}$, where $e(i)$ is a generalization error. Sampling is repeated until n objects are selected. Each S_i is processed by SPIDER. In each iteration, $e(i)$ is estimated by out-of-bag classifier. IIvotes iterates until $e(i)$ stops decreasing.

Algorithm 1. IIvotes

Input : LS – learning set; TS – testing set; n – size of learning data set; LA – learning algorithm; c_{min} – the minority class; k – the number of nearest neighbors; *opt* – pre-processing option of SPIDER
Output: C^* final classifier

Learning phase
while $e(i) < e(i-1)$ **do**
  S_i := `importance sample` of size n from LS
  S_i := `SPIDER` (S_i, c_{min}, k, *opt*) {selective pre-processing of S_i}
  C_i := `LA` (S_i) {construct a base classifier}
  $e(i)$:= estimate `generalization error` by out-of-bag classifier
  $i := i + 1$

Classification phase
foreach $\mathbf{x} \in TS$ **do**
  $C^*(\mathbf{x}) = \arg\max_X \sum_{i=1}^{T}(C_i(\mathbf{x}) = X)$ {the class with maximum number of votes is chosen as a final label for $\mathbf{x}$}

The SPIDER is presented in Figure 2. In the pseudo-code we use the following auxiliary functions (in all these functions we employ the heterogeneous value distance metric (HVDM) [8] to identify the nearest neighbors of a given object):

- `correct`(S, x, k) – classifies object x using its k-nearest neighbors in set S and returns true or false for correct and incorrect classification respectively.
- `flagged`(S, c, f) – identifies and returns a subset of objects from set S that belong to class c that are flagged as f.
- `knn`(S, x, k, c, f) – identifies and returns these objects among the k-nearest neighbors of x in set S that belong to class c and are flagged as f.
- `amplify`(S, x, k, c, f) – amplifies object x by creating its |`knn`(S, x, k, c, f)| copies and adding it to set S (where |.| denotes the cardinality of a set).

SPIDER consists of two main phases – *identification* and *pre-processing*. In the first phase it identifies the "local" characteristics of objects following the the idea of ENNR [8], flags them appropriately, and marks questionable objects from c_{maj} for possible removal. In the second phase , depending on the pre-processing option SPIDER amplifies selected objects from c_{min}, relabels selected

Algorithm 2. SPIDER

Input : DS – data set; c_{min} – the minority class; k – the number of nearest neighbors; *opt* – pre-processing option (weak = weak amplification, relabel = weak amplification and relabeling, strong = strong amplification)
Output: pre-processed DS

c_{maj} := an artificial class combining all the majority classes in DS

Identification phase
foreach $x \in DS$ **do**
  if `correct`(DS, x, k) **then** flag x as safe
  else flag x as noisy
RS := `flagged`(DS, c_{maj}, noisy)

Pre-processing phase
if *opt* = weak $\vee$ *opt* = relabel **then**
  foreach $x \in$ `flagged`(DS, c_{min}, noisy) **do** `amplify`(DS, x, k, c_{maj}, safe)
  if *opt* = relabel **then**
    foreach $x \in$ `flagged`(DS, c_{min}, noisy) **do**
      foreach $y \in$ `knn`(DS, x, k, c_{maj}, noisy) **do**
        change classification of y to c_{min}
        $RS := RS \setminus \{y\}$
else // *opt* = strong
  foreach $x \in$ `flagged`(DS, c_{min}, safe) **do** `amplify`(DS, x, k, c_{maj}, safe)
  foreach $x \in$ `flagged`(DS, c_{min}, noisy) **do**
    if `correct`(DS, x, $k + 2$) **then** `amplify`(DS, x, k, c_{maj}, safe)
    else `amplify`(DS, x, $k + 2$, c_{maj}, safe)
$DS := DS \setminus RS$

questionable objects from c_{maj} (i.e., their class is changed to c_{min}), and finally removes remaining questionable objects from c_{maj} from a resulting data set. Much more thorough description of the method is provided in [11,12].

Let us remark that Ivotes ensembles proved to improve their performance in terms of predictive accuracy with component classifiers that are able to abstain, i. e., they do not classify objects when they are not sufficiently certain [1]. We are interested in checking whether abstaining could also help in classifying objects from the minority class. According to our previous experience [1], abstaining can be implemented by changing classification strategies inside rule ensembles (by refraining from prediction, when the new object is not precisely covered by rules in the component classifiers).

4 Experiments

The main aim of our experiments was to evaluate the ability of the new IIvotes framework to balance the recognition of minority and majority classes. Thus, we compared the performance of IIvotes with three pre-processing options for

SPIDER (weak, relabel and strong – see Figure 2) to the performance of single classifiers combined with the same SPIDER pre-processing. Moreover, for comprehensive comparison we introduced the following baseline classifiers (further denoted as base) – Ivotes ensembles for IIvotes ensembles and single classifiers without any pre-processing for single classifiers with SPIDER.

We constructed all classifiers with two learning algorithms – C4.5 (J48 from WEKA) for decision trees and MODLEM for decision rules (MODLEM is described in [9,10] and applied together with Grzymala's LERS strategy for classifying new objects [**?**]). Both algorithms were run without prunning to get more precise description of the minority class. SPIDER was used with $k = 3$ neighbors and the size of sample n in IIvotes was set to 50% based on our experience from previous experiments. In case of rule ensembles, besides the basic construction, we additionally tested a version with abstaining of component classifiers [1]. All algorithms were implemented in Java using WEKA.

Table 1. Characteristics of data sets

Data set	Objects	Attributes	Minority class	Imbalance ratio
abdominal-pain	723	13	positive	27.94%
balance-scale	625	4	B	7.84%
breast-cancer	286	9	recurrence_events	29.72%
bupa	345	6	sick	42.03%
car	1728	6	good	3.99%
cleveland	303	13	positive	11.55%
cmc	1473	9	long-term	22.61%
ecoli	336	7	imU	10.42%
german	666	20	bad	31.38%
haberman	306	3	died	26.47%
hepatitis	155	19	die	20.65%
pima	768	8	positive	34.90%
transfusion	748	4	yes	23.80%

The experiments were carried out on 13 data sets listed in Table 1. They either came from the UCI repository[1] or from our medical case studies (abdominal pain). We selected data sets that were characterized by varying degrees of imbalance and that were used in other related works.

All experiments were run with a stratified 10-fold cross-validation repeated five times. Besides recording average values of sensitivity, specificity and overall accuracy we also used G-mean – geometric mean of sensitivity and specificity – to evaluate the balance between these two measures [7]. Although AUC measure could also be used, we stay with G-mean as it sufficiently characterizes deterministic (non-threshold) classifiers and it has been used in many studies on learning from imbalanced data. G-mean for tree- and rule-based classifiers are

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html

presented in Table 2 and 3. Moreover, in Table 4 we show G-mean for IIvotes rule ensembles with abstaining.

Table 2. G-mean for tree-based classifiers

Data set	Single C4.5				Ivotes / IIvotes + C4.5			
	Base	Weak	Relabel	Strong	Base	Weak	Relabel	Strong
abdominal-pain	0.7812	0.7859	0.7807	0.7919	0.8052	0.8216	0.8239	0.8157
balance-scale	0.0249	0.2648	0.3646	0.2562	0.0881	0.4584	0.3827	0.5232
breast-cancer	0.5308	0.5487	0.5824	0.5602	0.5467	0.6068	0.5868	0.5683
bupa	0.6065	0.6032	0.5628	0.6037	0.6635	0.6804	0.7019	0.6612
car	0.8803	0.9261	0.8603	0.9111	0.8093	0.9149	0.8945	0.9171
cleveland	0.3431	0.4531	0.5052	0.4079	0.2759	0.4411	0.3914	0.4896
cmc	0.5533	0.6378	0.6175	0.6310	0.5813	0.6620	0.6439	0.6547
ecoli	0.6924	0.7728	0.7788	0.7852	0.7443	0.8383	0.8122	0.8462
german	0.5828	0.6114	0.6113	0.6086	0.5947	0.6738	0.6615	0.6662
haberman	0.5375	0.6089	0.6083	0.6118	0.4750	0.6256	0.6085	0.6167
hepatits	0.5386	0.5971	0.6518	0.5534	0.7115	0.7642	0.7466	0.7422
pima	0.6949	0.6978	0.7046	0.6986	0.7255	0.7401	0.7340	0.7343
transfusion	0.5992	0.6276	0.6317	0.6252	0.5181	0.6492	0.6523	0.6309

Table 3. G-means for rule-based classifiers (rule ensembles without abstaining)

Data set	Single MODLEM				Ivotes / IIvotes + MODLEM			
	Base	Weak	Relabel	Strong	Base	Weak	Relabel	Strong
abdominal-pain	0.7731	0.7968	0.7914	0.7946	0.7933	0.8321	0.8183	0.8278
balance-scale	0.0000	0.1913	0.1613	0.1722	0.0634	0.1125	0.0729	0.1454
breast-cancer	0.5008	0.5612	0.5104	0.5687	0.4748	0.5571	0.5462	0.5837
bupa	0.6502	0.5969	0.6725	0.5989	0.6703	0.6800	0.7002	0.6920
car	0.8978	0.9547	0.9404	0.9489	0.9021	0.9722	0.9638	0.9779
cleveland	0.3292	0.4360	0.3738	0.4673	0.1063	0.3307	0.2364	0.3628
cmc	0.5171	0.6320	0.5770	0.6218	0.5304	0.6660	0.6029	0.6575
ecoli	0.6502	0.7736	0.6655	0.7763	0.6140	0.7879	0.7233	0.7969
german	0.5499	0.6147	0.5719	0.6337	0.5133	0.6272	0.5838	0.6382
haberman	0.4588	0.5382	0.4790	0.5702	0.4345	0.5403	0.4807	0.5570
hepatits	0.6140	0.6861	0.6082	0.6482	0.6142	0.6637	0.6702	0.6817
pima	0.6576	0.7190	0.6832	0.7148	0.6510	0.7356	0.6944	0.7271
transfusion	0.5128	0.6153	0.5422	0.6103	0.4848	0.6100	0.5693	0.6239

For pairwise comparison of classifiers over all data sets we used the Wilcoxon Signed Ranks Test (confidence $\alpha = 0.05$). Considering the results of GM for tree-based classifiers (see Table 2) all single classifiers with any SPIDER preprocessing and all IIvotes ensembles were always significantly better than their baseline versions. Also all IIvotes ensembles were significantly better than single classifiers with a corresponding SPIDER option. Moreover, the IIvotes ensembles

Table 4. G mean for rule ensembles with abstaining

Data set	Ivotes / IIvotes + MODLEM			
	Base	Weak	Relabel	Strong
abdominal-pain	0.7995	0.8345	0.8284	0.8400
balance-scale	0.0625	0.1637	0.0878	0.2470
breast-cancer	0.5203	0.5776	0.5716	0.5886
bupa	0.7045	0.7058	0.7124	0.6933
car	0.9426	0.9743	0.9780	0.9834
cleveland	0.2361	0.4028	0.3232	0.4420
cmc	0.5630	0.6684	0.6353	0.6709
ecoli	0.7098	0.8077	0.7706	0.8245
german	0.6055	0.6852	0.6512	0.6885
haberman	0.4944	0.5704	0.5044	0.5625
hepatits	0.6759	0.7047	0.7005	0.7240
pima	0.7049	0.7507	0.7306	0.7430
transfusion	0.5331	0.6212	0.5851	0.6324

Table 5. Overall accuracy [%] for tree-based classifiers

Data set	Single C4.5				Ivotes / IIvotes + C4.5			
	Base	Weak	Relabel	Strong	Base	Weak	Relabel	Strong
abdominal-pain	82.84	77.45	76.87	77.92	85.20	81.77	83.21	81.30
balance-scale	78.65	73.34	72.99	73.81	84.67	80.83	80.64	79.07
breast-cancer	65.40	59.12	59.89	58.91	66.71	63.36	62.87	56.78
bupa	65.56	60.18	56.84	60.20	69.39	67.42	69.86	65.28
car	93.99	95.04	94.20	94.78	92.89	92.91	93.02	92.88
cleveland	82.25	81.52	80.98	81.86	85.08	83.83	83.70	83.70
cmc	49.25	49.27	46.58	48.46	51.57	50.69	50.98	49.45
ecoli	91.91	90.55	89.23	91.50	92.80	91.90	92.68	91.19
german	66.00	65.44	63.33	65.50	71.05	71.86	73.06	70.54
haberman	70.08	61.26	59.87	60.88	92.06	90.00	90.65	90.56
hepatits	78.47	75.93	76.16	73.74	72.55	66.67	67.39	62.88
pima	73.96	69.42	69.63	69.66	84.39	83.10	83.10	82.84
transfusion	77.75	66.15	65.61	60.85	75.65	74.14	74.24	73.23

with the weak and strong options were always superior to any single classifier with any SPIDER option. After comparing pairs of Iivotes ensembles we were not able to reject the null hypothesis on equal performance for the weak and strong options, however, both of them were better than relabel.

We obtained similar results of the Wilcoxon test for rule ensembles with abstaining (see Table 4 and the left part of Table 3), although the superiority of the IIvotes ensemble with relabel over the single classifier with the same SPIDER option is slightly smaller ($p = 0.03$ while previously it was close to 0.01). Furthermore, the IIvotes ensembles with the strong option was nearly significant better than the IIvotes ensemble with the weak option ($p = 0.054$). Considering the

results for the non-abstaining ensembles (Table 3), the Wilcoxon test revealed that the IIvotes ensembles weak and strong option were significantly better than the single classifiers with the same pre-processing option, however, the advantage was smaller than for the variant with abstaining.

While analysing the sensitivity measure alone we noticed that SPIDER combined with single classifiers is still better than IIvotes for many data sets (due to page limits we cannot show more tables with detailed results). Finally, considering the overall accuracy results of Wilcoxon test show that IIvotes integrated with SPIDER is always better than its single classifier version (see Table 5 for trees, results for rules are analogous).

5 Final Remarks

In this paper we proposed a new framework that integrates the SPIDER method for selective data pre-processing into the Ivotes ensemble. This integration aims at obtaining a better trade-off between sensitivity and specificity for the minority class than SPIDER combined with a single classifier.

Experimental results showed that the proposed IIvotes framework led to significantly better values of G-mean than single classifier combined with SPIDER. Despite improving the sensitivity of the minority, a satisfactory value of specificity is preserved, what was not achieved by SPIDER alone and other related re-sampling techniques (previous experiments [12] showed that also NCR and to some extent SMOTE suffered from decreasing specificity). However, an integration with single classifiers could be still attractive if one wants to improve sensitivity only without caring about other measures.

After comparing possible pre-processing options of the IIvotes framework we can say that weak and strong amplification (particularly the latter) are more efficient than relabel. Moreover, IIvotes was successful in keeping the overall accuracy at an acceptable level, comparable to baseline classifiers. Let us notice that using the standard version of Ivotes ensemble was not successful – G-mean did not differ significantly from values reported for single classifiers. We expect that even using a re-sampling filter to transform the whole data before constructing the ensemble is also a worse solution than integrating it inside the ensemble – see the discussion in [4].

Abstaining turned out to be a useful extension of rule ensembles as it improved their performance with respect to all considered measures. Let us remind that component classifiers in the IIvotes ensemble use unordered rule sets and the LERS classification strategy, where the conflict caused by matching a classified object to multiple rules is solved by voting with rule support. This strategy is biased toward rules from the majority classes as they are stronger and more general than rules from the minority class. This is the reason why objects from the minority class are more likely to be misclassified. Thus, refraining from making wrong predictions in some classifiers gives a chance to other component classifiers (that are more expertized for the new object) to have greater influence on the final outcome of the rule ensemble.

Our future research in processing imbalance data with rule-based ensemble classifier covers two topics. The first one is studying the impact of changing the control criterion in the ensemble from general error (or accuracy) toward measures typical for imbalanced data. The second one is exploitation of other classification strategies which could improve the role of rules for the minority class and combining them with SPIDER.

References

1. Błaszczyński, J., Stefanowski, J., Zając, M.: Ensembles of Abstaining Classifiers Based on Rule Sets. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) ISMIS 2009. LNCS, vol. 5722, pp. 382–391. Springer, Heidelberg (2009)
2. Breiman, L.: Pasting small votes for classification in large databases and on-line. Machine Learning 36, 85–103 (1999)
3. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: Synthetic Minority Over-sampling Technique. J. of Artifical Intelligence Research 16, 341–378 (2002)
4. Chawla, N., Lazarevic, A., Hall, L., Bowyer, K.: SMOTEBoost: Improving Prediction of the Minority Class in Boosting. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 107–119. Springer, Heidelberg (2003)
5. Chawla, N.: Data mining for imbalanced datasets: An overview. In: Maimon, O., Rokach, L. (eds.) The Data Mining and Knowledge Discovery Handbook, pp. 853–867. Springer, Heidelberg (2005)
6. He, H., Garcia, E.: Learning from imbalanced data. IEEE Transactions on Data and Knowledge Engineering 21(9), 1263–1284 (2009)
7. Kubat, M., Matwin, S.: Addresing the curse of imbalanced training sets: one-side selection. In: Proc. of the 14th Int. Conf. on Machine Learning, ICML 1997, pp. 179–186 (1997)
8. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. Tech. Report A-2001-2, University of Tampere (2001)
9. Stefanowski, J.: The rough set based rule induction technique for classification problems. In: Proc. of the 6th European Conf. on Intelligent Techniques and Soft-Computing, EUFIT 1998, pp. 109–113 (1998)
10. Stefanowski, J.: On combined classifiers, rule induction and rough sets. In: Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J.W., Orłowska, E., Polkowski, L. (eds.) Transactions on Rough Sets VI. LNCS, vol. 4374, pp. 329–350. Springer, Heidelberg (2007)
11. Stefanowski, J., Wilk, S.: Improving Rule Based Classifiers Induced by MODLEM by Selective Pre-processing of Imbalanced Data. In: Proc. of the RSKD Workshop at ECML/PKDD, Warsaw, pp. 54–65 (2007)
12. Stefanowski, J., Wilk, S.: Selective Pre-processing of Imbalanced Data for Improving Classification Performance. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2008. LNCS, vol. 5182, pp. 283–292. Springer, Heidelberg (2008)

Learning from Imbalanced Data in Presence of Noisy and Borderline Examples

Krystyna Napierała, Jerzy Stefanowski, and Szymon Wilk

Institute of Computing Science, Poznań University of Technology,
ul. Piotrowo 2, 60–965 Poznań, Poland
{krystyna.napierala,jerzy.stefanowski,szymon.wilk}@cs.put.poznan.pl

Abstract. In this paper we studied re-sampling methods for learning classifiers from imbalanced data. We carried out a series of experiments on artificial data sets to explore the impact of noisy and borderline examples from the minority class on the classifier performance. Results showed that if data was sufficiently disturbed by these factors, then the focused re-sampling methods – NCR and our SPIDER2 – strongly outperformed the oversampling methods. They were also better for real-life data, where PCA visualizations suggested possible existence of noisy examples and large overlapping ares between classes.

1 Introduction

In some real-life problems, the distribution of examples in classes is highly imbalanced, which means that one of the classes (further called a *minority class*) includes much smaller number of examples than the other majority classes [1,3]. Class imbalance constitutes a difficulty for most learning algorithms, which assume even class distribution and are biased toward learning and recognition of the majority classes. As a result, minority examples tend to be misclassified.

This problem has been intensively researched in the last decade and several methods have been proposed (see [1,3] for a review). They are usually divided into solutions on the data level and the algorithmic level. Solutions on the data level are classifier-independent and consist in transforming an original data distribution to change the balance between classes, e.g., by re-sampling techniques. Solutions on the algorithmic level involve modification of either learning or classification strategies. Some researchers also generalize ensembles or transform the imbalance problem to cost sensitive learning [3].

In this paper we are interested in focused re-sampling techniques, which modify the class distribution taking into account local characteristics of examples. Inspired by [6] we distinguish between safe, borderline and noisy examples. *Borderline examples* are located in the area surrounding class boundaries, where the minority and majority classes overlap. *Safe examples* are placed in relatively homogeneous areas with respect to the class label. Finally, by *noisy examples* we understand individuals from one class occurring in safe areas of the other class. We claim that the distribution of borderline and noisy examples causes difficulties for learning algorithms, thus we focus our interest on careful processing of these examples.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 158–167, 2010.

Our study is related to earlier works of Stefanowski and Wilk on selective pre-processing with the SPIDER (Selective Preprocessing of Imbalanced Data) method [8,9]. This method employs the Edited Nearest Neighbor Rule (ENNR) to identify the local characteristic of examples, and then it combines removing the majority class objects that may result in misclassifying objects from the minority class with local over-sampling of these objects from the minority class that are "overwhelmed" by surrounding objects from the majority classes. Experiments showed that this method improved the recognition of the minority class and was competitive to the most related approaches SMOTE and NCR [9]. The observed improvements varied over different imbalanced data sets, therefore, in this study we have decided to explore conditions, where the SPIDER method could be more efficient than simpler re-sampling methods. To achieve this goal we have planned controlled experiments with special artificial data sets.

According to related works many experiments were conducted on real-life data sets (e.g., coming from UCI). The most well known studies with artificial data are the works of Japkowicz [4,5], who showed that simple class imbalance ratio was not the main difficulty. The degradation of performance was also related to other factors, mainly to small disjuncts, i.e., the minority class being decomposed into many sub-clusters with very few examples. Other researchers also explored the effect of overlapping between imbalanced classes – more recent experiments on artificial data with different degrees of overlapping also showed that overlapping was more important than the overall imbalance ratio [2].

Following these motivations we prepare our artificial data sets to analyze the influence of the presence and frequency of the noisy and borderline examples. We also plan to explore the effect of the decomposition of this class into smaller sub-clusters and the role of changing decision boundary between classes from linear to non-linear shapes. The main aim of our study is to examine which of these factors were critical for the performance of the methods dealing with imbalanced data. In the experiments we compare the performance of the SPIDER method and the most related focused re-sampling NCR method with the oversampling methods suitable to handle class decomposition [5] and the basic versions of tree- or rule-based classifiers.

2 Focused Re-sampling Methods

Here we discuss only these re-sampling methods that are used in our experiments. The simplest oversampling randomly replicates examples from the minority class until the balance with cardinality of the majority classes is obtained. Japkowicz proposed an advanced oversampling method (*cluster oversampling*) that takes into account not only *between-class imbalance* but also *within-class imbalance*, where classes are additionally decomposed into smaller sub-clusters [5]. First, random oversampling is applied to individual clusters of the majority classes so that all the sub-clusters are of the same size. Then, minority class clusters are processed in the same way until class distribution becomes balanced. This approach was successfully verified in experiments with decomposed classes [5]. In

[6] one side sampling was proposed, where noisy and borderline examples from the majority class are removed and the minority class remains unchanged. A similar idea was employed in the NCR (Neighborhood Cleaning Rule) method [7]. NCR applies ENNR to identify and remove noisy and borderline examples from the majority classes. NCR demonstrates a few undesirable properties (e.g., improvement of sensitivity at the cost of specificity) and their critical analysis has become a starting point for the family of the SPIDER methods. Following [6], they rely on the local characteristics of examples discovered by analyzing their k-nearest neighbors. SPIDER2 is presented in Alg. 1. To simplify the notation we do not distinguish between noisy and borderline examples and refer to them simply as not-safe.

Algorithm 1. SPIDER2

Input : DS – data set; c_{min} – the minority class; k – the number of nearest neighbors; *relabel* – relabeling option (yes, no); *ampl* – amplification option (no, weak, strong)
Output: preprocessed DS

c_{maj} := an artificial class combining all classes except c_{min}
foreach $x \in$ `class`(DS, c_{maj}) **do**
  if `correct`(DS, x, k) **then** flag x as safe
  else flag x as not-safe
RS := `flagged`(DS, c_{maj}, not-safe)
if *relabel* **then**
  foreach $y \in RS$ **do**
    change classification of y to c_{min}
    $SR := SR \setminus \{y\}$
else $DS := DS \setminus RS$
foreach $x \in$ `class`(DS, c_{min}) **do**
  if `correct`(DS, x, k) **then** flag x as safe
  else flag x as not-safe
if *ampl* = weak **then**
  foreach $x \in$ `flagged`(DS, c_{min}, not-safe) **do** `amplify`(DS, x, k)
else if *ampl* = strong **then**
  foreach $x \in$ `flagged`(DS, c_{min}, not-safe) **do**
    if `correct`(DS, x, $k + 2$) **then** `amplify`(DS, x, k)
    else `amplify`(DS, x, $k + 2$)

In the pseudo-code we use the following auxiliary functions: `correct`(S, x, k) – classifies example x using its k-nearest neighbors in set S and returns true or false for correct and incorrect classification respectively; `class`(S, c) – returns a subset of examples from S that belong to class c; `flagged`(S, c, f) – returns a subset of examples from S that belong to class c and are flagged as f; `knn`(S, x, k, c) – identifies and returns these examples among the k-nearest neighbors of x in S that belong to class c; `amplify`(S, x, k) – amplifies example x by creating its n-copies and adding them to S. n is calculated as $|$`knn`(DS, x, k, c_{maj})$|$ – $|$`knn`(DS, x, k, c_{min})$| + 1$. In these functions we employ the heterogeneous value distance metric (HVDM) [7] to identify the nearest neighbors of a given example.

SPIDER2 consists of two phases corresponding to pre-processing of c_{maj} and c_{min} respectively. In the first phase (lines 2–10) it identifies the characteristics of examples from c_{maj}, and depending on the *relabel* option it either removes or relabels noisy examples from c_{maj} (i.e., changes their classification to c_{min}). In the second phase (lines 11–19) it identifies the characteristic of examples from c_{min} considering changes introduced in the first phase. Then, noisy examples from c_{min} are amplified (by replicating them) according to the ampl option. This two-phase structure is a major difference from the first SPIDER version [8], which first identified the nature of examples and then simultaneously processed c_{maj} and c_{min}. As we noticed in [9] such processing could result in too extensive modifications in some regions of c_{maj} and deteriorated specifity – this drawback has been addressed in SPIDER2. Minor differences include the scope of relabeling noisy examples from c_{maj} and the degree of amplifying noisy examples from c_{min}.

3 Experiments with Artificial Data Sets

3.1 Preparation of Data Sets

Following the discussion in Section 1 on the factors influencing the performance of classifiers learned from imbalanced data, we decided to prepare artificial data sets in order to control these factors. We focused on binary classification problems (the minority vs. the majority class) with examples randomly and uniformly distributed in the two-dimensional space (both attributes were real-valued).

We considered three different shapes of the minority class: *subclus*, *clover* and *paw*, all surrounded uniformly by the majority class. In *subclus*, examples from the minority class are located inside rectangles following related works on small disjuncts [4]. *Clover* represents a more difficult, non-linear setting, where the minority class resembles a flower with elliptic petals (Fig. 1 shows *clover* with 5 petals). Finally, in *paw* the minority class is decomposed into 3 elliptic subregions of varying cardinalities, where two subregions are located close to each other, and the remaining smaller sub-region is separated (see Fig. 2). Such a shape should better represent real-life data than *clover*. Moreover, both *clover* and *paw* should be more difficult to learn than simple circles that were considered in some related works.

We generated multiple data sets with different numbers of examples (ranging from 200 to 1200) and imbalance ratios (from 1:3 to 1:9). Additionally, following Japkowicz's research on small disjuncts [4], we considered a series of the *subclus* and *clover* shapes with the number of sub-regions ranging from 1 to 5, and from 2 to 5 respectively. In a preliminary experiment we used tree- and rule-based classifiers on these data sets. Due to the space limit, we are not able to present the complete results. Let us only comment that they are consistent with the observations reported in [5] – increasing the number of sub-regions combined with decreasing the size of a data set degraded the performance of a classifier.

According to the results of the preliminary experiment we finally selected a group of data sets with 800 examples, the imbalance ratio of 1:7, and 5 subregions for the *subclus* and *clover* shapes. All these sets presented a significant

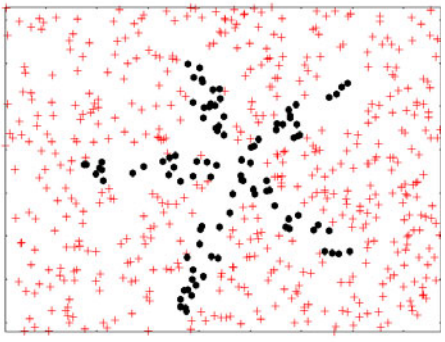


Fig. 1. Clover data set

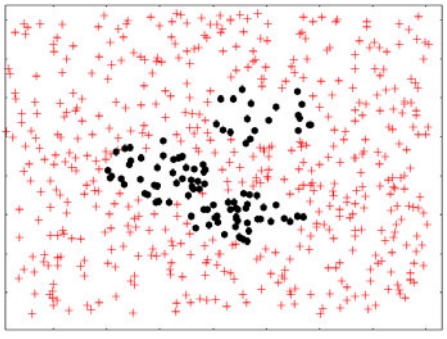


Fig. 2. Paw data set

challenge for a stand-alone classifier. We also observed similar behavior for data sets with 600 examples, but due to space limit we did not describe these data sets in the paper.

3.2 Disturbing Borders of the Minority-Class Subregions

In the first series of experiments we studied the impact of disturbing the borders of sub-regions in the minority class. We simulated it by increasing the ratio of borderline examples from the minority class subregions. We changed this ratio (further called the *disturbance ratio*) from 0 to 70%. The width of the borderline overlapping areas was comparable to the width of the sub-regions. We employed rule- and tree-based classifiers induced with the MODLEM and C4.5 algorithms, as they had been used in earlier studies [9] and had shown to be sensitive to the class imbalance. Both algorithms were run without prunning to get more precise description of the minority class.

The constructed classifiers were combined with the following pre-processing methods: random oversampling (RO), cluster oversampling (CO), NCR and SPIDER2 (SP2). Cluster oversampling was limited to the minority class, and our method was used with relabeling and strong amplification (*relabel* = yes, *ampl* = strong – see Section 2 for details) as such combination performed best in our earlier studies. For baseline results (Base), we ran both classifiers without any pre-processing. As evaluation measures we used sensitivity and specificity for the minority class, their geometric mean (G-mean), and the overall accuracy. We chose G-mean over AUC because it was more intuitive and suited to deterministic rule- and tree-based classifiers. All measures were estimated by 10-fold cross validation repeated 5 times.

Table 1 presents sensitivity recorded for data sets of different shapes (*subclus*, *clover* and *paw*) and different degrees of disturbance (0, 30, 50 and 70%). Increasing this degree strongly deteriorated the performance of both baseline classifiers. Pre-processing always improved performance in comparison to Base. RO and CO performed comparably on all data sets, and on non-disturbed data

Table 1. Sensitivity for artificial data sets with varying degree of the disturbance ratio

Data set	MODLEM					C4.5				
	Base	RO	CO	NCR	SP2	Base	RO	CO	NCR	SP2
subclus-0	0.8820	0.8820	0.9040	0.8640	0.8800	0.9540	0.9500	0.9500	0.9460	0.9640
subclus-30	0.5600	0.5520	0.5500	0.5540	0.5540	0.4500	0.6840	0.6720	0.7160	0.7720
subclus-50	0.3400	0.3580	0.3960	0.5300	0.4360	0.1740	0.6160	0.6000	0.7020	0.7700
subclus-70	0.1980	0.2380	0.2600	0.4300	0.3900	0.0000	0.6380	0.7000	0.5700	0.8300
clover-0	0.5720	0.5740	0.6060	0.6380	0.6560	0.4280	0.8340	0.8700	0.4300	0.4860
clover-30	0.4300	0.4300	0.4520	0.5700	0.5000	0.1260	0.7180	0.7060	0.5820	0.7260
clover-50	0.2860	0.3420	0.3380	0.5420	0.4040	0.0540	0.6560	0.6960	0.4460	0.7700
clover-70	0.2100	0.2520	0.2740	0.5100	0.3700	0.0080	0.6340	0.6320	0.5460	0.8140
paw-0	0.8320	0.8460	0.8560	0.8640	0.8180	0.5200	0.9140	0.9000	0.4900	0.5960
paw-30	0.6100	0.6260	0.6180	0.6660	0.6440	0.2640	0.7920	0.7960	0.8540	0.8680
paw-50	0.4560	0.5000	0.4980	0.6260	0.5500	0.1840	0.7480	0.7200	0.8040	0.8320
paw-70	0.2880	0.3700	0.3600	0.5900	0.4740	0.0060	0.7120	0.6800	0.7460	0.8780

sets they often over-performed NCR and SP2. On more difficult sets (disturbance = 50–70%) neighbor-based methods (NCR and SP2) were better than oversampling. Finally, MODLEM worked better with NCR, while C4.5 with SP2, especially on more difficult data sets.

In terms of specificity, Base performed best as expected. As previously, RO and CO were comparable and they were the second ones. Moreover, the relationship between NCR and SP2 was dependent on the induction algorithm – NCR performed better than SP2 when combined with C4.5, and for MODLEM SP2 won over NCR. However, considering G-mean (see Table 2), NCR and SP2 were better than oversampling methods. Finally, linear rectangle shapes (*subclus*) were easier to learn than non-linear ones (*clover* or *paw*). We are aware that tree- and rule-based classifiers are known to be sensitive to non-linear decision boundaries, and in future research we plan to study other classifiers (e.g., support vector machines) as well.

3.3 Impact of Different Types of Testing Examples

In the second series of experiments we concentrated on the impact of noisy examples from the minority class, located outside the borderline area, on the performance of a classifier. To achieve this, we introduced new noisy examples (single and pairs) and denoted them with C. Similarly to the first series of experiments we used data sets of three shapes (*subclus*, *clover* and *paw*), 800 examples and the imbalance ratio of 1:7. We also employed rule- and tree-based classifiers combined with the same pre-processing methods. However, we changed the 10-fold cross validation to the train-test verification in order to ensure that learning and testing sets had similar distributions of the C noise. In each training set 30% of the minority class examples were safe examples located inside subregions, 50% were located in the borderline area (we denote them with B), and the remaining 20% constituted the C noise.

Table 2. G-mean for artificial data sets with varying degree of the disturbance ratio

Data set	MODLEM					C4.5				
	Base	RO	CO	NCR	SP2	Base	RO	CO	NCR	SP2
subclus-0	0.9373	0.9376	0.9481	0.9252	0.9294	0.9738	0.9715	0.9715	0.9613	0.9716
subclus-30	0.7327	0.7241	0.7242	0.7016	0.7152	0.6524	0.7933	0.7847	0.7845	0.8144
subclus-50	0.5598	0.5648	0.6020	0.6664	0.6204	0.3518	0.7198	0.7113	0.7534	0.7747
subclus-70	0.4076	0.4424	0.4691	0.5957	0.5784	0.0000	0.7083	0.7374	0.6720	0.7838
clover-0	0.7392	0.7416	0.7607	0.7780	0.7908	0.6381	0.8697	0.8872	0.6367	0.6750
clover-30	0.6361	0.6366	0.6512	0.7221	0.6765	0.2566	0.7875	0.7652	0.6758	0.7686
clover-50	0.5066	0.5540	0.5491	0.6956	0.6013	0.1102	0.7453	0.7570	0.6184	0.7772
clover-70	0.4178	0.4658	0.4898	0.6583	0.5668	0.0211	0.7140	0.7027	0.6244	0.7665
paw-0	0.9041	0.9126	0.9182	0.9184	0.8918	0.6744	0.9318	0.9326	0.6599	0.7330
paw-30	0.7634	0.7762	0.7701	0.7852	0.7780	0.3286	0.8374	0.8334	0.8527	0.8337
paw-50	0.6587	0.6863	0.6865	0.7517	0.7120	0.3162	0.8013	0.7858	0.8200	0.8075
paw-70	0.5084	0.5818	0.5691	0.7182	0.6506	0.0152	0.7618	0.7472	0.7824	0.8204

For each training set we prepared 4 testing sets containing the following types of examples from the minority class: only safe examples, only B examples, only C examples, and B and C examples combined together (BC). Results are presented in Table 3. They clearly show that for the "difficult" noise (C or BC) SP2 and in most cases NCR were superior to RO, CO and Base. SP2 was also comparable to RO and CO in case of safe and (sometimes) B examples.

4 Experiments on Real-Life Data Sets

The goal of the third series of experiments was to discover the differences between those real-life data sets where NRC and SPIDER2 were superior to oversampling, and those, for which there was no such advantage. Moreover, we wanted to relate these differences to the factors explored in the previous experiments (see Section 3.1 and 3.2).

Experiments in this series were conducted on imbalanced data sets that we had used in our previous study [9]. They came either from the UCI repository or

Table 3. Sensitivity for artificial data sets with different types of testing examples

Data set	MODLEM					C4.5				
	Base	RO	CO	NCR	SP2	Base	RO	CO	NCR	SP2
subcl-safe	0.5800	0.5800	0.6200	0.7800	0.6400	0.3200	0.8400	0.8600	0.9800	1.0000
subcl-B	0.8400	0.8400	0.8400	0.8600	0.8400	0.0000	0.8200	0.8400	0.3600	0.9200
subcl-C	0.1200	0.1000	0.1600	0.2400	0.2600	0.0000	0.5400	0.0000	0.0000	0.5200
subcl-BC	0.4800	0.4700	0.5000	0.5500	0.5500	0.0000	0.6800	0.4200	0.1800	0.7200
clover-safe	0.3000	0.3800	0.4400	0.7000	0.6000	0.0200	0.9600	0.9200	0.0400	0.9800
clover-B	0.8400	0.8200	0.8200	0.8400	0.8600	0.0400	0.9400	0.9200	0.0400	0.9400
clover-C	0.1400	0.0800	0.1400	0.2400	0.3600	0.0000	0.3000	0.0200	0.0000	0.4000
clover-BC	0.4900	0.4500	0.4800	0.5400	0.6100	0.0200	0.6200	0.4700	0.0200	0.6700
paw-safe	0.8400	0.9200	0.8400	0.8400	0.8000	0.4200	0.9000	0.9600	0.7400	1.0000
paw-B	0.8800	0.8800	0.8600	0.8800	0.9000	0.1400	0.9000	0.9000	0.4000	0.9200
paw-C	0.1600	0.1400	0.1200	0.2600	0.1600	0.0400	0.2000	0.0000	0.0000	0.3400
paw-BC	0.5200	0.5100	0.4900	0.5700	0.5300	0.0900	0.5500	0.4500	0.2000	0.6300

Table 4. Sensitivity for real data sets

Data set	MODLEM					C4.5				
	Base	RO	NCR	SP1	SP2	Base	RO	NCR	SP1	SP2
Acl	0.8050	0.8050	0.9000	0.8250	0.8350	0.8550	0.8400	0.9200	0.8500	0.8450
Breast can.	0.3186	0.3430	0.6381	0.5386	0.5983	0.3867	0.4683	0.6478	0.5308	0.5611
Bupa	0.5199	0.5931	0.8734	0.8047	0.8580	0.4910	0.5720	0.7549	0.6995	0.7487
Cleveland	0.0850	0.1717	0.3433	0.2350	0.2300	0.2367	0.2383	0.3983	0.3017	0.3067
Ecoli	0.4000	0.5400	0.6833	0.6367	0.6217	0.5800	0.5567	0.7583	0.6900	0.7100
Haberman	0.2397	0.2961	0.6258	0.4828	0.5431	0.4103	0.6069	0.6081	0.6600	0.6775
Hepatitis	0.3833	0.4017	0.4550	0.4367	0.4867	0.4317	0.5583	0.6217	0.4750	0.5633
New thyr.	0.8117	0.8733	0.8417	0.8650	0.8867	0.9217	0.9217	0.8733	0.9133	0.8917
Pima	0.4853	0.5206	0.7933	0.7377	0.8188	0.6013	0.6512	0.7678	0.7146	0.7655

from our medical case studies (*acl*). As in the previous two series of experiments, we used C4.5 and MODLEM without prunning to induce classifiers and combined them we the pre-processing methods listed in Section 3.2. We only had to exclude cluster oversampling due to difficulties with defining the proper number of sub-clusters in the minority class. Moreover, for comprehensive comparison we included the first version of SPIDER with strong amplification (SP1). Evaluation measures were estimated in 10-fold cross validation repeated 5 times and the results for sensitivity are given in Table 4.

We used the Wilcoxon Signed Ranks Test (with confidence $\alpha = 0.05$) for pairwise comparison of pre-processing methods over all data sets. For MODLEM, all the pre-processing methods outperformed Base. The same conclusion applied to C4.5 with the exception of RO. Moreover, SP2 outperformed SP1, and differences between NCR and SP2 were not significant according to the test. Although NCR demonstrated slightly better sensitivity, its specificity (not reported here) was lower than for SP2.

When examining the performance of pre-processing methods on individual data sets we found some (e.g. *new thyroid*) for which all methods were comparable. Moreover, for data sets like *acl*, the advantage of SP2 or NCR over RO and CO was smaller than for the others, e.g., *breast cancer, bupa* or *pima*. We wanted to explore the characteristic of these sets by visualizing the distributions of the minority and majority classes in the two-dimensional space. Since all data sets included more than two attributes, we used the PCA method to identify two most important principal components for visualization. We are aware that such analysis may have yielded approximate results (for some data sets more than two components may have been important), nevertheless it led to interesting observations that are reported below.

On the one hand, the minority and majority classes in *acl* and *new thyroid* were easily separable (see Fig. 3 for *new thyroid*), thus even high imbalance ratio was not a serious problem and oversampling methods (RO, CO) were comparable to focused re-sampling (SP2, NCR). On the other hand, the distributions of classes in data sets where NCR or SP2 outperformed RO and CO, e.g., *haberman*, *bupa* or *pima*, were definitely more complicated (see Fig. 4 for *haberman*). Examples from the minority and majority classes were shuffled, there was no clear class boundary, the overlapping area was very large and there were many noisy

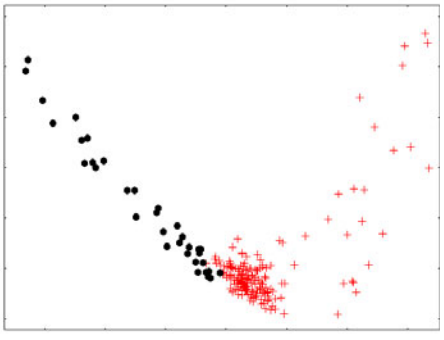


Fig. 3. New thyroid data set

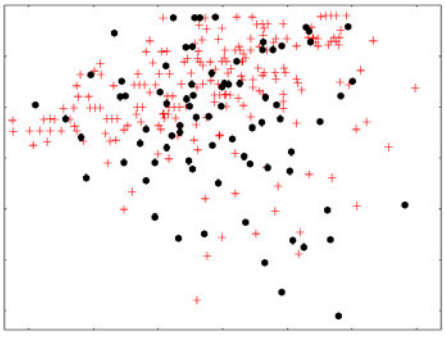


Fig. 4. Haberman data set

examples. This may explain the superior performance of focused re-sampling, as both employed methods (NCR and SPIDER2) were meant to deal with such conditions.

5 Conclusions and Final Remarks

We have presented an experimental study on the impact of critical factors on re-sampling methods dealing with imbalanced data. The first series of experiments show that the degradation in performance of a classifier is strongly affected by the number of borderline examples. If the overlapping area is large enough (in comparison to the area of the minority sub-clusters), and at least 30% of examples from the minority class are located in this area (i.e., are borderline examples), then focused re-sampling methods (NCR, SPIDER2) strongly outperform random and cluster oversampling with respect to sensitivity and G-mean. Moreover, the performance gain increases with the number of borderline examples. On the contrary, if the number of borderline examples is small, then oversampling methods sufficiently improve the recognition of the minority class.

The second series of experiments reveals the superiority of SPIDER2 and in most cases NCR in handling noisy examples located inside the majority class (also accompanied with borderline ones). Such result has been in a way expected, as both methods were introduced to handle such situations. The experiments also demonstrate that oversampling is comparable to SPIDER2 and better than NCR in classifying safe examples from the minority class.

The last series of experiments on real-life imbalanced data sets also provides interesting observations on their nature. We think that PCA-based visualizations of the data sets, on which NCR and both SPIDER methods performed best, are similar to visualizations of artificial data sets with multiple noisy examples and large overlapping ares. In the data sets, where all pre-processing methods worked comparatively, the minority and majority classes are easily separable and the number of "disturbances" is very limited. Thus, we can hypothesize

that difficulties with real-life data are associated with distributions and shapes of classes, their decomposition, overlapping and noise, however, this should be investigated closer in future research.

Although other authors [4,2] have already claimed that class imbalance is not a problem in itself, but the degradation of classification performance is related to other factors related to data distributions (e.g., small disjuncts), we hope that our experimental results expand the body of knowledge on the critical role of borderline and noisy examples.

References

1. Chawla, N.: Data mining for imbalanced datasets: An overview. In: Maimon, O., Rokach, L. (eds.) The Data Mining and Knowledge Discovery Handbook, pp. 853–867. Springer, Heidelberg (2005)
2. Garcia, V., Sanchez, J., Mollineda, R.: An empirical study of the behavior of classifiers on imbalanced and overlapped data sets. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 397–406. Springer, Heidelberg (2007)
3. He, H., Garcia, E.: Learning from imbalanced data. IEEE Transactions on Data and Knowledge Engineering 21(9), 1263–1284 (2009)
4. Japkowicz, N.: Class imbalance: Are we focusing on the right issue? In: Proc. II Workshop on Learning from Imbalanced Data Sets, ICML, pp. 17–23 (2003)
5. Jo, T., Japkowicz, N.: Class Imbalances versus small disjuncts. SIGKDD Explorations 6(1), 40–49 (2004)
6. Kubat, M., Matwin, S.: Addresing the curse of imbalanced training sets: one-side selection. In: Proc. of the 14th Int. Conf. on Machine Learning, pp. 179–186 (1997)
7. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. Tech. Report A-2001-2, University of Tampere (2001)
8. Stefanowski, J., Wilk, S.: Improving rule based classifiers induced by MODLEM by selective pre-processing of imbalanced data. In: Proc. of the RSKD Workshop at ECML/PKDD, pp. 54–65 (2007)
9. Stefanowski, J., Wilk, S.: Selective pre-processing of imbalanced data for improving classification performance. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2008. LNCS, vol. 5182, pp. 283–292. Springer, Heidelberg (2008)

Tracking Recurrent Concepts Using Context

João Bártolo Gomes[1,⋆], Ernestina Menasalvas[1,⋆⋆], and Pedro A.C. Sousa[2]

[1] Facultad de Informatica - Universidad Politecnica Madrid, Spain
joao.bartolo.gomes@alumnos.upm.es,
emenasalvas@fi.upm.es
[2] Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Portugal
pas@fct.unl.pt

Abstract. The problem of recurring concepts in data stream classification is a special case of concept drift where concepts may reappear. Although several methods have been proposed that are able to learn in the presence of concept drift, few consider concept recurrence and integration of context. In this work, we extend existing drift detection methods to deal with this problem by exploiting context information associated with learned decision models in situations where concepts reappear. The preliminary experimental results demonstrate the effectiveness of the proposed approach for data stream classification problems with recurring concepts.

Keywords: Data Stream Mining, Concept Drift, Recurring Concepts, Context-awareness, Ubiquitous Knowledge Discovery.

1 Introduction and Motivation

Learning from data streams where the data distributions and target concepts change over time is a challenging problem [9] in stream mining. In real world stream classification problems, however, it is common for previously seen concepts to reappear [11]. This represents a particular case of concept drift [9], known as recurrent concepts [11,5,1,12].

A weather prediction model usually changes according to the seasons. The same happens with product recommendations where customer interests change over time due to fashion, economy or other *hidden context* [11,3]. Several methods have been proposed to detect and adapt to concept drift [2], but even though concept recurrence is very common in real world problems, these methods do not consider recurrence. The usual approach is to use a forgetting mechanism and learn a new decision model when drift is detected. One possible solution to exploit recurrence is to store learned models that represent previously seen concepts, thus avoiding the need to relearn a concept when it reappears [1,12].

⋆ The work of J.P. Bártolo Gomes is supported by a Phd Grant of the Portuguese Foundation for Science and Technology (FCT).

⋆⋆ The research is partilally financed by project TIN2008-05924 of Spanish Ministry of Education.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 168–177, 2010.

The method presented in this paper extends existing drift detection methods by associating context information with learned decision models to improve the adaptation of the learning process to drift, under the assumption that recurring concepts are related with context. This method is part of a data stream classification process able to handle concept drift with recurrence. The main contribution lies in exploiting context information associated with decision models in the mechanism to store and retrieve past models when drift is observed. This approach enhances the adaptation to drift when concept recurrence is present in the data stream and the underlying concepts are related with observable context information.

The proposed method is integrated in a data stream learning process framework that is divided into two levels: a base learner level, where the model representing the current concept is learned; and a meta-learning level where drift is detected. When drift occurs the meta-learning level is responsible for: a) store the current model and its associated context; b) from previously stored models select the most adequate for the current data and observable context.

This paper is organized as follows. Section 2 describes the related work. The proposed solution is found in section 3, with its assumptions and requirements, as well as a detailed description of its components. In section 4 the experimental results obtained from our prototype are presented and discussed. Finally we provide some concluding remarks and outline future research work in section 5.

2 Related Work

A review of literature related to the problem of concept drift can be found in [9], where the problem is defined and several approaches to handle it are discussed. However, most of these approaches discard old information and the possibility for concept recurrence is not even considered. We focus our review on works that address recurrence of concepts. These works approach concept drift in one of the following ways:

- using a window of records or a decay function, old records are forgotten and the decision model is updated from the most recent data as it appears in the window. Some approaches are able to handle recurrence by storing models learned from fixed chunks of records and building an ensemble classifier with them [5,7,10]. When the concept changes, the weights of the models are adjusted to maximize the performance of the ensemble according to the current concept.
- using a drift detection method [2], which signals when drift occurs. When it does, the old model is discarded and a new one begins to be learned so it can represent the new concept. More sophisticated approaches that exploit concept recurrence [1,12] employ some strategy to store learned models.

Both approaches address the issue of concept drift by ensuring that the classification model used at time α represents the concept observed at that time, thus obtaining high classification accuracy for the underlying concept in the stream.

Related with recurrent concepts in[7], an ensemble method is proposed where in the case that no classifier in the ensemble performs better than the error threshold, a classifier for the new concept is learned and stored in a global set. The classifiers with better performance on the current data are part of the ensemble for labeling new examples. Similarly in[5] an ensemble is used but incremental clustering was exploited to maintain information about historical concepts. The proposed framework captures batches of examples from the stream into conceptual vectors. Conceptual vectors are clustered incrementally by their distance and for each cluster a new classifier is learned and then becomes part of the ensemble. Another method for reusing historical concepts is proposed in[12]. Its major contribution consists in using a proactive approach, which means to reuse a concept from the history of concepts that is treated like a Markov chain, to select the most probable concept according to a given transition matrix. In [1] the focus lies in the selection of previously learned models. Referees are used to choose the most appropriate model from history. Whenever drift is detected the classifier and its referee are stored in a pool for further use. The model is re-used when the percentage of votes of its referee exceeds some given threshold, otherwise a new classifier is learned.

In [3] an off-line, meta-learning approach to deal with concept drift using context is presented. The approach uses an existing batch learner and a process called contextual clustering to identify stable hidden contexts and the associated context specific stable concepts. The approach is applicable to the extraction of context reflected in time and spatial attributes.

3 Proposed Learning Process

This work proposes a mechanism that associates context information to learned decision models in order to improve the learning process on a data stream mining classification scenario. The requirements of this learning process are:

- **i)** adapt the current learning model to concept drift[9].
- **ii)** recognize and use past models from previously seen concepts when they reappear[9].
- **iii)** use contextual information to improve adaptation to drift[3].

The proposed mechanism stores the learned models and context information together. This information is later used when a concept reappears. We integrate the proposed mechanism in a two-level framework with: a) base learner level where an incremental algorithm learns a concept, in the form of a classification model; b) meta-learning level where detection and adaptation to concept drift is performed and the context-aware mechanism is used. We assume that context information is available and that the target concepts are related to context.

The most similar approaches with the one proposed in this work are the ones presented in [12,1]. Both use drift detection and store past models in order to adapt to concept drift and recurrence. However, as reviewed in section 2 the difference lies in the mechanism used to store and select past models. Our context-aware approach of exploiting the context resembles and shares the motivation

with the one presented in [3] where the method infers the periods where *hidden context* is stable from available context features, which are described as contextual clusters. The main differences are that we present an on-line method which uses context spaces [6] to model context and we don't require the partition of the dataset into batches because our concepts are of arbitrary size as determined by the drift detection method. Instead we use the relation between stable concepts and context to improve the adaptation to drift when these concepts recur.

3.1 Challenges

The main challenges of the proposed approach are related to decision model storage and retrieval. In this work, the models are stored in a model repository with a reference to the associated context as described in subsection 3.2. This represents the context observed in the records used to learn that model, as illustrated in figure 3.3. For retrieval, the repository is searched for a model that represents the current data and with context similar to the currently occurring one. The exact metrics used for this comparison are described in subsection 3.4.

3.2 Context Management

We base our context modeling on the Context Spaces model [6]. The Context Spaces model represents contextual information in a multidimensional Euclidean space. A context state C is defined as a tuple of N attribute-values, C = (a_1^v, ..., a_n^v), where a_n^v represents the value of attribute a_n. A context space defines the regions of acceptable values for these attributes. An occurring context is defined as a sub-space in this multidimensional space, most often as a point in this space. In this work we use numerical context attributes and the similarity between contexts is measured by the Euclidean distance.

When setting up a particular application one must define the context features and the similarity thresholds used to compare contexts. For example, temporal and spatial features in ubiquitous applications. In different scenarios other similarity measures can be explored.

3.3 Learning Process Framework

Figure 3.3 represents the learning process framework that integrates the proposed mechanism. In what follows we describe its components.

Base Learner. In the base learning task, we used Naive Bayes algorithm. We choose Naive Bayes because it is a well known incremental classifier algorithm which can deal with different types of attributes and is easy to implement as it only requires updating its probability tables. This is also an advantage when storing the model for later use. We should note that any incremental classifier can be used in this task.

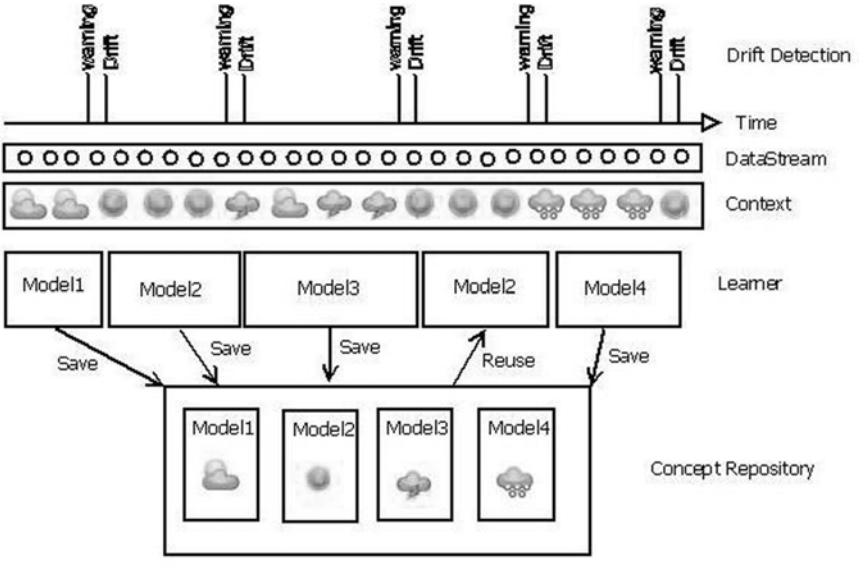


Fig. 1. Data Stream Learning Process Framework

Meta Learning. In the meta-learning level a drift detection mechanism is used, which monitors the performance of the learning algorithm and triggers an event when drift occurs. The proposed context-aware mechanism to handle concept recurrence is part of the meta-learning level.

Drift Detection method. When the records distribution is stationary the classifier error-rate decreases as the number of training records grows. This assumption is shared by most approaches dealing with drift [9], as it is a natural property in real world problems where periods of stable concepts are observed followed by changes leading to a new period of stability with a different target concept. Proposed in [2] is a method for drift detection which uses this assumption to find drift events.

The Drift Detection Method [2], stores the probability of misclassifying p_i = (F /i) and the standard deviation $s_i = \sqrt{pi(1-pi)/i}$, where i is the number of trials and F is the number of false predictions. These values are updated incrementally. Two levels are defined, a warning level and a drift level, which are reached according to some condition based on p_i and s_i and constants that define the minimum values of p_i and s_i. Note that it is possible to observe an increase in the error-rate reaching the warning level, followed by a decrease, which can be considered to correspond to a false alarm.

Context Manager. In accordance with the description in section 3.2, we assume all the context related pre-processing is performed by this component.

Model Repository. The model repository is the structure that stores learned models and associated context description.

3.4 Learning Algorithm

The proposed learning process with drift detection that integrates our mechanism is outlined below:

1. process the incoming records from the data stream using an incremental learning algorithm (base learner) to obtain a decision model capable of representing the underlying concept.
2. the drift detection method[2] monitors the error-rate of the learning algorithm.
3. when the error-rate goes up the drift detection method signals one of the following events:
   - **warning level** - store the incoming records into a warning window and prepare a new learner that processes incoming records while the warning level is signaled.
   - **drift level** - store the current model and its associated context into the model repository; use the model repository to find a past model with a context similar to the current one that performs well with the new data records (i.e. represents the current concept). Reuse the model from the repository as a base learner to continue the learning process as in point 1. If no model is found use the learner that was initiated during warning level as base learner.
   - **false alarm (normal level after warning)** - the warning window is cleared and the learner used during the warning period is discarded. The learning process continues as in point 1, which is also the normal level in terms of drift detection.

Note that if the warning level is maintained for a number of records greater than a specified threshold and context change is observed (i.e. a significant distance between the context in the new learner and the past one), the drift level case is executed.

The algorithm depicted in 1 describes the pseudo-code in which our mechanism is integrated with the learning process.

The function getModel is the one in charge of model retrieval. Its main objective is to find the model that better represents the underlying concept and has context similar to the current context. This implies that the stored model achieves high accuracy with this underlying concept. Our approach searches the model repository and calculates the error of the past models with the current data, using the records available in the WarningWindow. If the number of records is lower than a specified threshold, the newLearner is returned (and will be used as baseLearner). Since we are unable to estimate the performance of past models, the same also happens when no model is found (i.e. the repository is empty or the performance of the stored models is too low).

3.5 Mechanism Metrics

The error prediction of the past models is calculated by the mean square error (as proposed in [10] to obtain the weight of the classifier in an ensemble approach)

Algorithm 1. Data Stream Learning Process

Require: Data stream DS, ModelRepository MR

```
repeat
    Get next record DS_i from DS;
    prediction = baseLearner.classify(DS_i);
    DriftDetection.update(prediction);
    switch DriftDetection.level
    case Normal
        baseLearner.train(DS_i);
    case Warning
        WarningWindow.add(DS_i);
        newLearner.train(DS_i);
    case FalseAlarm
        WarningWindow.clear();
        newLearner.delete();
    case Drift
        MR.store(baseLearner, baseLearner.context);
        baseLearner=MR.getModel(newLearner,newLearner.context,WarningWindow);
    end switch
until END OF STREAM
```

of classifier C_i, using the WarningWindow W_n of n records in the form of (x, c), where c is the true class label for that record. The error of C_i on record (x, c) is 1 - $f_c^i(x)$, where $f_c^i(x)$ is the probability given by C_i that x is an instance of class c. For cost-sensitive applications a benefit matrix can be used as is proposed in [10]. We only use as candidate models C_i that have an error below a certain threshold. The MSI_i can be expressed by:

$$MSE_i = \frac{1}{|W_n|} \sum_{(x,c) \in W_n} (1 - f_c^i(x))^2$$

One important aspect of the approach is to minimize the similarity between the occurring context and the context observed in stored models, as this indicates that we are observing a similar context. This is achieved by means of the function $Dis(Context_i, OContext)$, where $Context_i$ is the context associated with C_i and $OContext$ is the current context. The distance used is the Euclidean distance, as discussed in section 3.2. Thus the selection criterion resulting from the two metrics is the utility function:

$$u(MSE_i, Dis(Context_i, OContext))$$

To maximize this utility value, both metrics should be minimal. Weights are assigned to the context and learner components of the utility function. The model with the highest utility value is selected, and the weight between the model and context is increased in the repository, signaling that it was used successfully as a recurrent model for that context. Although resource-awareness is not the focus of this work, this strategy can be exploited to delete past models that are not

reused as a way to decrease the memory consumption of the approach. If no model in the repository reaches the utility threshold value, the newLearner is returned instead of a past concept.

4 Experimental Results

The implementation of the proposed learning process was developed in Java, using the MOA[4] environment as a test-bed. The evaluation features and the *SingleClassifierDrift* class that implements the drift detection method of[2], provided a starting point to implement the specific components of our approach. The experiments were run on an Intel Core 2 Duo 2.10 GHz CPU with 2 GB of RAM.

As dataset the SEA Concepts[8] were used with MOA[4] as stream generator. SEA Concepts is a *benchmark* data stream that uses different functions to simulate concept drift, allowing control over the target concepts and its recurrence in our experiment. The experiment used 250000 records and changed the underlying concept every 15000 records. The test was repeated with a 10% noise value, which means that the class value is wrong in 10% of the records, testing how sensitive is the approach to noise in data. The context feature season {winter, spring, summer, autumn} was generated independently as a context stream where the season is associated with the target concept function with probability 0.9. For example, if target concept is function 1, context variable season will be winter with probability 0.9 with the remaining values belonging to one of the other possible seasons as noise. We assigned 100 records as the threshold of the warningWindow. The weights 0.8 and 0.2 were assigned to the classifier accuracy and context distance, respectively as values in the utility function described in section 3.

4.1 Results

We compared the approach proposed in this paper with the *SingleClassifierDrift* implemented in MOA[4] in terms of accuracy. *SingleClassifierDrift* detects drift using the drift detection method[2]. When drift occurs, it learns a new model and completely forgets the old one. As we can see in Figure 2 our approach leads to better results than *SingleClassifierDrift* when recovering from concept drift, in both experiments. In general, our approach adapted to drift faster and the models selected by the context-aware mechanism were able to represent the target concepts as can be seen by the accuracy obtained. The *SingleClassifierDrift* approach always has to relearn the underlying concept from scratch after drift is detected. However, in some situations for example at record 19500 in the dataset with 10% noise, the selected model seems to represent the target concept at first but the *SingleClassifierDrift* approach is able to achieve better results. In this case the fast adaptation of our greedy approach led to the selection of a weaker model. A more conservative approach could be used by increasing the number of records in the warningWindow. It is also noticeable that the proposed approach

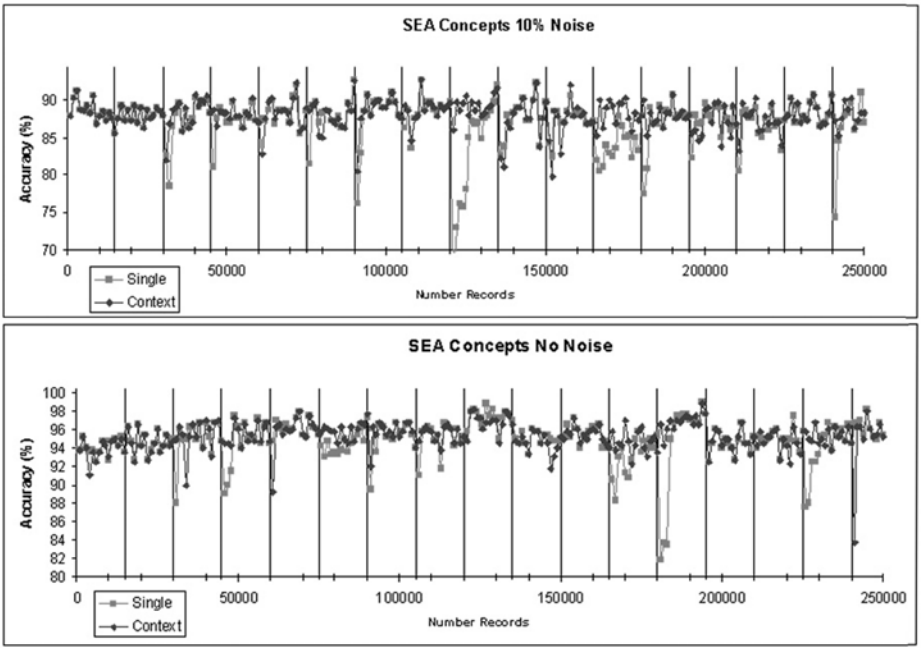


Fig. 2. Comparison of accuracy with Proposed approach(Context) vs SingleClassifier-Drift(Single) using the SEA concepts dataset. Black vertical lines show when drift occurs.

achieves a more stable accuracy over time, because it recovers much faster from drift than the approach without stored models. This is more significant when using the dataset with 10% noise, where the proposed approach obtained 2046 more correct predictions compared to 802 in the dataset without noise. The usage of context as part of our mechanism enables us to exploit the associations between recurrent concepts and context as a way to track concept recurrence and achieve better results in situations where this association exists.

5 Conclusions and Future Work

In this work, we have proposed a method for the problem of data stream classification with concept recurrence that associates context information with learned models, improving adaptation to drift. The main contribution of this work lies in the mechanism that associates context with stored models to track recurrence when drift occurs. A description of the requirements, assumptions, and components of the solution is presented.

We also present preliminary results with the artificial benchmark data set SEA Concepts. The results are promising for situations where concepts reoccur periodically and are associated with context.

As future work we plan to further explore the proposed method, using different storage and retrieval mechanisms for past models, a resource-aware approach for applications where storage costs are severely limited, methods to predict the context information associated with the data stream records, as well as testing the current approach in real world problems. Furthermore, the preliminary experimental results seem to indicate that reusing previously stored models can lead to gains proportional to the effort required to learn a model from scratch. It would be interesting to study such gains on these harder learning problems.

References

1. Gama, J., Kosina, P.: Tracking Recurring Concepts with Meta-learners. In: Lopes, L.S., Lau, N., Mariano, P., Rocha, L.M. (eds.) EPIA 2009. LNCS, vol. 5816, pp. 423–434. Springer, Heidelberg (2009)
2. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS (LNAI), vol. 3171, pp. 286–295. Springer, Heidelberg (2004)
3. Harries, M.B., Sammut, C., Horn, K.: Extracting hidden context. Machine Learning 32(2), 101–126 (1998)
4. Holmes, G., Kirkby, R., Pfahringer, B.: MOA: Massive Online Analysis (2007), http://sourceforge.net/projects/moa-datastream/
5. Katakis, I., Tsoumakas, G., Vlahavas, I.: Tracking recurring contexts using ensemble classifiers: an application to email filtering. In: Knowledge and Information Systems, pp. 1–21
6. Padovitz, A., Loke, S.W., Zaslavsky, A.: Towards a theory of context spaces. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, pp. 38–42 (2004)
7. Ramamurthy, S., Bhatnagar, R.: Tracking recurrent concept drift in streaming data using ensemble classifiers. In: Proc. of the Sixth International Conference on Machine Learning and Applications, pp. 404–409 (2007)
8. Street, W.N., Kim, Y.S.: A streaming ensemble algorithm (SEA) for large-scale classification. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 377–382. ACM, New York (2001)
9. Tsymbal, A.: The problem of concept drift: definitions and related work. Computer Science Department, Trinity College Dublin (2004)
10. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 226–235. ACM, New York (2003)
11. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine learning 23(1), 69–101 (1996)
12. Yang, Y., Wu, X., Zhu, X.: Combining proactive and reactive predictions for data streams. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, p. 715. ACM, New York (2005)

Support Feature Machine for DNA Microarray Data

Tomasz Maszczyk and Włodzisław Duch

Department of Informatics, Nicolaus Copernicus University
Grudziądzka 5, 87-100 Toruń, Poland
{tmaszczyk,wduch}@is.umk.pl
http://www.is.umk.pl

Abstract. Support Feature Machines (SFM) define useful features derived from similarity to support vectors (kernel transformations), global projections (linear or perceptron-style) and localized projections. Explicit construction of extended feature spaces enables control over selection of features, complexity control and allows final analysis by any classification method. Additionally projections of high-dimensional data may be used to estimate and display confidence of predictions. This approach has been applied to the DNA microarray data.

Keywords: Features generation, dimensionality reduction, learning, support feature machines, support vector machines.

1 Introduction

Data mining packages such as Weka [1], Rapid Miner [2], or KNIME [3] offer enormous number of modules for pre-processing, optimization, training, classification, clustering, visualization and post-processing. For example, combining all modules available in Rapid Miner 3.4 over 5 billion data models may be created. Comparison of all these models on a single dataset would be impossible and will anyway manifest results of "oversearching", with some models producing good results on a given dataset by pure chance [4]: contrary to the common opinion generalization of learning systems (including decision trees and neural networks) trained using global optimization methods (such as evolutionary algorithms) is frequently worse than results obtained from the greedy, best-first methods (such as the gradient methods). Meta-learning, or creating on demand optimal adaptive system suitable for a given problem is an answer to this crises of abundance [5,6]. Different approaches to meta-learning have been proposed [7], based either on recommendations of particular learning methods depending on some data characteristics (landmarking approach), combining many data models together [8], or defining frameworks that allow for systematic creation of data models of growing complexity. In particular, a framework based on evaluation of similarity [5], that includes many variants of the nearest neighbor methods, neural networks and kernel methods [9], has proved to be very powerful in practical applications. Within such framework methods that have a proper bias for particular problem may be found.

Each data model M_i is defined in some hypotheses space $\mathcal{H}$ that includes all functions that this model may learn. Combination of diverse classifiers $p(C|\boldsymbol{x}, M_i)$ that predict probabilities or provide binary indicators for each class C, may improve results. Mixture of experts [8], and in particular the boosted mixtures of experts [10],

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 178–186, 2010.

assign large weights to classifiers (called experts) that improve performance around localized areas of the input space where most models fail. Committees of competent models [11,12] have weights $\boldsymbol{w}_i(\boldsymbol{x})p(C|\boldsymbol{x}, M_i)$ that explicitly depend on the input vectors $\boldsymbol{x}$, decreasing the influence of classifiers that have not been competent, or had low confidence in their predictions in some regions of the input space. Stacking classifiers is another technique that trains to predict errors that the current set of classifiers makes [13].

Committees and mixture of experts are based on component models optimized for the whole training data. All these methods decrease comprehensibility of solutions and are not the most efficient way of summarizing the data. Recently we became interested in ensembles of simple models at finer granulation, defining classifiers based on single feature and provide data models that are competent only for relatively small subsets of all samples. The most common Gaussian kernel Support Vector Machine (SVM) [9] selects a subset of training vectors $\boldsymbol{x}_i$ close to the decision border (called "support vectors") and calculates $k(\boldsymbol{x}, \boldsymbol{x}_i) = \exp(-\beta \sum |\boldsymbol{x}_i - \boldsymbol{x}|^2)$, with fixed dispersion β. The final discriminant function is constructed as a linear combination of such kernels, creating in fact a weighted nearest neighbor solution. Other similarity estimation based on specific kernels (similarity functions) may help to increase flexibility of decision borders, decreasing the number of kernels need for accurate classification.

Each support vector used in a kernel may provide a useful feature, but this type of solution is optimal only for data with particular distributions, and will not work well for example on parity data [14] or other problems with complex logical structure [15]. For some highly-non-separable problems localized linear projections may easily solve the problem [16]. Adding new types of features extends the hypothesis space. Useful features may be created by random linear projections [17], or principal components derived from data, or projection pursuit algorithms based on Quality of Projected Clusters (QPC) indices [18]. Non-linear combinations of input features provided by neural network nodes may also be used.

Creation of appropriate feature space facilitates finding optimal solutions, and thus is worthwhile exploring not only at the level of combination of classifiers, but in the first place to learn from other models what interesting features they have discovered. They may come in form of prototypes, linear combinations, or fragments of branches in decision trees, forming useful “knowledge granules” in data, as it is done in our Universal Learning Machines [19]. The final model – linear discrimination, Naive Bayes, nearest neighbor or a decision tree – is secondary, if appropriate space has been set up.

In the next section SFM algorithm as used here is introduced, and in section 3 tested on several microarray data. Brief discussion of further research directions concludes this paper.

2 Support Feature Machine Algorithm

Support Vector Machines [20,9] used as classifiers are based on linear discrimination, with maximization of classification margin that ensures good generalization and allows for control of complexity. Problems that require non-linear decision borders may be linearized by projecting the data into high-dimensional feature space. According to the

Cover theorem [21] such mapping increases the probability of making the data separable, "flattening" decision borders. Kernel methods implicitly provide new features $z_i(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}_i)$ constructed around support vectors $\boldsymbol{x}_i$, selected from the training close to the decision borders. The number of these features is equal to the number of training vectors n. The ability to solve classification problem using only the kernel matrix $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ has some advantages, as instead of the original input space $\boldsymbol{x}_i$ one works in the space of kernel features $z_i(\boldsymbol{x})$,called further “the kernel space”.

The number of input features N may be smaller or greater than n. If $N \ll n$ the number of the kernel features is usually unnecessarily large – think of the simple case of two Gaussian distributions that are optimally solved by forming a projection connecting their means, instead of a large number of support vectors with Gaussian kernels close to the decision border. Adding such linear projections to the list of features, and performing feature selection may in such situation remove most kernel features as irrelevant. In the reverse situation, when $n \ll N$ (as is the case for the microarray data), instead of a projection into high-dimensional space SVM reduces the number of features to no more than n. Also in this case other types of features may be useful. In particular original input features may be kept in the enhanced feature space (SVM does not use them), or various projections may be added. This should guarantee that simplest solutions to easy problems are not overlooked (SVM will miss them, even if a single binary feature is sufficient).

Support Feature Machines used here generalize SVM approach by explicitly building enhanced space that includes kernel features $z_i(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}_i)$ together with any other features that may provide useful information. This approach has several advantages comparing to standard SVM:

1. With explicit representation of features interpretation of discriminant function is as simple as in any linear discrimination method.
2. Kernel-based SVM is equivalent to linear SVM in the explicitly constructed kernel space, therefore enhancing this space should lead to improvement of results.
3. Kernels with various parameters may be used, including various degrees of localization, and the resulting discriminant may select global features, combining them with local features that handle exceptions (SVM algorithms are usually formulated using single kernel, with a few exceptions [22]).
4. Complexity of SVM is $O(n^2)$ due to the need of generating kernel matrix; SFM may select smaller number of kernel features at $O(n)$ cost from those vectors that project on overlapping regions in linear projections.
5. Many feature selection methods may be used to estimate usefulness of new features that define support feature space.
6. Many algorithms may be used in the support feature space to generate the final solution.

Although one can use various methods to solve classification problem in the space created by support features, linear SVM approach is used here for several reasons. First, there are many excellent and well-tested SVM implementations available. Second, linear SVM includes regularization term, ensuring large margin solution. For microarray

data other linear techniques designed for small sample problems may be more appropriate [23] and will be tried in future. We shall consider other machine learning algorithms in future SFM versions.

In this paper only basic version of this approach is implemented (see Algorithm 1) using linear discrimination to obtain final decision function. SFM algorithm starts from standardization of the whole data, followed by feature selection, based here on the Relief algorithm [24], leaving only features with positive weights. Such reduced, but still high dimensional data, is used to generate two types of new features.

First type of features is made by projections on $m = N_c(N_c - 1)/2$ directions obtained by connecting pairs of centers $\boldsymbol{w}_{ij} = \boldsymbol{c}_i - \boldsymbol{c}_j$, where $\boldsymbol{c}_i$ is the mean of all vectors that belong to the $C_i, i = 1...N_c$ class. In high dimensional space such features $r_i(\boldsymbol{x}) = \boldsymbol{w}_i \cdot \boldsymbol{x}$ help a lot, as can be seen in Fig. 1, where smoothed histograms of data projected on such directions are shown. Fisher discriminant analysis may provide even better directions here, but this is the least expensive solution, always worth adding to the pool of new features.

The second type are features based on similarity to the training vectors, or kernel features. While many types of kernels may be mixed together, including the same types of kernels with different parameters, in the initial implementation only Gaussian kernels with fixed dispersion β are taken for each training vector $t_i(\boldsymbol{x}) = \exp(-\beta \sum |\boldsymbol{x}_i - \boldsymbol{x}|^2)$. Various ways of selecting suitable training vectors may be considered, but for small sample data all of them may be taken into account. QPC algorithm [18] has been used on this feature space, generating additional orthogonal directions that are useful for visualization and as new features. The number of QPC features has been arbitrarily set to 5, although optimizing this parameter in crossvalidation should give better final result.

Algorithm 1. Algorithm

Require: Fix the Gaussian dispersion β and the number of QPC features N_Q.

1: Standardize dataset.
2: Normalize the length of each vector to 1.
3: Perform Relief feature ranking, select only those with positive weights $RW_i > 0$.
4: Calculate class centers $\boldsymbol{c}_i, i = 1...N_c$, create m directions $\boldsymbol{w}_{ij} = \boldsymbol{c}_i - vc_j, i > j$.
5: Project all vectors on these directions $r_{ij}(\boldsymbol{x}) = \boldsymbol{w}_{ij} \cdot \boldsymbol{x}$ (features r_{ij}).
6: Create kernel features $t_i(\boldsymbol{x}) = \exp(-\beta \sum |\boldsymbol{x}_i - \boldsymbol{x}|^2)$.
7: Create N_Q QPC directions $\boldsymbol{w}_i$ in the kernel space, adding QPC features $s_i(\boldsymbol{x}) = \boldsymbol{w}_i \cdot \boldsymbol{x}$.
8: Build linear model on the new feature space.
9: Classify test data mapped into the new feature space.

In essence SFM algorithm requires construction of new features, followed by a simple linear model (linear SVM has been used here) or any other learning model. More attention is paid to generation of features than to the sophisticated optimization algorithms or new classification methods. Although several parameters may be used to control the process of feature creation and selection they are either fixed or set in an automatic way. Solutions are given in form of linear discriminant function and thus are easy to understand. New features created in this way are based on those transformations

of inputs that have been found interesting for some task, and thus have meaningful interpretation (see Fig. 1, 2. The importance of generating new features has already been stressed in our earlier papers, but they have been based either on random projections [17], extracted from several types of algorithms such as decision trees or the nearest-neighbor methods [19], or provided by classification algorithms [25]. Systematic creation of support feature spaces as described in this paper seems to be an unexplored approach with great potential.

3 Illustrative Examples

The usefulness of SFM approach has been tested on several DNA microarray datasets and compared with SVM and SSV[26] results. These high-dimensional datasets have been used in the Discovery Challenge at the 7-th International Conference on Rough Sets and Current Trends in Computing (RSCTC 2010), Warsaw, Poland (28-30 June, 2010). A summary of these datasets is presented in Tab. 1.

Table 1. Summary of used datasets

Title	#Features	#Samples	Class distribution
Data 1	54675	123	88–35
Data 2	22283	105	40–58–7
Data 3	22277	95	23–5–27–20–20
Data 4	54675	113	11–31–51–10–10
Data 5	54613	89	16–10–43–20
Data 6	59004	92	11–7–14–53–7

Because the number of samples in some classes is very small, in the k-fold cross-validation tests performed for evaluation of SFM performance, k is equal at least to the number of vectors in the smallest class (from 5-10), and at most 10. Average results are collected in Table 2, with accuracy, balanced accuracy (accuracy for each class) and standard deviation given for each dataset. Additionally values of parameters determined in internal crossvalidation are given. Number of features after the relief selection is still very large and it is clear that more conservative selection could be used; these features are used only to generate a small number of r, t and s-type of features. Number of new features NF includes 5 QPC and $N_c(N_c - 1)/2$ directions connecting centers, so together with the kernel feature the support feature space has at most 129 dimensions.

Optimal dispersion of kernel features is in most cases quite large, up to 2^8, creating very smooth, partially almost linear decision border in the kernel space. These results show by no means the limits of SFM algorithm, as fixed parameters have been used in all tests, more features could be generated using random projections, alternative kernels, more QPC directions, and other linear discrimination algorithms or other machine learning algorithms should be tested to generate final decision functions. Unfortunately exploration of all these possibilities is quite time consuming and software to perform all necessary comparisons in an automatic way is under development.

Table 2. Accuracies (ACC) and balanced accuracies (BACC) for each used dataset. Also for SFM optimal parameter β is noted, number of features after relief selection (FAS), number of new features (NF), and number of folds used in crossvalidation tests.

Dataset	Info			SVM		SSV		SFM		
	FAS	NF	CV	ACC	BACC	ACC	BACC	ACC	BACC	β
Data 1	51259	117	10	93.5±6.3	90.5±9.7	74.0±13.0	71.0±17.0	91.5±9.3	89.9±11.5	2^3
Data 2	14880	98	7	60.9±10.5	54.2±18.3	56.2±8.4	35.4±6.3	67.8±15.5	67.1±14.3	2^8
Data 3	15628	91	5	75.8±7.9	65.9±6.1	74.7±4.4	66.6±4.3	96.0±5.4	89.3±8.9	2^3
Data 4	7529	116	10	38.1±9.9	28.9±8.9	38.8±12.2	19.8±8.1	54.1±15.8	41.5±13.1	2^{-4}
Data 5	31472	91	10	60.4±18.4	55.1±21.7	59.4±12.3	49.0±21.3	68.6±7.9	64.9±9.6	2^3
Data 6	47307	88	7	61.9±8.5	46.4±13.4	57.5±2.6	24.1±8.2	79.4±17.2	53.1±15.5	2^0

The microarray data in particular require small sample methods with appropriate smoothing. This is not the best domain to evaluate learning algorithms, as drawing conclusions from balanced accuracy on data with a few such samples per class in a space of such huge dimension without any domain knowledge is impossible. However, this is quite challenging data and therefore a comparison of SFM with SVM results is interesting.

4 Discussion and New Directions

Support Feature Machine algorithm introduced in this paper is focused on generation of new features, rather than improvement of optimization and classification algorithms. It may be regarded as an example of mixture of experts, where each expert is a simple model based on projection on some specific direction (random, or connecting clusters), localization of projected clusters (QPC), optimized directions (for example by Fisher discriminant analysis), or kernel methods based on similarity to reference vectors. Obviously there is a lot of room for improvement. For some data kernel-based features are most important, for other projections and restricted projections discover more interesting aspects. Recently more sophisticated ways of creating new features have also been introduced [19,25], deriving new features from various data models. For example, combination of several features with appropriate thresholds obtained from decision trees, create interesting semi-local features.

Instead of simple linear combination competent committee may be introduced. Calculating probability $p(C_i|\boldsymbol{x};M)$ of assigning $\boldsymbol{x}$ vector to class C_i by the final model M, given probabilities for each feature $P(C_i|\boldsymbol{x};M_j)$ (as shown in Fig. 1), coefficients of linear combination are determined from the least-mean square solution of:

$$p(C_i|\boldsymbol{x};M) = \sum_{j=1}^{m} \sum_{m} W_{ij} F(\boldsymbol{x}_j) P(C_i|\boldsymbol{x}_j) \tag{1}$$

where the incompetence factors $F(\boldsymbol{x}_j)$ are estimated from histograms $P(C_i|\boldsymbol{x}_j)$. These factors simply modify probabilities $F(\boldsymbol{x};M_j)P(C_i|\boldsymbol{x}_j)$ that are used to set up models in the enhanced feature space. This approach requires only minimal change in the whole algorithm. After renormalization $p(C_i|\boldsymbol{x};M)/\sum_j P(C_j|\boldsymbol{x};M)$ gives final probability

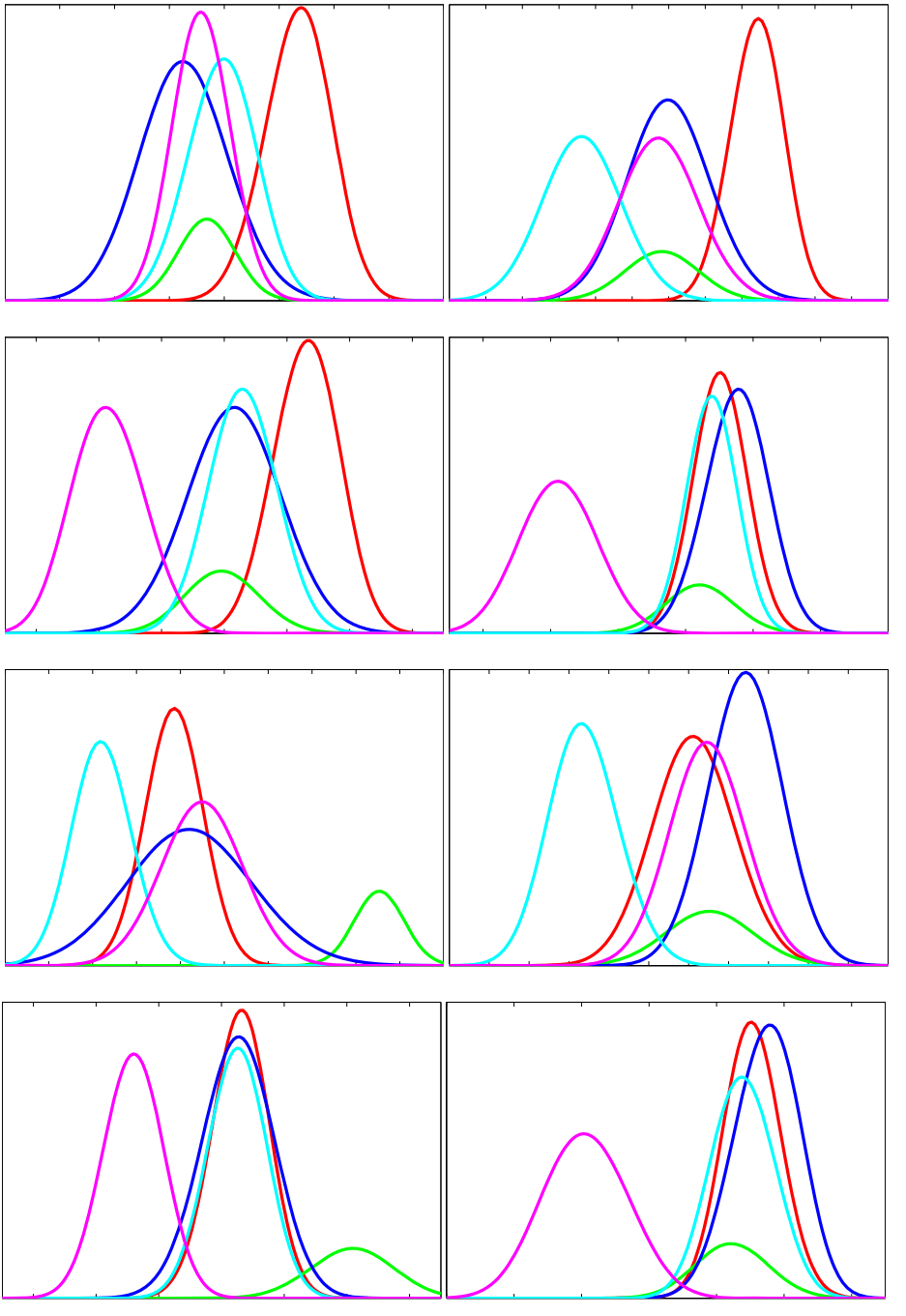


Fig. 1. Histograms for 5 classes (Dataset 3), two QPC projections on lines connecting centers of these classes

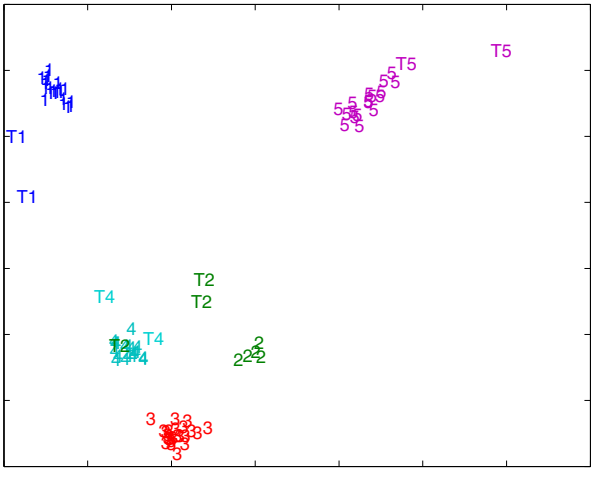


Fig. 2. Scatterogram showing distribution of training and test vectors in QPC directions derived from kernel features, taken from one of the 10-fold crossvalidation folds

of classification. In contrast to AdaBoost and similar procedures [13] explicit information about regions of incompetence, or quality of classifier based on single new feature in different feature space areas, is used. Many other variants of basic SFM algorithm reported here are possible. It is worth noting that kernel-based SVM is equivalent to the use of kernel features combined with linear SVM. Mixing different kernels and different types of features creates much better enhanced features space then a single-kernel solution. For example, complex data may require decision borders of different complexity, and it is rather straightforward to introduce multiresolution in the presented algorithm, for example using different dispersion β for every $\boldsymbol{t}_i$, while in the standard SVM approach this is difficult to achieve.

References

1. Witten, I., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
2. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale: Rapid prototyping for complex data mining tasks. In: Proc. 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, KDD 2006 (2006)
3. Berthold, M., Cebron, N., Dill, F., Gabriel, T., Kötter, T., Meinl, T., Ohl, P., Sieb, C., Thiel, K., Wiswedel, B.: KNIME: The Konstanz Information Miner. In: Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007). Springer, Heidelberg (2007)
4. Quinlan, J., Cameron-Jones, R.: Oversearching and layered search in empirical learning. In: Proc. of the 14th International Joint Conference on Artificial Intelligence, pp. 1019–1024. Morgan Kaufmann, San Francisco (1995)

5. Duch, W., Grudziński, K.: Meta-learning via search combined with parameter optimization. In: Rutkowski, L., Kacprzyk, J. (eds.) Advances in Soft Computing, pp. 13–22. Physica Verlag, Springer, New York (2002)
6. Grąbczewski, K., Jankowski, N.: Meta-learning with machine generators and complexity controlled exploration. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 545–555. Springer, Heidelberg (2008)
7. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Metalearning: Applications to Data Mining. In: Cognitive Technologies. Springer, Heidelberg (2009)
8. Kuncheva, L.: Combining Pattern Classifiers. Methods and Algorithms. J. Wiley & Sons, New York (2004)
9. Schölkopf, B., Smola, A.: Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2001)
10. Avnimelech, R., Intrator, N.: Boosted mixture of experts: An ensemble learning scheme. Neural Computation 11, 483–497 (1999)
11. Duch, W., Itert, L.: Committees of undemocratic competent models. In: Rutkowski, L., Kacprzyk, J. (eds.) Proc. of Int. Conf. on Artificial Neural Networks (ICANN), Istanbul, pp. 33–36 (2003)
12. Duch, W., Itert, L.: Competent undemocratic committees. In: Rutkowski, L., Kacprzyk, J. (eds.) Neural Networks and Soft Computing, pp. 412–417. Physica Verlag/Springer, Heidelberg (2002)
13. Smyth, P., Wolpert, D.: Linearly combining density estimators via stacking. Machine Learning 36, 59–83 (1999)
14. Brown, D.A.: N-bit parity networks. Neural Networks 6, 607–608 (1993)
15. Grochowski, M., Duch, W.: Learning highly non-separable Boolean functions using Constructive Feedforward Neural Network. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) ICANN 2007. LNCS, vol. 4668, pp. 180–189. Springer, Heidelberg (2007)
16. Duch, W.: k-separability. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4131, pp. 188–197. Springer, Heidelberg (2006)
17. Duch, W., Maszczyk, T.: Almost random projection machine. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) ICANN 2009. LNCS, vol. 5768, pp. 789–798. Springer, Heidelberg (2009)
18. Grochowski, M., Duch, W.: Projection Pursuit Constructive Neural Networks Based on Quality of Projected Clusters. In: Kůrková, V., Neruda, R., Koutník, J. (eds.) ICANN 2008, Part II. LNCS, vol. 5164, pp. 754–762. Springer, Heidelberg (2008)
19. Duch, W., Maszczyk, T.: Universal learning machines. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) ICONIP 2009, Part II. LNCS, vol. 5864, pp. 206–215. Springer, Heidelberg (2009)
20. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)
21. Cover, T.M.: Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Transactions on Electronic Computers 14, 326–334 (1965)
22. Sonnenburg, S., Raetsch, G., Schaefer, C., Schoelkopf, B.: Large scale multiple kernel learning. Journal of Machine Learning Research 7, 1531–1565 (2006)
23. Tebbens, J., Schlesinger, P.: Improving implementation of linear discriminant analysis for the small sample size problem. Computational Statistics & Data Analysis 52, 423–437 (2007)
24. Robnik-Sikonja, M., Kononenko, I.: Theoretical and empirical analysis of relieff and relieff. Machine Learning 53, 23–69 (2003)
25. Maszczyk, T., Grochowski, M., Duch, W.: Discovering Data Structures using Meta-learning, Visualization and Constructive Neural Networks. In: Advances in Machine Learning II. Studies in Computational Intelligence, vol. 262, pp. 467–484. Springer, Heidelberg (2010)
26. Grąbczewski, K., Duch, W.: The separability of split value criterion. In: Proceedings of the 5th Conf. on Neural Networks and Soft Computing, Zakopane, Poland, pp. 201–208. Polish Neural Network Society (2000)

Is It Important Which Rough-Set-Based Classifier Extraction and Voting Criteria Are Applied Together?

Dominik Ślęzak[1,2] and Sebastian Widz[3,4]

[1] Institute of Mathematics, University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
[2] Infobright Inc., Poland
Krzywickiego 34 pok. 219, 02-078 Warsaw, Poland
[3] XPLUS SA
ul. Gżegżółki 4, 02-804 Warsaw, Poland
[4] Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warsaw, Poland
slezak@infobright.com, sebastian.widz@xplus.pl

Abstract. We propose a framework for experimental verification whether mechanisms of voting among rough-set-based classifiers and criteria for extracting those classifiers from data should follow analogous mathematical principles. Moreover, we show that some of types of criteria perform better for high-quality data while the others are useful rather for low-quality data. The framework is based on the principles of approximate attribute reduction and probabilistic extensions of rough-set-based approach to data analysis. The framework is not supposed to produce the best-ever classification results, unless it is extended by some additional parameters known from the literature. Instead, our major goal is to illustrate in a possibly simplistic way that it is worth unifying mathematical background for the stages of learning and applying rough-set-based classifiers.

1 Introduction

Construction of classifiers is an important application of machine learning. In order to combine accuracy with clarity, people often use symbolic machine learning techniques, such as collections of decision rules or trees. From this perspective the algorithms developed within the framework of rough sets [1,6] can be regarded as symbolic machine learning methods. Rough-set-based classifiers usually comprise of collections of decision rules learnt in an offline fashion. Rules can be extracted explicitly [3,9], in analogy to popular machine learning rule-based approaches, or implicitly, in the process of attribute subset selection based on the concept of a decision reduct [10,13]. In this paper, we follow the latter scenario. We call it *reduct-based classification*.

A reduct-based classifier is built and used as follows: 1) find several decision reducts by following some simple optimization criteria; 2) for each object to be classified, derive from reducts the decision rules that match its values and, if multiple rules apply, use some simple voting methods. The rules do not even need to be explicitly stored, as they can be computed directly from the training data for each reduct and each new object. The only truly computationally expensive task is to find a good ensemble of reducts that do not contain redundant attributes and yield decision rules of good quality.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 187–196, 2010.

A reduct-based classifier comprises of two layers: the higher one is a set of reduct-decision pairs interpreted as *approximate* functional dependencies (or conditional independencies); the lower one corresponds to parameters deciding how to derive and vote among rules. In our previous research, we focused on that higher layer. For example, we studied relationship between a degree of attribute overlap among reducts in an ensemble and its efficiency in classification and knowledge representation. The main goal of this paper, which is verification whether the reduct construction criteria and the rule voting parameters should be based on analogous principles, refers clearly to the lower out of the above layers. Surely, once the experimental verification framework is created, we can use it to analyze other aspects of reduct-based classification as well.

The paper is organized as follows: Section 2 recalls examples of rough-set-based attribute subset selection, rule generation and voting methods. All examples are already quite well-known in the literature, although gathering them within a unified framework required a thorough study that might be regarded as one of main paper's contributions. The reported methods are based on machine learning and probabilistic interpretations of the rough set methodology [2,16]. In some aspects, e.g., with regards to the voting policies, they are far behind the state of the art [5,12]. On the other hand, equations (1) and (2), as well as Table 1 should clearly illustrate what we want to achieve.

Section 3 summarizes parameters taken into account in our experiments. It includes a number of footnotes with comments how our framework can be extended in future. In Section 4, we experimentally prove that the voting methods should indeed go in pair with the attribute selection criteria. We also show that original rough set principles yield quite surprisingly good classifiers, if applied within the framework of approximate attribute reduction [7,8]. On the other hand, the same experiments conducted on benchmark data with artificially decreased quality suggest that classifiers based on probabilistic extensions of rough sets may be more reliable. In general, most of our results should be obvious to a reader with machine learning background. However, they enable to look at reduct-based classifiers from yet uninvestigated perspective.

2 Reducts, Rules, Voting

Let us start by recalling three functions labeling attribute subsets with degrees of determining a decision attribute. Formally, we should introduce them as $\gamma(B)$ [6], $M(B)$ [7], and $R(B)$ [11], for a decision table $\mathbb{A} = (U, A \cup \{d\})$ and subsets $B \subseteq A$. We will refer to their usage as *reduction* types POS, M and R. In the following, we operate with decision classes $X \in U/\{d\}$ and indiscernibility classes $E \in U/B$[1]. Probabilities $P(\cdot)$, $P(\cdot,\cdot)$ and $P(\cdot|\cdot)$ are derived directly from the training data.

$$\begin{aligned} \gamma(B) = |POS(B)|/|U| &= \textstyle\sum_{E \in U/B : P(X^M_E|E)=1} P(E) && \textit{// reduction } POS \\ M(B) &= \textstyle\sum_{E \in U/B} P(X^M_E, E) && \textit{// reduction } M \\ R(B) &= \textstyle\sum_{E \in U/B} P(E|X^R_E) - 1 && \textit{// reduction } R \end{aligned} \tag{1}$$

[1] By $U/\{d\}$ and U/B we denote the sets of equivalence classes induced by d and B. In rough sets, they are referred as indiscernibility classes [6]. Although in this paper we deal only with a simplified framework, one needs to remember that for, e.g., numeric or ordinal attributes equivalence classes need to be replaced by some better adjusted constructs [2,4].

Table 1. Six options of weighting decisions by rules, corresponding to *voting* types $PLAIN$, $CONF$ and $COVER$, and *weighting* types $SINGLE$ and $SUPPORT$. E denotes the support of a rule's premise. X_E^* denotes X_E^M or X_E^R depending on *decision* type. In case of *reduction* type POS, the weight is assigned to a given X_E^* only if $P(X_E^*|E) = 1$.

	$SINGLE$	$SUPPORT$
$PLAIN$	1	$P(E)$
$CONF$	$P(X_E^*\|E)$	$P(X_E^*, E)$
$COVER$	$P(X_E^*\|E)/P(X_E^*)$	$P(E\|X_E^*)$

wherein decision classes

$$\begin{array}{l} X_E^M = \arg\max_{X \in U/\{d\}} P(X|E) \;\; \textit{// decision } DIRECT \\ X_E^R = \arg\max_{X \in U/\{d\}} P(E|X) \;\; \textit{// decision } RELATIVE \end{array} \tag{2}$$

can be interpreted according to the mechanism of assigning a rule supported by $E \in U/B$ with a decision class that it should point at. We refer to equations (2) as *decision* types – $DIRECT$, which means pointing at decision that is the most frequent within E, as well as $RELATIVE$, which points at decision that is the most frequent within E relative to the prior probability of that decision (cf. [16]).

All functions (1) are monotonic with respect to inclusion, i.e., for $C \subseteq B$, we have $\gamma(C) \leq \gamma(B)$, $M(C) \leq M(B)$, $R(C) \leq R(B)$ [7,11]. It is important when designing the attribute reduction criteria and algorithms (cf. [2,15]). In our experiments, we search for (F, ε)-reducts, where F may mean γ (POS), M or R, and $\varepsilon \in [0, 1)$ decides how much of quality of determining d we agree to lose when operating with smaller subsets $B \subseteq A$ (thus, shorter rules), according to the following constraint:

$$F(B) \geq (1 - \varepsilon)F(A) \quad \textit{// Approximate Attribute Reduction Criterion} \tag{3}$$

Surely, there are many attribute reduction heuristics (cf. [4,15]). An advantage of the following one is that it yields multiple reducts. It is based on random generation of permutations of attributes. Each permutation τ is used as an input into so called (F, ε)-REDORD algorithm, which tries to remove τ-consecutive attributes with no loss of inequality (3). Such obtained (F, ε)-reducts can be sorted with respect to some simple optimization criteria in order to select those that should be taken into the ensemble. In [8,14], one can find references to a more sophisticated *order-based genetic algorithms (o-GA)* that encode permutations of attributes as chromosomes. However, according to experiments reported in [10,13], letting the (F, ε)-REDORD algorithm work with totally randomly chosen permutations provides satisfactory results as well.

Once the (F, ε)-reduct ensemble is established, it remains to specify the weights that rules should give to their decisions when voting about new objects. Table 1 illustrates parameters that we take into account with this respect. There are three *voting* types: with 1 ($PLAIN$), with rule's confidence ($CONF$), and rule's confidence divided by decision's prior probability ($COVER$)[2]. We can also strengthen the rule's vote using its premise's support (*weighting* type $SUPPORT$) or not ($SINGLE$).

[2] In case of (POS, ε)-reducts we vote only using the rules with confidence equal to 1.

Table 2. All parameters used in our experiments. *Reduction* type and $\varepsilon \in [0,1)$ influence the way of computing reducts. *Incompleteness* corresponds to preparation of test data sets. The other parameters decide how the rules induced from reducts are applied to classify new objects.

Parameter	Parameter's Values
Reduction	$\{POS, M, R\}$
Approximation	$\varepsilon \in [0,1)$
Decision	$\{DIRECT, RELATIVE\}$
Voting	$\{PLAIN, CONF, COVER\}$
Weighting	$\{SINGLE, SUPPORT\}$
Matching	$\{STRICT, FLEXIBLE\}$
Incompleteness	$\{0, 1, 3, 5, 7, 9, 10, 20, 30\}$ (%)

3 Experimental Framework

We analyzed seven data sets taken from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/): Optical Recognition of Digits (*optdigits*), Pen-Based Recognition of Digits (*pendigits*), Letter Image Recognition (*letter*), StatLog version of DNA-splices (*dna*) and data sets related to the Monk's Problems (*monks*-1, *monks*-2, *monks*-3). All these data sets are by design split onto the training and testing samples. The experiments were performed according to the following procedure:

1. Generate randomly k permutations τ over the set of conditional attributes[3]
2. For each τ, $\varepsilon \in [0,1)$ and $F \in \{POS, M, R\}$, apply (F,ε)-REDORD [8,14]
3. Choose n (F,ε)-reducts that are best according to simple optimization criteria[4]
4. For each given *incompleteness* level, remove randomly chosen attribute values
5. For each object to be classified, use each of reducts to derive rules that match it[5]
6. For each combination of parameter settings, compute accuracy over the test data[6]

Table 2 summarizes parameters used in experiments. *Matching* types are described in Section 4. Before this, it is important to explain why we consider *incompleteness* levels. In our previous research [13], we studied correlation between data quality and optimal settings of $\varepsilon \in [0,1)$. In [13], evaluation of data quality was simple – given the task of MRI segmentation, the quality was related to the MRI slice thickness and the resulting inaccuracy of attribute values. Here, if we wanted to simulate the same for each of considered benchmark sets, we would have to carefully follow their semantics in order to introduce inaccuracies appropriately. Instead, we suggest that removing values from the test samples is perhaps a naïve but still valid way of lowering data quality. Surely, this is just a simulation, incomparable to dealing with truly incomplete data [3,9].

[3] In our experiments we use $k = 1,000$.

[4] We choose $n = 10$ reducts $B \subseteq A$ that minimize $|B|$ (first criterion) and $|U/B|$ (second criterion); this means, we minimize number of rules induced by reducts) [1,14]. One can also choose reducts that are reached by the highest number of permutations [8].

[5] More sophisticated rough-set-based methods involve filtering out rules with low support [1,14]. We omit this option for simplicity. On the other hand, let us note that in case of POS-based approach we implicitly filter out rules with confidence below 1. Let us also emphasize one more time that rule generation needs to be adjusted to the attribute types [2,4].

[6] For each combination, the experiment was repeated 10 times and the results were averaged.

4 Experimental Results

The procedure described in Section 3 leads towards a kind of results' repository that is useful for verifying various hypotheses. The idea is to analyze particular parameters shown in Table 2 by letting the remaining ones maximize the classifiers' accuracy. For example, Fig. 1 shows the maximum accuracies that can be obtained for particular settings of *reduction* type and $\varepsilon \in [0, 1)$, and by varying *decision*, *voting* and *weighting* types. In this case, the comparison is drawn only for original test sets, i.e., with 0% of *incompleteness*. Hence, we ignore *matching* type that controls how to handle partially incomplete test cases (see further below).

Fig. 1 shows an interesting trend, which was first observed while studying classification of gene expression data in [10]. It turns out that for $\varepsilon = 0$, which is the case of the original rough-set-based attribute reduction framework, more sophisticated functions, such as M or R, lead to more efficient reducts/rules than POS. However, POS catches up for higher approximation thresholds. Actually, the ability of POS to maximize accuracy with respect to $\varepsilon \in [0, 1)$ is usually higher than in case of its M- and R-based equivalents[7]. Given simplicity of (POS, ε)-based classification, this is quite a surprising result that requires further study.

The second part of our experiment relates to the thesis that the mechanism of inducing approximate reducts and rules from data should have the same mathematical background as the mechanism of using these rules in voting about new objects. Fig. 2 shows the maximum accuracies (with respect to other parameters) of classifiers based on (M, ε)-reducts and on different *voting* types. From mathematical point of view it is clear that $PLAIN$-voting does not match M-based reduction. Function M involves careful analysis of the confidence values $P(X_E^*|E)$ weighted by $P(E)$. Experimental results prove that ignoring the values of $P(X_E^*|E)$ and voting simply with 1 or $P(E)$ is not a good idea in case of M-based attribute reduction.

Fig. 3 goes back to the comparison of *reduction* types, now also involving the level of data quality modeled by introducing *incompleteness* into the test data. The goal is to check whether varying data quality may influence the trends observed in Fig. 1. The results show that an advantage of M-/R-based reduction over POS tends to grow together with *incompleteness* levels. Although our method of simulating lower-quality data may be regarded as oversimplified, it still shows that M-/R-based classifiers have better chances to provide good accuracies in real-life applications.

The results illustrated by Fig. 3 were obtained using so called $STRICT$ *matching* type, which means that a given test object can be classified by a decision rule based on a reduct $B \subseteq A$ only if it does not miss any of the values on attributes in B. Clearly, it is not the only way of dealing with such cases. Fig. 4 shows an alternative *matching* type, referred as $FLEXIBLE$, wherein, whenever there are misses at some $C \subseteq B$, we generate the decision rule based on the set $B \setminus C$. This latter option is significantly better. However, its computational complexity is prohibitive for large data sets. Further research is needed with this respect, taking into account both machine learning and rough-set-based methods developed for incomplete data [3,9].

[7] When looking at the figures, remember that our main goal in this paper is not reporting exact percentages of classification accuracy, but rather illustrating some comparative trends.

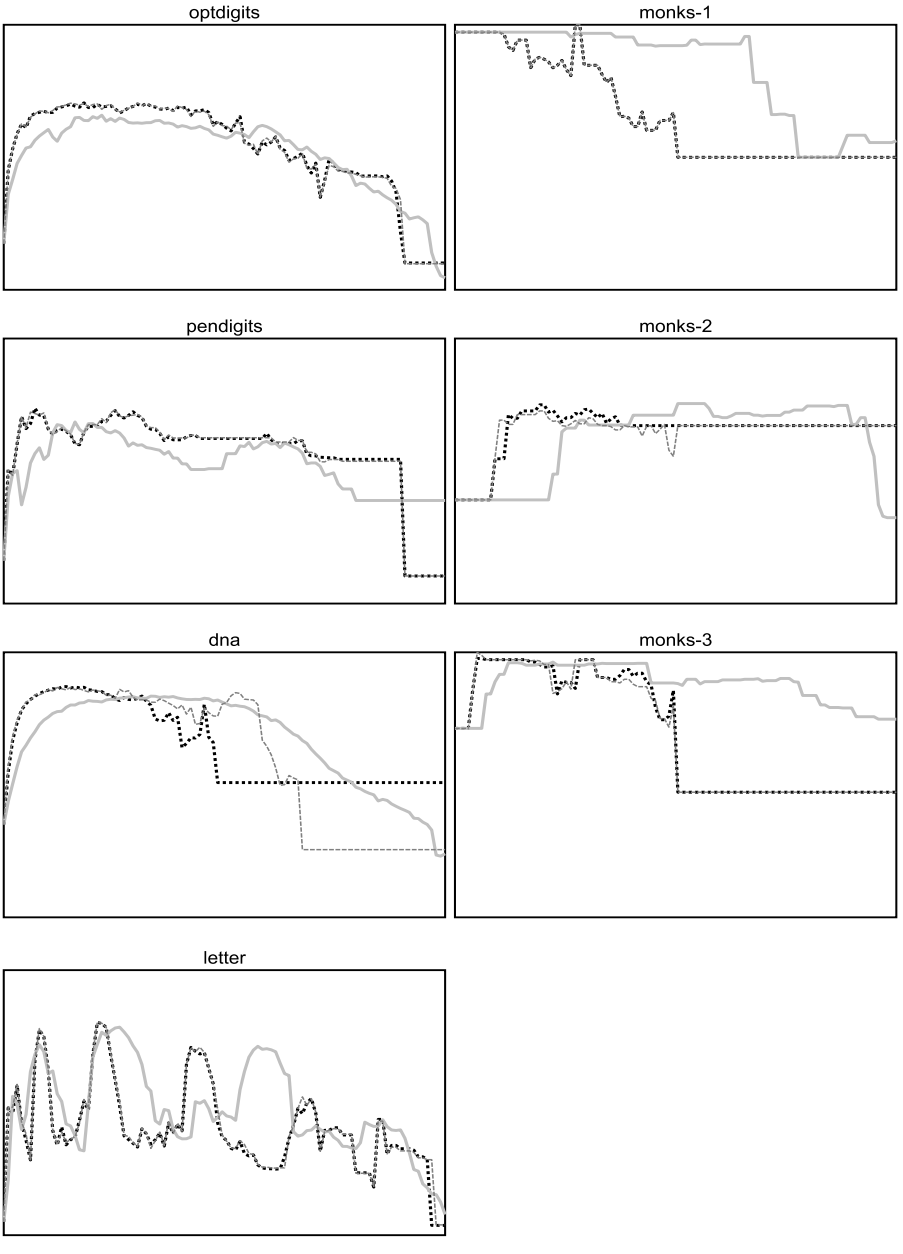


Fig. 1. Classification accuracies obtained for three *reduction* types: POS (light-grey-solid), M (black-dotted) and R (dark-grey-dashed). X-axis corresponds to approximation threshold $\varepsilon \in [0, 1)$ (which increases from left to right). Y-axis corresponds to the maximum accuracy out of all classifiers constructed based on $POS/M/R$ and ε. POS is more frequently better than M/R for higher settings of ε. Also, POS usually leads to the best or at least comparable maximum and average scores over the whole range of ε.

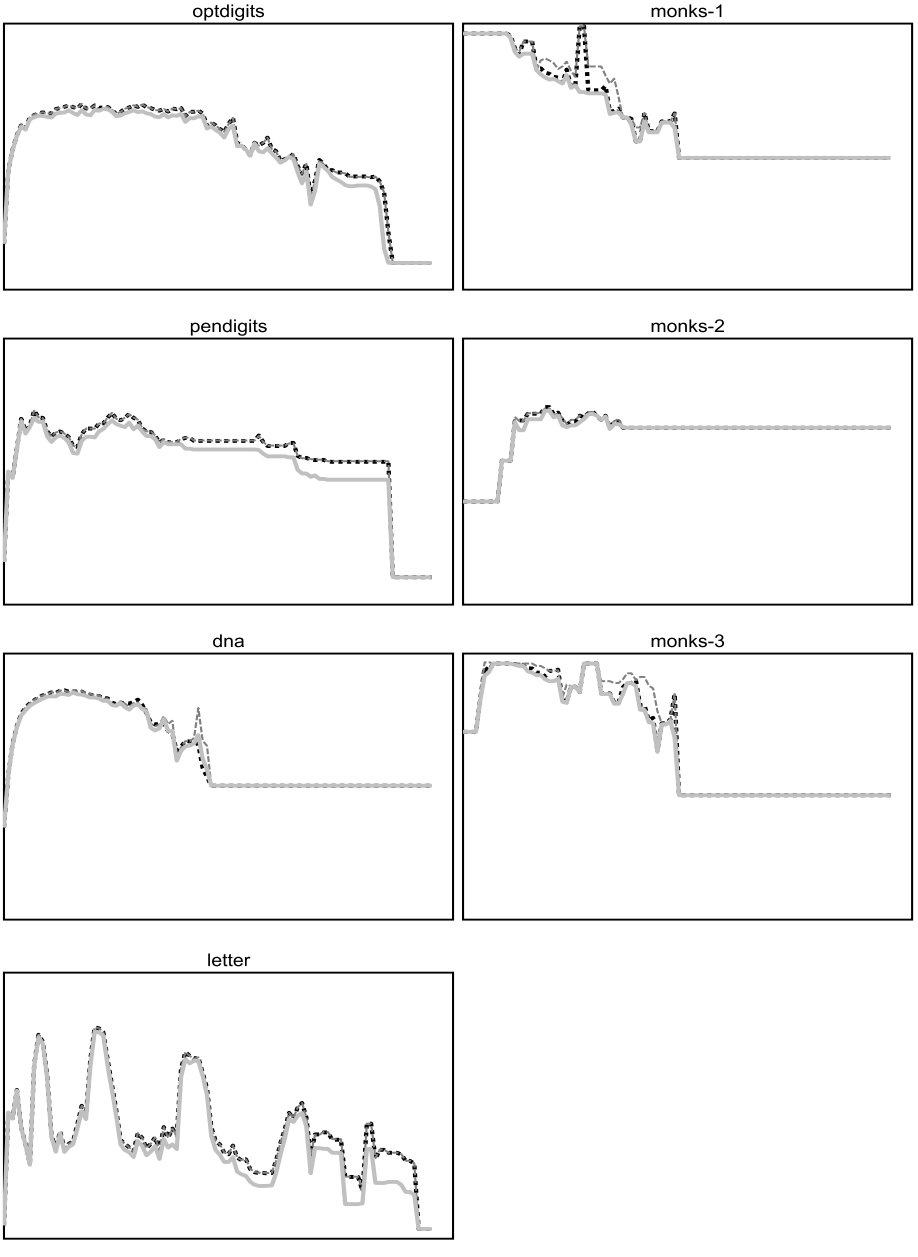


Fig. 2. Classification accuracies obtained for three *voting* types: $PLAIN$ (light-grey-solid), $CONF$ (black-dotted) and $COVER$ (dark-grey-dashed). X-axis means the same as in Fig. 1. However, *reduction* type is now fixed to M. Thus, Y-axis corresponds to the maximum accuracy out of all classifiers based on M, $\varepsilon \in [0, 1)$ and $PLAIN$/$CONF$/$COVER$. One can see that $CONF$ and $COVER$ are never worse and usually quite better than $PLAIN$, when applied together with *reduction* type M.

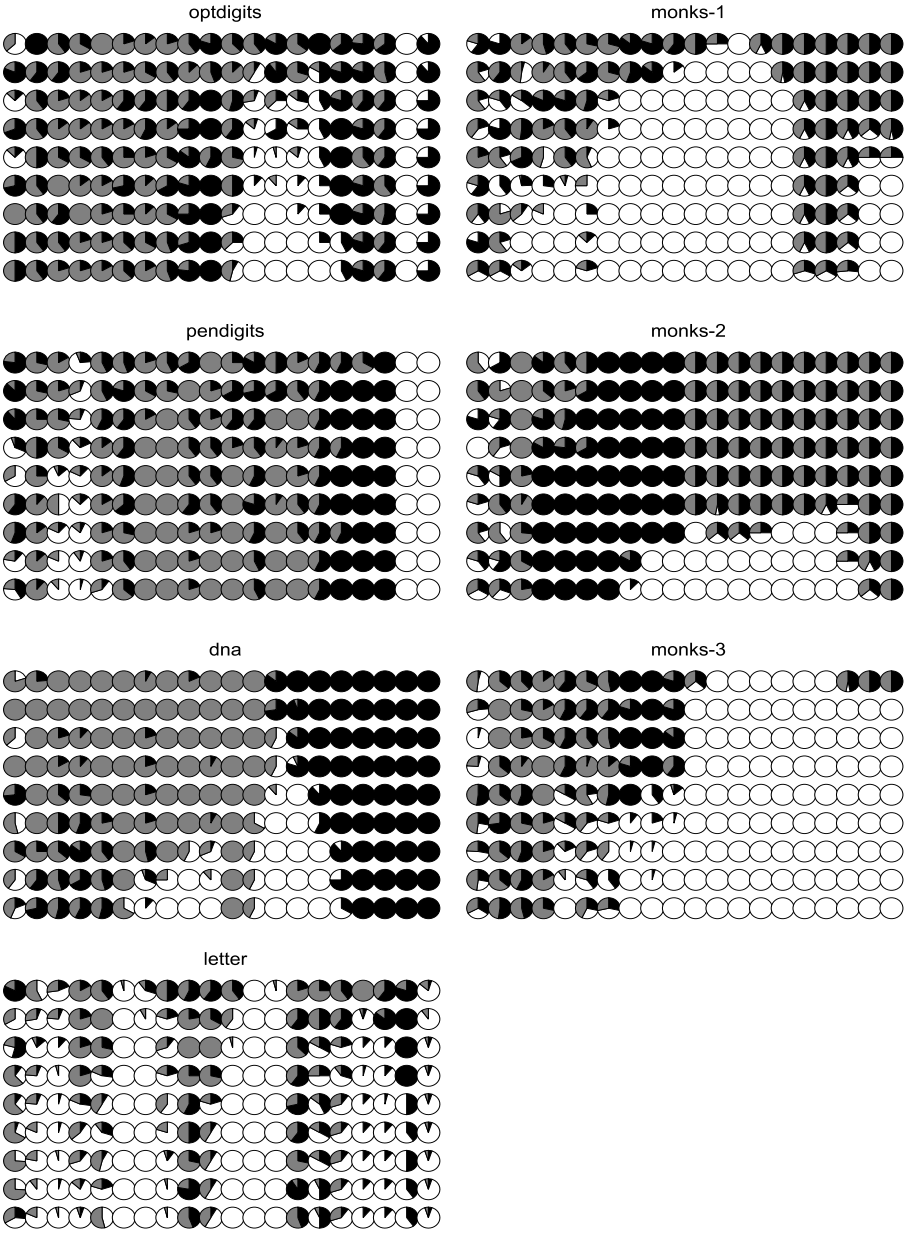


Fig. 3. The pies correspond to various combinations of the settings of $\varepsilon \in [0, 1)$ (X-axis; ε increases from the left to the right) and *incompleteness* (Y-axis; 0% at the bottom toward 30% at the top – see Table 2). The pie area represents distribution of classifiers with best accuracy that are based on particular *reduction* types: POS (white), M (black), R (grey). It shows that POS usually stops being competitive when comparing to M/R at higher *incompleteness* levels.

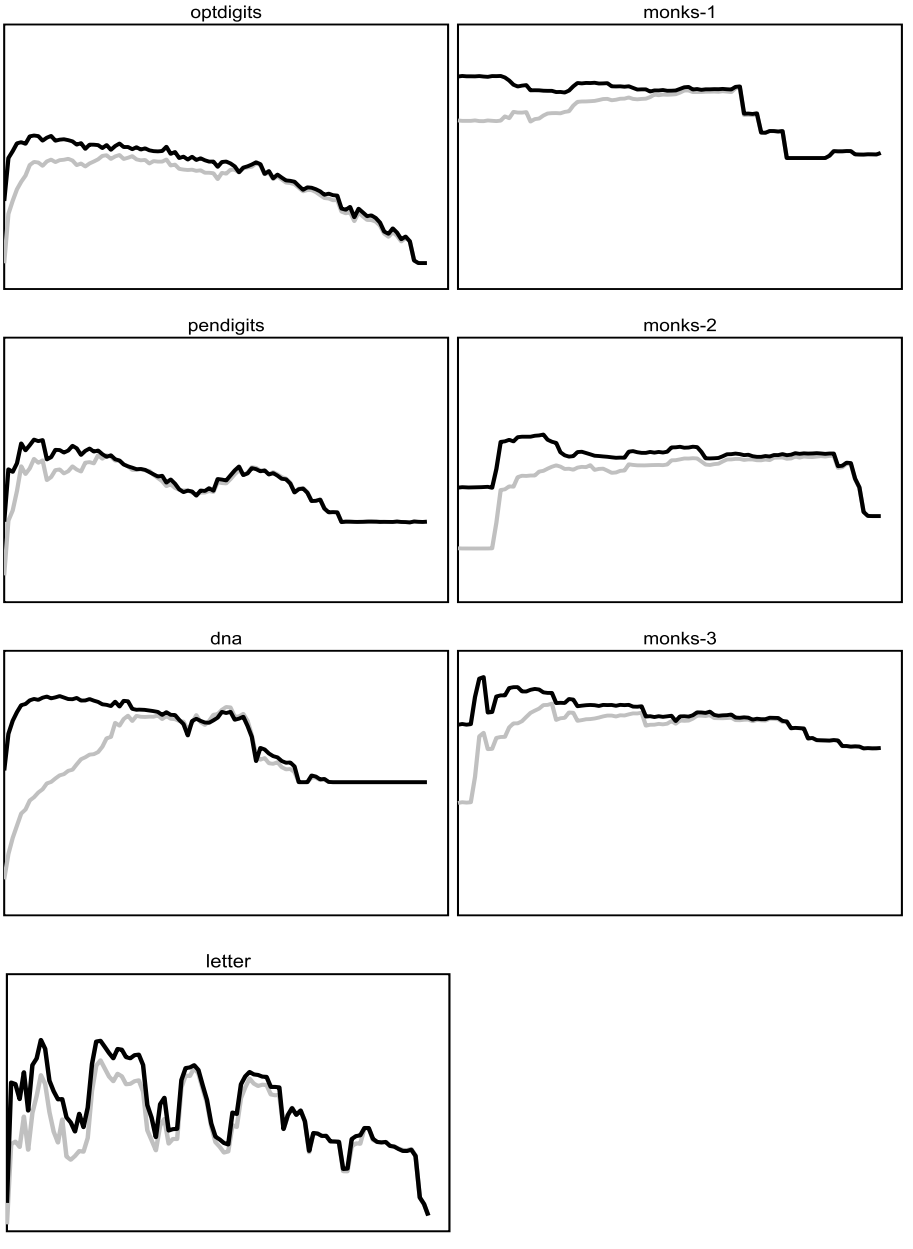


Fig. 4. Classification accuracies obtained for two *matching* types: $STRICT$ (light-grey) and $FLEXIBLE$ (black). X-axis means the same as in Fig. 1. Y-axis corresponds to maximum accuracy scores, like in Fig. 1, but now maximized also with respect to *reduction* types. Maximum scores are computed for each of considered *incompleteness* levels (reported in Table 2) and then averaged. We can see that $FLEXIBLE$ is better than $STRICT$, although the difference fades away when $\varepsilon \in [0, 1)$ increases.

5 Conclusions

We introduced a framework for experimental analysis of so called reduct-based classifiers that origin from the theory of rough sets. The goal was to verify hypotheses taken from our previous research, e.g., whether it is truly important to use analogous mathematical principles at the phases of classifier construction and voting. Although our framework reflects the state of the art in the areas of rough-set-based and, more generally, symbolic machine learning classification in a highly simplified way, the described experiments support our claims.

Acknowledgements. This paper was supported by grants N N516 368334 and N N516 077837 from the Ministry of Science and Higher Education of the Republic of Poland. Authors would like to thank the Reviewers for their valuable comments on this article.

References

1. Bazan, J.G., Nguyen, H.S., Nguyen, S.H., Synak, P., Wróblewski, J.: Rough Set Algorithms in Classification Problem. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.) New Developments in Knowledge Discovery in Information Systems, pp. 49–88. Physica Verlag, Heidelberg (2000)
2. Błaszczyński, J., Greco, S., Słowiński, R., Szeląg, M.: Monotonic Variable Consistency Rough Set Approaches. Int. J. Approx. Reasoning 50(7), 979–999 (2009)
3. Grzymała-Busse, J.W.: Rule Induction, Missing Attribute Values and Discretization. Encyclopedia of Complexity and Systems Science, 7797–7804 (2009)
4. Jensen, R., Shen, Q.: Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches. Wiley, IEEE (2008)
5. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley, Chichester (2004)
6. Pawlak, Z., Skowron, A.: Rudiments of Rough Sets. Inf. Sci. 177(1), 3–27 (2007)
7. Ślęzak, D.: Normalized Decision Functions and Measures for Inconsistent Decision Tables Analysis. Fundam. Inform. 44(3), 291–319 (2000)
8. Ślęzak, D.: Rough Sets and Functional Dependencies in Data: Foundations of Association Reducts. T. Computational Science 5, 182–205 (2009)
9. Ślęzak, D., Sakai, H.: Automatic Extraction of Decision Rules from Non-deterministic Data Systems: Theoretical Foundations and SQL-Based Implementation. In: Proc. of DTA 2009. CCIS, vol. 64, pp. 151–162. Springer, Heidelberg (2009)
10. Ślęzak, D., Wróblewski, J.: Roughfication of Numeric Decision Tables: The Case Study of Gene Expression Data. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 316–323. Springer, Heidelberg (2007)
11. Ślęzak, D., Ziarko, W.: The Investigation of the Bayesian Rough Set Model. Int. J. Approx. Reasoning 40(1-2), 81–91 (2005)
12. Szczuka, M.S.: Refining classifiers with neural networks. Int. J. Intell. Syst. 16(1), 39–55 (2001)
13. Widz, S., Ślęzak, D.: Approximation Degrees in Decision Reduct-Based MRI Segmentation. In: Proc. of FBIT 2007, pp. 431–436. IEEE CS, Los Alamitos (2007)
14. 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)
15. Yao, Y.Y., Zhao, Y., Wang, J.: On Reduct Construction Algorithms. T. Computational Science 2, 100–117 (2008)
16. Ziarko, W.: Probabilistic Approach to Rough Sets. Int. J. Approx. Reasoning 49(2), 272–284 (2008)

Improving Co-training with Agreement-Based Sampling

Jin Huang[1], Jelber Sayyad Shirabad[1], Stan Matwin[1,2], and Jiang Su[1]

[1] School of Information Technology and Engineering, University of Ottawa, Canada
{jhuang33,jsayyad,stan,jsu}@site.uottawa.ca
[2] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Abstract. Co-training is an effective semi-supervised learning method which uses unlabeled instances to improve prediction accuracy. In the co-training process, a random sampling is used to gradually select unlabeled instances to train classifiers. In this paper we explore whether other sampling methods can improve co-training performance. A novel selective sampling method, agreement-based sampling, is proposed. Experimental results show that our new sampling method can improve co-training significantly.

1 Introduction

In some real world applications, it is difficult or expensive to obtain enough labeled data to train a strong classifier. At the same time, large quantities of unlabeled data are often available. Semi-supervised learning uses both labeled and unlabeled instances to build strong classifiers.

Some commonly used semi-supervised learning algorithms include generative methods such as EM (Expectation Maximization) [1], graph-based methods [2], and co-training [3]. Co-training [3] is one of the most widely used semi-supervised learning method. It assumes that the data can be described as two disjoint views; i.e. sets of features. In co-training, two classifiers are first trained on the initial (small) labeled training set using the two separate views. During the algorithm iterations, each classifier supplies the other one with a set of new positive and negative examples. These examples are selected from a working pool which is replenished by randomly sampling from the set of unlabeled instances. Blum and Mitchell (1998) [3] showed that when the two views are independent and each one is sufficient for learning, co-training can effectively build an efficient model. However, in real world applications this assumption is seldom held. Some previous research [4,5] showed that while co-training may still be effective in the case of dependent data views, its performance is usually worse than using independent and sufficient data views.

Active learning, similar to semi-supervised learning, uses both labeled and unlabeled instances to build strong classifiers. Its difference with semi-supervised learning is that in active learning a domain expert will assign labels to some "most informative" unlabeled instances. Therefore, sampling the most informative unlabeled instances from a large unlabeled instance pool is the key issue for

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 197–206, 2010.

active learning. Previous research shows that in active learning, random sampling usually leads to poor performance. Significant research in active learning has focused on designing new sampling strategies [6,7,8]. In the original co-training algorithm, there is also a sampling process which samples some unlabeled instances from a large pool to replenish a small working pool. The original co-training algorithm [3] uses a simple random sampling which may result in different unlabeled instances to be used in different co-training runs. Since in active learning random sampling usually cannot achieve the best performance, we would like to know if we can use other sampling strategies to obtain better results in the case of co-training. To the best of our knowledge, no previous work demonstrated that sampling is also an important factor that influences co-training performance. In this paper, we propose a simple selective sampling method to improve co-training. We performed experiments using this new sampling method on several UCI datasets. The results for several datasets show that, for the same view attributes, our new method can result in significantly more accurate classifiers when compared with the original co-training.

2 Sampling in Co-training

The co-training algorithm is depicted in Figure 2. A small unlabeled set U' is first randomly sampled from unlabeled dataset U. Then in the co-training process, two classifiers $h1$ and $h2$ trained on two views $V1$ and $V2$ of the labeled dataset L are used to label all unlabeled instances in U'. Classifiers $h1$ and $h2$ each select p positive and n negative most confidently predicted instances from U' and add them to L. The unlabeled set U' is then replenished by random sampling from U. This process is repeated many times until the co-training is finished.

Since random sampling usually does not achieve the best performance, we explore whether another sampling method can improve the co-training performance. We review some research on sampling in active learning because sampling plays an important role in active learning. Tong and Koller [6] used a sampling strategy by minimizing the version space to improve active learning for support vector machines. Muslea and Minton [9] proposed an active sampling method which is called co-testing. Similar to co-training, co-testing also trains two classifiers on two views. The two view classifiers are used to classify all unlabeled instances. Some of the instances that theses classifiers disagree the most on their label are then presented to domain experts for labeling. This sample of expert labeled instances is then added to the labeled instance set.

Intuitively, one may say that we can directly apply the sampling methods used in active learning to the co-training process. Unfortunately, this approach is infeasible. Co-training is actually a passive learning process. The sampling process in active learning selects the most informative unlabeled instances to be labeled by domain expert. Those most informative unlabeled instances are usually least confidently predicted by the classifiers used in co-training. If those unlabeled instances are sampled in co-training, they are very likely to be mislabeled by the view classifiers. This will degrade the co-training performance.

Given:

- a learning problem with two views $V1$ and $V2$
- a learning algorithm
- the sets L and U of labeled and unlabeled examples
- the number k of iterations to be performed
- a working set U' with a given size u

Co-train(L, U, k, V1, V2, u, sample-func):

- let h1 and h2 be the classifiers learned from training data L using views V1 and V2, respectively
- $U' =$ *sample-func(u)* from U
- LOOP for k iterations
  - select the most confidently predicted $p + n$ instances by $h1$ from U'
  - select the most confidently predicted $p + n$ instances by $h2$ from U'
  - add $2p + 2n$ instances to L
  - remove $2p + 2n$ instances from U'
  - $U' =$ *sample-func*$(2p + 2n)$ from U
  - let $h1$ and $h2$ be the two classifiers trained on two views from L
- combine the prediction of $h1$ and $h2$

random-sample(n):

- randomly select and extract n instances from U
- return selected n instances

sample-with-agreement(n):

- use $h1$ and $h2$ to classify all instances in U
- FOR each instance x_i in U
  - $mean(x_i) = (p_1(x_i) + p_2(x_i))/2$
  - Let the score function
  - $s(x_i) = I(x_i) + max\{mean(x_i), 1 - mean(x_i)$
- rank all instances of U according to score function values in decreasing order
- return the top ranked n instances

Co-training = Co-train(L, U, k, V1, V2, u, random-sample)
Co-training with agreement sampling = Co-train(L, U, k, V1, V2, u, sample-with-agreement)

Fig. 1. Co-train and co-train-AS

Therefore the sampling strategies adopted by active learning cannot be directly applied in co-training. Consequently, we have to design a new sampling method for co-training.

3 Agreement-Based Sampling for Co-training

We propose a novel sampling method to replace the random sampling used by co-training. This new strategy is called agreement-based sampling. It is motivated by the co-testing method proposed in [9]. Co-testing is an active learning method which borrows the idea of co-training. Co-testing also uses redundant views to expand the labeled dataset to build strong learning models. The major difference between co-testing and co-training is that co-testing uses two classifiers to sample unlabeled instances to be labeled by the domain experts, while co-training randomly samples some unlabeled instances and uses the two view classifiers to assign labels to them. Results presented in [9] showed that co-testing benefits from the sampling method it uses. This motivated us to use a better sampling method for co-training.

Co-testing first trains the view classifiers on the two views of the labeled dataset. It then uses the two view classifiers to classify all unlabeled instances. The unlabeled instances that the two view classifiers *disagree the most* on their classification are then sampled. In our method we also use two view classifiers to sample from unlabeled instances. We first use the two view classifiers to classify all unlabeled instances. But we always sample unlabeled instances from U that the two view classifiers *agree the most* on their classification. These are used to replenish U'. We use a ranking function to rank all the unlabeled instances according to the predictions of the two view classifiers. The ranking score function for an unlabeled instance x_i is defined as

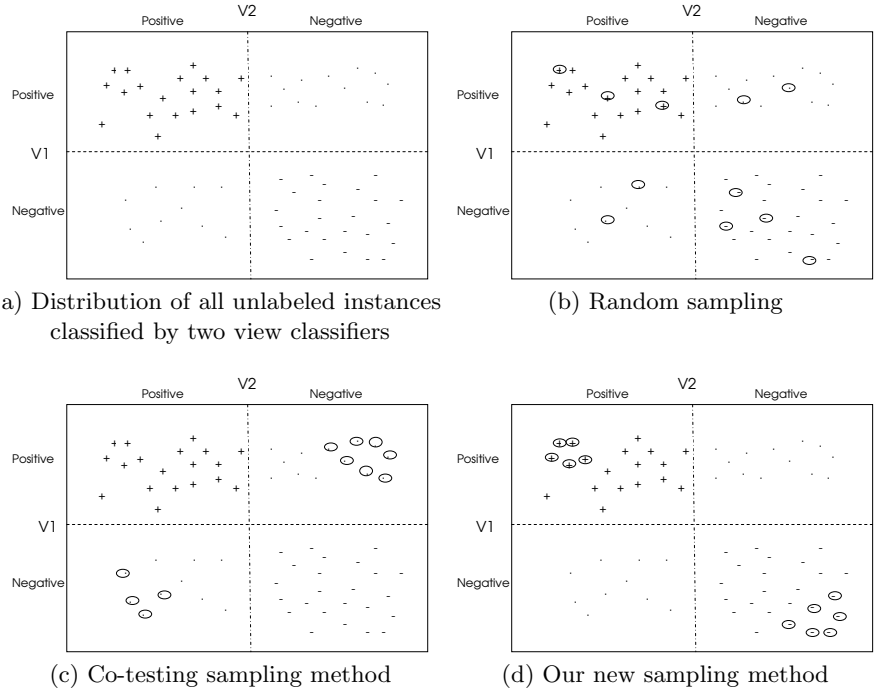
$$s = I(x_i) + max\{(p_1((x_i) + p_2(x_i))/2, 1 - (p_1(x_i) + p_2(x_i))/2\} \quad (1)$$

where

$$I(x_i) = \begin{cases} 1 \text{ if the two view classifiers assign the same label to } x_i \\ 0 \text{ otherwise} \end{cases}$$

$p_1(x_i)$ and $p_2(x_i)$ are predicted probabilities for the positive class by the two view classifiers respectively.

Scores generated by formula 1 results in a rank where instances in the highest positions are the ones that both view classifiers assign the same label, with high confidence., to them. The first term $I(x_i)$ guarantees that the unlabeled instances for which the two view classifiers produce the same labels are always ranked higher than those given different labels. The term $max\{(p_1(x_i) + p_2(x_i))/2, 1 - (p_1(x_i) + p_2(x_i))/2\}$ selects the larger one of the average predicted probabilities for the positive and negative classes by the two view classifiers. Therefore the instances with higher predicted confidence are given higher scores. Figure 2 shows the original co-training algorithm and co-training using agreement-based



(a) Distribution of all unlabeled instances classified by two view classifiers

(b) Random sampling

(c) Co-testing sampling method

(d) Our new sampling method

Fig. 2. Comparison of different sampling methods

sampling. Here we use Co-train-AS to refer to co-training using agreement-based sampling.

To better understand different sampling methods, In Figure 2 we compare random sampling, co-testing sampling and our new sampling method. For a given labeled dataset, we assume two view classifiers $h1$ and $h2$ are trained on views $V1$ and $V2$. The two view classifiers are then used to classify all unlabeled instances. Figure 2(a) shows the distribution of the classified unlabeled instances. Here the horizontal line $V1$ represents the classification boundary of the view classifier $h1$. Instances that lie above the line $V1$ are classified as positive by $h1$; while those that lie below this line are classified as negative. The vertical line $V2$ represents the classification boundary of the view classifier $h2$. Similarly, instances to the left of the line $V2$ are classified as positive by $h2$; otherwise they are classified as negative. The confidence of classification for an instance is proportional to its distance to the boundary. The farther an instance is from the boundary, the higher is the confidence in its label.

In Figure 2(a) the instances which are in the upper left and lower right regions are assigned the same labels by the two view classifiers. We use "+" and "-" to represent the unlabeled instances that are classified as positive and negative by both $h1$ and $h2$. The instances in the other two regions are those that the two view classifiers disagree on their label. We use "." to represent these instances.

Figure 2(b) shows how random sampling method works. It randomly selects some instances from all unlabeled instances. We use circles to represent the sampled instances. Figure 2(c) shows the sampling method of co-testing. It clearly shows that co-testing samples the instances that the two view classifiers label differently and that are farthest to both classification boundaries. Figure 2(d) shows our new sampling method where instances that both view classifiers label the same and that are also farthest from the boundaries are selected.

4 Experiments

We chose 16 UCI datasets [10] to investigate the performance of using agreement-based sampling for co-training.The characteristics of each dataset along with the co-training setup used are shown in Table 1. The attributes of all datasets are split into two sets and are used as two co-training views. This is a practical approach for generating two views from a single attribute set. To thoroughly explore the co-training performance on these views we would like to experiment with all possible combinations of the views. Unfortunately, since the number of pairs of views is exponential to the number of attributes, it is impossible to enumerate all such pairs when the number of attributes is large. In our experiments we randomly generate some pairs of views. For all datasets except for balance-scale we randomly select at least 150 pairs of views to run co-training. The last column of Table 1 represents the number of view pairs used in our experiments and the total number of all possible view splits. The balance-scale, primary-tumor, splice, soybean, segment, and vowel multi-class datasets are converted to binary by grouping some classes as positive and the rest as negative. The resulting balance-scale, splice, soybean, and segment binary datasets are roughly balanced. In the case of the balance-scale dataset, "L" and "B" classes are categorized as the positive class, and "R" class as the negative

Table 1. Datasets and co-training settings used in our experiments

Dataset	Instances	Attributes	Class Dist.	Co-training settings						
				$\|L\|$	$\|U\|$	$\|U'\|$	$\|test\|$	p	n	V
anneal	898	39	1:3.2	12	612	50	224	1	3	$150/2^{38}$
balance-scale	625	5	1:1	8	411	50	156	1	1	15/15
breast-c	286	9	1:2.3	24	190	50	72	1	2	255/255
credit-a	690	16	1:1.3	23	497	50	170	1	1	$400/2^{15}$
colic	368	22	1:1.7	12	260	50	96	1	2	$250/2^{21}$
diabetes	768	9	1:1.7	12	564	60	192	1	2	255/255
hypothyroid	3772	30	1:12	39	2634	156	943	1	12	$200/2^{29}$
letter	20000	17	1:1	10	14492	500	5000	1	1	$150/2^{16}$
primary-tumor	339	18	1:3.2	16	188	50	85	1	3	$200/2^{17}$
segment	2310	20	1:1.3	10	1620	100	580	1	1	$150/2^{19}$
sick	3772	30	1:15	48	2805	288	919	1	15	$200/2^{29}$
soybean	683	36	1:1	8	504	50	171	1	1	$200/2^{35}$
splice	3190	60	1:1	12	2328	50	800	1	1	$100/2^{59}$
vehicle	846	19	1:3	12	562	60	212	1	3	$200/2^{18}$
vote	435	17	1:1.6	26	300	50	109	1	2	$200/2^{16}$
vowel	990	14	1:4.5	20	647	75	248	1	4	$200/2^{13}$

Table 2. Classification error rates for co-train and co-train-AS

	Naive Bayes		Decision Tree	
Dataset	cotrain	cotrain-AS	cotrain	cotrain-AS
anneal	12.2±3.3	11.5±3.6	12.5±2.9	11.9±2.9
balance-scale	20.6±3.6	20.7±3.5	42.1±4.2	41.6±4.5
breast-c	34.1±2.7	31.4±2.8 ○	36.7±2.5	33.2±2.8 ○
credit-a	16.0±2.1	13.2±2.1 ○	22.6±3.1	21.9±3.1
colic	31.1±3.1	27.7±3.4 ○	33.5±3.7	33.1±3.1
diabetes	25.3±2.4	28.4±2.4 ●	27.0±3.2	28.5±3.6 ●
hypothyroid	22.6±3.0	21.7±3.2 ○	21.6±3.0	19.1±3.2 ○
letter	25.5±3.2	25.9±3.3	23.8±3.0	23.9±2.8
primary-tumor	41.5±4.4	41.5±4.8	44.3±4.2	43.9±4.3
segment	12.9±3.2	13.1±3.5	15.2±3.4	14.4±3.0
sick	11.6±2.5	11.7±2.5	12.8±1.1	12.5±1.2
soybean	20.7±3.5	18.5±3.1 ○	17.6±3.5	15.4±3.2 ○
splice	12.7±3.9	11.0±4.1 ○	13.8±4.1	13.3±4.2
vehicle	33.8±3.1	34.5±3.5	37.1±3.3	37.2±3.2
vote	12.2±3.0	11.3±2.8	13.8±3.0	14.1±2.7
vowel	24.1±2.9	23.9±3.3	24.0±3.0	21.4±2.8 ○

○: significantly better than original co-training.
●: significantly worse than original co-training.

class. All of the primary-tumor dataset classes except "lung" are categorized as negative. "EI" and "IE" classes of the splice dataset are categorized as positive, and class "N" as negative. The first 9 classes of the soybean dataset are categorized as the positive class and the remaining 10 classes as negative. The first 3 classes of the segment dataset, are relabeled as the positive class and the remaining 4 as the negative class. Finally, the first 2 classes of the vowel dataset are categorized as the positive class and the remaining 9 classes as the negative class.

Our experiments compared the performance of the the original co-training and the new agreement-based sampling method (cotrain-AS) . We used Naive Bayes and J48 decision tree learning algorithms. In the case of Naive Bayes learning, numeric attributes of all datasets are discretized by using the ten-bin unsupervised discretization method of Weka [11]. We ran each method on each dataset 10 times. In each run we split the whole dataset into 5 equal-sized non-overlapping subsets. We repeatedly used each subset as the testing set. The remaining four subsets formed a randomly selected labeled set L and unlabeled set U. We then ran the original and the new co-training algorithm using this setup and measured the performance of the generated classifiers on the independent testing set.

Semi-supervised learning can be viewed from two perspectives. First, it is used to build a strong model from labeled and unlabeled instances. Second, it is used to expand the original limited labeled dataset. In many real world applications, the choice of the required model is constrained. For instance one may need an explainable model which eliminates the use of black box algorithms such as SVMs. We call this classifier a *modeling classifier*. The question then is, *How can we use an unlabeled dataset to train the desired modeling classifier?* The obvious answer is to somehow obtain the label of these unlabeled instances and then train the modeling classifier using this larger dataset.

We believe co-training, viewed as a method to automatically expand an initial small labeled set, could be a cost effective answer to this question. Consequently, the quality of the resulting expanded labeled set is of particular interest to us.

We train a *modeling classifier* at the end of the co-training process and evaluate its performance on the test split mentioned above. The modeling classifier is trained on all attributes of the expanded labeled set. For the experiments reported in this paper we have chosen the same learning algorithm for the view classifiers and the modeling classifier, but this is not a necessary condition. Table 2 shows the average error rates of the modeling classifiers for different datasets.

We performed a paired t-test with 95% confidence level on the error rates to show whether one model is significantly different then the other. Our results show that when using naive Bayes, cotrain-AS is significantly better than original co-training in 7 out of 16 cases. This indicates that cotrain-AS can indeed make improvements over the original co-training. Table 2 shows that for only one dataset, diabetes, the original co-training is significantly better than cotrain-AS. The detailed experimental results on diabetes show that the two view classifiers perform quite differently in many view splits. In many cases they give totally different labels to unlabeled instances. Therefore a method that only samples the instances that are assigned the same label by both view classifiers becomes less reliable and may result in degradation of performance.

In the case of decision tree learning, Table 2 shows that cotrain-AS is significantly better than original co-training in 4 out of 16 datasets. These results indicate that our new method can be more beneficial when using naive Bayes versus J48 decision tree learning.

5 Discussion

In this section we investigate why agreement-based sampling method can improve the performance of co-training. As shown in Section 3, agreement-based sampling selects the unlabeled instances that the two view classifiers agree the most about their label. When two view classifiers are sufficient and independent, the sampled instances are more reliably labeled. Thus selecting those instances that the two view classifiers agree on their label is less likely to introduce errors in expanded labeled dataset. On the other hand, co-testing sampling method selects instances that the two view classifiers disagree the most on their labels. This means that one of the view classifiers assigns the wrong label to the instance, which may lead to labeling errors in the expended labeled dataset. One approach to investigate the reason why our new sampling method works is to explore the labeling errors.

In a deployed application, we can not calculate the labeling error rates as the real labels of unlabeled instances are not known. But since the datasets used in our experiments are labeled, it is possible for us to calculate the labeling error rates. We calculated the labeling error rate of the original co-training and

Table 3. Correlation coefficient for classification errors and labeling errors

Dataset	naive Bayes	Decision tree
anneal	0.889	0.855
balance-scale	0.904	0.861
breast-c	0.878	0.899
credit-a	0.846	0.903
colic	0.887	0.899
diabetes	0.854	0.697
hypothyroid	0.890	0.824
letter	0.718	0.701
primary-tumor	0.881	0.902
segment	0.908	0.625
sick	0.914	0.847
soybean	0.617	0.728
splice	0.893	0.866
vehicle	0.702	0.871
vote	0.872	0.853
vowel	0.911	0.891

cotrain-AS in the experiments discussed in Section 4. Due to space limitation, the detailed labeling error rates are omitted. Instead we present the correlation between the classification errors and labeling errors.

As mentioned earlier, for each view split of a given dataset we run the two co-training methods $A1$ and $A2$; i.e. original co-training and cotraining-AS, 10*5 times. This results in 50 pairs of expanded labeled sets and the corresponding 50 pairs of modeling classifiers. For each expanded labeled set and modeling classifier we can calculate the labeling error and the testing error. At this point there are 50 pairs of labeling errors and 50 pairs of testing errors representing the one to one performance comparison of the methods $A1$ and $A2$ for each view split. We then run two separate t-tests to compare the significance of the difference in labeling errors and the significance of the difference in testing errors of $A1$ and $A2$. We use the value 1 to show that $A2$ is significantly better than $A1$ for a given performance measure and 0, otherwise. Consequently, for each split we know if $A2$ had a significantly better labeling error compared to $A1$ and similarly if it had a significantly better testing (or classification) error. We can obtain an array of the above value pairs for each dataset. The size of this array is equivalent to the number of splits generated for the dataset. We then calculate the correlation between the pairs in the array by using spearman's footrule ρ [1]. This shows the correlation between the labeling performance comparison of the cotrain-AS versus the original co-training.and the corresponding classification performance comparison for each dataset. As the results in Table 3 show, for all datasets and learning algorithms the spearman correlation is greater than 0.6. For most datasets the correlation is much higher.

These results indicate that in the case of our experiments there is a strong correlation between better modeling predictions and better labeling performance. The better classification error rates obtained by cotrain-AS is due to its smaller labeling error rates.

[1] For two given lists of $a_1, a_2, \cdots, a_n$ and $b_1, b_2, \cdots, b_n$, if a_i, b_i is 0 or 1, the spearman's footrule $\rho = 1 - \sum_{i=1}^{n}(a_i - b_i)^2/n$.

6 Conclusions and Future Work

In this paper we propose a novel agreement-based sampling method to improve the performance of two-view co-training algorithm. The basic idea of this sampling method is to select unlabeled instances that the two view classifiers agree the most on their label. The criteria used in this sampling method is opposite to that of co-testing. Our experiments show that this new sampling method can indeed make a significant performance improvements over the original co-training. Finally we empirically explored why agreement-based sampling method works better. For our future work, we intend to apply the new sampling method to other semi-supervised learning methods such as co-EM.

Acknowledgements

This research has been supported by the Natural Sciences and Engineering Research Council of Canada.

References

1. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.M.: Text classification from labeled and unlabeled documents using EM. Machine Learning 39(2/3), 103–134 (2000)
2. Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts. In: Proc. 18th International Conf. on Machine Learning, pp. 19–26. Morgan Kaufmann, San Francisco (2001)
3. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: COLT: Proceedings of the Workshop on Computational Learning Theory, pp. 92–100. Morgan Kaufmann Publishers, San Francisco (1998)
4. Balcan, N., Blum, A., Yang, K.: Co-training and expansion: Towards bridging theory and practice. In: Proceedings of the Eighteenth Annual Conference on Neural Information Processing Systems, NIPS 2004 (2004)
5. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: CIKM, pp. 86–93 (2000)
6. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. In: Proceedings of 17th International Conference on Machine Learning, ICML 2000, pp. 999–1006. Morgan Kaufmann Publishers, San Francisco (2000)
7. Freund, Y., Seung, H.S., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. Machine Learning 28(2-3), 133–168 (1997)
8. Wang, W., Zhou, Z.: On multi-view active learning and the combination with semi-supervised learning. In: Proceedings of the 25th International Conference on Machine Learning, ICML 2008 (2008)
9. Muslea, I., Minton, S., Knoblock, C.A.: Selective sampling with redundant views. In: AAAI/IAAI, pp. 621–626 (2000)
10. Blake, C., Merz, C.: UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
11. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, San Francisco (2000)

Experienced Physicians and Automatic Generation of Decision Rules from Clinical Data

William Klement[1], Szymon Wilk[1,3], Martin Michalowski[2], and Ken Farion[4]

[1] MET Research Group, University of Ottawa, Canada
[2] Adventium Labs, Minneapolis MN 55401, USA
[3] Institute of Computing Science, Poznan University of Technology, Poland
[4] Division of Emergency Medicine, Children's Hospital of Eastern Ontario, Canada

Abstract. Clinical Decision Support Systems embed data-driven decision models designed to represent clinical acumen of an experienced physician. We argue that eliminating physicians' diagnostic biases from data improves the overall quality of concepts, which we represent as decision rules. Experiments conducted on prospectively collected clinical data show that analyzing this filtered data produces rules with better coverage, certainty and confirmation. Cross-validation testing shows improvement in classification performance.

1 Introduction

The functionality of clinical decision support systems (CDSS) relies on their embedded decision models that represent knowledge acquired from either data or domain experts. Data-driven models are created to acquire knowledge by deriving relationships between data features and decision outcome. In medical domains, while the data describes patients with a clinical condition, the decision indicates a diagnostic outcome. These diagnostic decisions are normally transcribed from patient charts and are verified for correctness, e.g. by a follow-up. Traditionally, a verified outcome forms the gold standard (GS) used in the analysis of the decision models. In this paper, we assume a data-driven approach but argue that the reliance in the analysis on the GS may skew the resulting relationships. Our goal is to show that the use of cases, where the experienced physician (EP) makes correct diagnoses according to the verified patient outcome (GS), results in better decision models. To acquire unbiased clinical knowledge, we argue that it is essential to eliminate records where EP decisions do not match GS prior to constructing a decision model from data.

In clinical domains, patient records represent instances of a relationship, between attribute values, describing the status of patients' health and diagnostic decisions made by the physician. Several studies establish that EP makes good clinical decisions, particularly when dealing with critical cases. However, EPs often err in favor of caution and tend to over-diagnose patients who are relatively healthy. Therefore, their decisions are characterized by high sensitivity and lower specificity. From a decision making perspective, low specificity if not controlled introduces noise in the data. This can be evident by discrepancies between decisions made by EP and those established by the GS.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 207–216, 2010.

Our approach in identifying unbiased clinical knowledge in the data represents a departure from the established common practice where relationships are sought between patient attribute values and class labels (diagnoses) derived from the GS. These decision models are constructed from data collected by EPs but the diagnostic outcome is established from the GS. This practice is skewed because resulting models represent knowledge distorted by erroneous characteristics associated with the EP's decision models. As a remedy, we propose to eliminate EPs' biases from the data. Cases for which GS and EP decisions match constitute "correct decisions", are particularly valuable, and provide "sound" clinical knowledge. Classical methods construct their decision models from all available records regardless of the correctness of their EP decisions. We argue that such model construction may introduce bias in discovered knowledge, and we propose to focus the analysis on cases with correct EP decisions only.

This paper aims to demonstrate the value of these "correct" cases in the context of knowledge acquisition from patient data. We apply our analysis to two clinical domains, the diagnosis of pediatric abdominal pain (AP) and pediatric asthma exacerbation (AE). To this extend, our experiment shows that filtering the data, based on the match between EP decisions and the GS, produces crisp knowledge. Naturally, we must clarify what form of knowledge we extract, and how we measure its quality. To represent knowledge, we exploit concepts of rough set theory that represent it by a set of minimal decision rules [9,8]. We apply the MODLEM algorithm [11,12] to generate these decision rules. For evaluation, we employ several metrics to assess the quality of rules based on their structure and their performance on data. They include; the number of generated decision rules, length, coverage, certainty, and confirmation and are reviewed in [4].

The paper is organized as follows. Section 2 discusses clinical domains used for our experiments, and section 3 describes basic principles behind rough sets, the MODLEM algorithm used for rule generation, and rule evaluation metrics. Experimental design and results are discussed in section 4, and conclusions can be found in section 5.

2 The Role of EP's Expertise

Clinical decision-making is a complex process influenced by a verity of uncertain factors and should include the integration of clinical expertise [5]. Information technology solutions have been commonly considered as decision support mechanisms to provide clinicians with appropriate information while making clinical decisions. Such solutions include Clinical Decision Support Systems (CDSS) which have increasingly captured the attention of the medical community in recent years. A CDSS is defined as *"any program designed to help healthcare professionals make clinical decisions"* [7]. CDSS provide information in three widely accepted categories including; information management, focusing attention on specific health events, and patient-specific recommendations. The latter helps physicians make two types of decisions. The first is diagnostic where the focus is set on the patients underlying health condition, and the second type

deals with patient management with regards to what treatment plan is most appropriate for the patient. Despite the varying techniques for extracting expert knowledge, patient-specific CDSS decision models almost always reflect clinician expertise with the embedding of the knowledge of the 'best practice'. Obviously, the knowledge of an EP is vast. It has been documented that a physician is considered an expert after 10 years of training [10] who is able to summarize information and to develop a complex network of knowledge [1].

The focus of our research is to enhance and to support the process of acquiring expert knowledge from patient-specific data, we are able to capture it by filtering the data according to the EP's correct practice. Such knowledge can be used in the construction of decision models ready for integration into CDSS. To this extent, we wish to exploit decisions made by the EPs which reflect their clinical acumen. When comparing their decisions to the verified patient outcomes, the GS, physician's diagnostic biases become clear. Investigating the circumstances of these biases is a difficult task as they can be caused by multitude of factors including differing expertise of physicians [5]. To account for the diagnostic biases, we propose to rely only on correct EP decisions, and therefore we consider data for those patients where EP decisions match the GS.

AE data was collected as a part of a study conducted at the Children Hospital of Eastern Ontario (CHEO), and it includes patients who visited the hospital emergency department (ED) experiencing asthma exacerbation. In the ED, a patient is repeatedly evaluated by multiple clinicians at variable time intervals. This information is documented and collected prospectively for each patient. The resulting patient records contain information about history, nursing, physician triage assessment, and reassessment information collected approximately 2 hours after triage. Records in the AE data set are assigned to one of two outcome classes: mild or other severity of exacerbation. The verified severity of exacerbation is used as a GS. The dynamic nature of asthma exacerbation and the collection of assessments over time would lend itself naturally to a temporal representation for analysis of data. However, inconsistencies in data recording meant it was not possible to incorporate a temporal aspect into the analysis.

The AP data is also collected in the ED of CHEO and includes patients who have serious conditions, mostly appendicitis, who require surgery. However, most records describe benign causes. Before a cause can be found, symptoms often resolve without complications so that a definitive diagnosis is not possible during the ED visit. Therefore, choosing the correct triage plan is an important proxy [2] and we use it as a class label. This triage plan may involve discharging the patient, continuing observation, or asking for a specialty consultations. In our AP data, these outcomes are transformed into binary values indicating whether a patient requires specialist consultation. As with the AE data, a GS was established from verified patient outcomes.

3 Generating and Assessing Decision Rules

Based on the mathematical model of rough set theory [9], we generate a minimal set of decision rules using the MODLEM algorithm described in [13]. These

Table 1. The contingency table of a decision rule "*if X then Y*"

	Y	$\overline{Y}$	
X	a	b	r_1
$\overline{X}$	c	d	r_2
	c_1	c_2	

decision rules represent knowledge extracted from data, and we assess their quality using several measures presented in [4] and discussed later in this section. Our objective is to show that analysis conducted on cases where EP decisions are correct result in better rules, and therefore higher quality knowledge, than those performed on data with the GS being the class label.

Rough set analysis rely on an information table which contains data points (examples) described by a finite set of attributes. Such table becomes a decision table when we are able to identify a set of condition attributes C and relate them to a set of decisions D. From a decision table, we can induce decision rules of the form "*if* $\cdots$, *then* $\cdots$". We now describe an intuitive illustration appropriate for our domains which appears in [4]. Given a data sample describing patients and their diseases, the set of signs and symptoms $S = \{s_1, \cdots, s_n\}$ contains their condition attributes and a set of diseases $D = \{d_1, \cdots, d_m\}$ as their decision attributes. A decision rule has the form "*if symptoms* $s_i, s_j, \cdots, s_w$ *appear, then there is disease* d_v" with $s_i, s_j, \cdots, s_w \in S$ and $d_v \in D$.

The MODLEM algorithm [13] is designed to induce such decision rules based on the idea of *sequential covering* to generate a *minimal set* of decision rules for every decision concept. A decision concept may be the decision class or a rough approximation of the decision class in the presence of inconsistent examples. The objective of this minimal set of decision rules is to cover all the positive examples in the positive class without covering any negative examples. A benefit of using MODLEM lies in its ability to process numerical attributes without discretization. In addition, this algorithm has been shown to produce effective and efficient single classification models [12]. The process of generating the set of minimal decision rules is iterative. For every decision class, the MODLEM algorithm repeatedly builds decision rules to cover examples in that class, then, it removes examples covered by this rule from the data. This process continues until all examples in the class are covered, and the "*best*" rules are selected according to a chosen criterion, e.g. class entropy. For more detailed description of the MODLEM algorithm, we refer to [13].

Comparing the characteristics and performance of decision rules has long been a subject of research. In this paper, we utilize several classical rule evaluation measures including the rule confirmation measure, all of which, are reviewed in details in [4]. We also illustrate calculations and present interpretations of metrics used in our experiments. For simplicity, let "*if X then Y*" be a decision rule where X is a subset of conditions and Y is a decision class. Applying this decision rule to data produces entries that populate the contingency Table 1. Essentially, this table depicts counts of examples that are covered by all possible combinations of either side of the decision rule. In the data, while there are a

examples that satisfy the set of conditions X whose decision is Y, examples that fail conditions X and their decision is $\overline{Y}$[1] are depicted by d. Similarly, b is the number of examples that meet conditions X but their decision is $\overline{Y}$, and finally, c is the count of examples that fail conditions X but their decision is Y. While c_1 and c_2 represent the column summations, r_1 and r_2 are the row summations. Their interpretations are simple; the column summations show the number of examples whose decision class is Y or $\overline{Y}$ respectively, and the row summations indicate the number of examples that satisfy X or $\overline{X}$ also respectively.

Rule evaluation measures are well established and fall into two main categories; the first involves assessing the structure of the rule, and the second relates to their performance. While the former is based primarily on the length of the rule, the latter includes rule *coverage, certainty,* and *confirmation.* These measures are discussed in [4], and we compute their values for each rule by counting entries in Table 1, then, we substitute their values in equations 1, 2, and 3.

$$coverage(X, Y) = \frac{a}{c_1} \tag{1}$$

$$certainty(X, Y) = \frac{a}{r_1} \tag{2}$$

$$confirmation(X, Y) = \frac{ad - bc}{ad + bc + 2ac} \tag{3}$$

While higher coverage values depict the strength of the rule, a high value of rule certainty indicates higher confidence. In addition, several measures have been proposed to asses rule confirmation. However, we use the $f()$ measure presented in [4], which quantifies the degree to which the observed evidence supports for, or against, a given hypothesis. The findings in [4] show its effectiveness.

To assess the quality of knowledge extracted from data in the form of decision rules, we first consider characteristics describing this set. Such characteristics include the number of decision rules, the number of conditions, and the average length of a rule with the associated standard deviation. Such characteristics reflect the complexity of the concept in the sense that, while complex concepts may have more rules, these rules tend to be longer because they include more conditions. This is consistent with simpler, more effective rules having fewer conditions, thus they are shorter in length and there are fewer of them. From a performance assessment perspective, concepts which are described by fewer rules, show greater coverage, produce higher levels of certainty, and have better confirmation are considered of better quality.

4 Experimental Design and Results

The objective of our experiment is to demonstrate that the quality of knowledge, acquired from clinical data, is improved by including only those patient cases for which the EP makes correct decisions as indicated by the verified patient

[1] $\overline{X}$ is $\neg X$, the complement of X.

Table 2. Characteristics of the clinical data sets

		GS Outcome			EP Decisions	
Data	Examples	Positives	Negatives	Ratio	Positives	Negatives
AE_{all}	240	131	109	55%	136	72
AE_{corr}	140	90	50	64%	90	50
AP_{all}	457	48	409	11%	55	402
AP_{corr}	422	34	388	8%	34	388

outcome, the GS. Furthermore, the experiment shows that the performance of the associated classification model can also be improved using the proposed filtering of patient records. We conduct our experiment in two phases. In the first phase, the knowledge acquisition phase, we generate a minimal set of decision rules and record their characteristics for analysis. The second phase uses 10-fold cross-validation runs repeated 5 times to evaluate the performance of the decision models after filtering the training data.

Data sets used in the experiment and their characteristics are listed in Table 2. The subscript *all* for AP and AE indicates that all patient records are analyzed (non-filtered). Similarly, the subscript *corr* indicates filtered data sets, where class labels correspond to EP decisions that match GS after eliminating mismatching records. Examining Table 2 reveals that while the class distribution of the AE data is almost balanced, it is not so for the AP data where the ratio of positive examples is less than 11%. Therefore, we use an under-sampling technique to balance it by randomly selecting, without replacement, an equal number of examples in both classes to retain the complete set of positive examples. A data set after under-sampling is labeled APS. This set is processed in two settings; the first consists of all examples that are randomly under-sampled and is indicated by APS_{all}. The second is denoted by APS_{corr} and includes an under-sample set of cases for which EP decision are correct, i.e. we under-sample data resulting from EP-based filtering.

A pairwise comparison of the number of examples on the *all* rows to those on the *corr* rows of Table 2, respectively, shows that EP makes more correct decisions in the AP domain than in the AE. In both domains, however, EPs over-diagnose patients as having a positive condition more often than indicated by the GS. This is seen by comparing the number of positive EP decisions for both *all* and *corr* rows in both domains.

4.1 Discussion

We begin discussion by considering characteristics recorded for each concept extracted from the data. A concept is represented by a set of decision rules for which we show the number of conditions, the number of rules, and the average rule length with its standard deviation for both and for individual classes. In this order, these values are shown in the columns of Table 3. Examining these

Table 3. Characteristics of resulting concepts

	Both Classes			+ Class			– Class		
Data	Cond.	Rules	Ave. Length	Cond.	Rules	Ave. Length	Cond.	Rules	Ave. Length
AE_{all}	199	50	3.98 ±1.19	92	23	4.00 ±1.00	107	27	3.96 ±1.34
AE_{corr}	13	6	2.17 ±0.98	5	3	1.67 ±1.16	8	3	2.67 ±0.58
AP_{all}	163	38	4.29 ±1.51	99	21	4.71 ±1.65	64	17	3.77 ±1.15
AP_{corr}	83	24	3.46 ±1.14	37	10	3.70 ±1.25	46	14	3.29 ±1.07
APS_{all}	91	25	3.64 ±0.95	51	14	3.64 ±1.08	40	11	3.64 ±0.81
APS_{corr}	19	8	2.38 ±0.52	9	4	2.25 ±0.50	10	4	2.50 ±0.58

characteristics on individual classes shows that the rules are almost evenly distributed on the positive and on the negative class. Individually, they have an almost equal number of conditions, number of rules, and average rule length.

An important observation points to the fact that values recorded for data sets with *corr* index are smaller than those recorded for the non-filtered data sets. For all three data sets, AE, AP, and APS, the set of decision rules generated form data containing correct EP decisions results in fewer conditions, fewer rules, and shorter average rule length with a lower standard deviation. This observation remains consistent whether we consider the set of decision rules describing both classes or individual classes. This suggests that using data with correct EP decisions produces concepts with less complexity and possibly ones with higher quality.

Results of evaluating the performance of these decision rules are shown in Table 4. The pairwise comparison of average values for each performance measure reveals that they remain unchanged or increase when rules are generated form data containing correct EP decisions. With such filtering, the average rule coverage increases dramatically for the AE data. For the AP data, the use of under-sampling allows the average rule coverage a higher increase than that obtained without under-sampling. This is seen when we compare the difference in average coverage values of APS_{all} and APS_{corr} against that for AP_{all} and AP_{corr}. This is attributed to the imbalanced class distribution of the AP data.

The average certainty on the AE domain achieves its maximum value of 1 and remains unaffected by the elimination of cases with incorrect EP decisions, see average certainty values for AE_{all} and AE_{corr}. A similar statement can be made for the average rule confirmation in the AE domain in the same table. On the AP data and regardless of using under-sampling, the average rule certainty and the average confirmation are both improved by our filtering, their average values for AP_{corr} and APS_{corr} are increased over AP_{all} and APS_{all} respectively. Such results lead to the conclusion that generating decision rules using examples with correct EP decisions enhances the coverage, the certainty and the confirmation of the rules. However, the standard deviation increases for the rule coverage measure in both domains. This is not surprising because data with cases of correct EP decisions are always smaller than their respective original sets, their sizes

Table 4. Assessing the performance of resulting decision rules

Measure	Data	Both classes	+ Class	– Class
Coverage	AE_{all}	0.055 ±0.044	0.062 ±0.052	0.048 ±0.036
	AE_{corr}	0.534 ±0.311	0.507 ±0.362	0.560 ±0.330
	AP_{all}	0.117 ±0.153	0.068 ±0.051	0.176 ±0.209
	AP_{corr}	0.158 ±0.204	0.176 ±0.130	0.145 ±0.248
	APS_{all}	0.137 ±0.132	0.131 ±0.128	0.145 ±0.143
	APS_{corr}	0.423 ±0.209	0.338 ±0.098	0.507 ±0.271
Certainty	AE_{all}	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000
	AE_{corr}	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000
	AP_{all}	0.955 ±0.157	0.929 ±0.208	0.988 ±0.030
	AP_{corr}	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000
	APS_{all}	0.969 ±0.104	0.964 ±0.134	0.974 ±0.053
	APS_{corr}	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000
Confirmation	AE_{all}	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000
	AE_{corr}	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000
	AP_{all}	0.914 ±0.219	0.953 ±0.165	0.867 ±0.269
	AP_{corr}	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000
	APS_{all}	0.937 ±0.210	0.928 ±0.270	0.949 ±0.104
	APS_{corr}	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000

Table 5. Classification performance of resulting decision rules

Data	Sensitivity	Specificity	Accuracy	Geometric Mean†
AE_{all}	0.7024 ± 0.1510	0.5930 ± 0.1634	0.7158 ± 0.0865	0.6341 ± 0.1059
AE_{corr}	0.7908 ± 0.1268	0.5070 ± 0.1309	0.6825 ± 0.0803	0.6243 ± 0.1029
AP_{all}	0.4930 ± 0.2231	0.9640 ± 0.0264	0.9252 ± 0.0322	0.6619 ± 0.1952
AP_{corr}	0.5877 ± 0.2236	0.9526 ± 0.0368	0.9305 ± 0.0310	0.7284 ± 0.1696
APS_{all}	0.7913 ± 0.2246	0.7856 ± 0.0525	0.8127 ± 0.0406	0.7777 ± 0.1296
APS_{corr}	0.7470 ± 0.2353	0.8560 ± 0.0642	0.8559 ± 0.0509	0.7909 ± 0.1396

† Entries are averaged over 5 runs of 10-fold cross validation.

are shown in Table 2, but the average values of rule certainty and confirmation achieve their maximum of 1 with a standard deviation of 0. Clearly, our filtering helps the model achieve high certainty and strong confirmation.

Our final results are given in Table 5, which shows the average sensitivities, specificities, accuracies, and the geometric means[2] of sensitivities and specificities resulting from testing the classification performance of the decision rules. The testing method relies on the 10-fold cross-validation repeated 5 times for

[2] The geometric mean measure is used for imbalanced class distributions [6].

which the above averages are recorded. Results for AE data show a clear gain in sensitivity with a loss in specificity, accuracy and geometric mean. Clearly, examples with correct EP decisions set their focus on the positive class.

For the AP data, the classification performance improves in principle. Balancing the AP_{all} data by under-sampling, to produce APS data, improves the classification performance with the exception of specificity (0.96 to 0.79). However, under-sampling the EP-filtered data, which produces the APS_{corr} data, recovers the specificity (0.79 to 0.86). Consequently, combining our filtering approach with sampling techniques must be done with care.

Given that the positive class represents an acute medical condition, the need for a specialist consult for AP and the pronounced asthma exacerbation for AE, the resulting sensitivity values show that our decision rules produce a reasonable classification performance. The latter can be improved by conducting a comprehensive experiment to select an appropriate data mining method.

5 Conclusions

Data-driven knowledge acquisition techniques used to extract knowledge describing EP decision making is a complex process which involves various factors. This paper shows that capturing knowledge in the form of decision rules from examples of correct EP decisions results in a better description of knowledge. This is exemplified by reduced complexity characterized with fewer, shorter rules. The performance of these rules is also enhanced with better coverage, higher certainty, and increased confirmation. With enhanced quality of knowledge, the classification performance is shown to improve with increased sensitivity.

Acknowledgment

The authors acknowledge the support of the Natural Sciences and Engineering Council of Canada. The second author also acknowledges support of the Polish Ministry of Science and Higher Education (grant N N519 314435).

References

1. Arocha, J.F., Wang, D., Patel, V.: Identifying reasoning strategies in medical decision making: A methodological guide. Biomedical Informatics 38(2), 154–171 (2005)
2. Farion, K., Michalowski, W., Rubin, S., Wilk, S., Correll, R., Gaboury, I.: Prospective evaluation of the MET-AP system providing triage plans for acute pediatric abdominal pain. Int. Journal of Medical Informatics 77(3), 208–218 (2008)
3. Farion, K., Michalowski, W., Wilk, S., O'Sullivan, D., Matwin, S.: A tree-based decision model to support prediction of the severity of asthma exacerbations in children. Journal of Medical Systems (2009) (forthcoming)
4. Greco, S., Pawlak, Z., Slowinski, R.: Can Bayesian confirmation measures be useful for rough set decision rules? Engineering App. of AI 17, 345–361 (2004)

5. Hine, M.J., Farion, K., Michalowski, W., Wilk, S.: Decision Making By Emergency Room Physicians And Residents: Implications for the Design of Clinical Decision Support Systems. International Journal of Healthcare Information Systems and Informatics 4(2), 17–35 (2009)
6. Kubat, M., Holte, R.C., Matwin, S.: Machine Learning for the Detection of Oil Spills in Satellite Radar Images. Machine Learning 30(2-3), 195–215 (1998)
7. Musen, M.A., Sahar, Y., Shortliffe, E.H.: Clinical decision support systems. In: Medical Informatics, Computer applications in healthcare and biomedicine, ($n^n d$ edn.), pp. 574–609. Springer, New York
8. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1998)
9. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Boston (1991)
10. Prietula, M.J., Simon, H.: The experts in your midst. Harvard Business Review 67(1), 120–124 (1989)
11. Stefanowski, J.: The rough set based rule induction technique for classification problems. In: Proceedings of Sixth European Conference on Intelligent Techniques and Soft Computing, EUFIT 1998, pp. 109–113 (1998)
12. Stefanowski, J.: Algorithms of rule induction for knowledge discovery. Habilitation Thesis published as Series Rozprawy, vol. 361. Poznan University of Technology Press, Poznan (2001) (in Polish)
13. Stefanowski, J.: On combined classifiers, rule induction and rough sets. In: Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J.W., Orłowska, E., Polkowski, L. (eds.) Transactions on Rough Sets VI. LNCS, vol. 4374, pp. 329–350. Springer, Heidelberg (2007)
14. Wilk, S., Slowinski, R., Michalowski, W., Greco, S.: Supporting triage of children with abdominal pain in the emergency room. European Journal of Operational Research 160, 696–709 (2005)

Gene-Pair Representation and Incorporation of GO-based Semantic Similarity into Classification of Gene Expression Data

Torsten Schön[1,2], Alexey Tsymbal[2], and Martin Huber[2]

[1] Hochschule Weihenstephan-Triesdorf, Freising, Germany
info@torsten-schoen.de
[2] Corporate Technology Div., Siemens AG, Erlangen, Germany
alexey.tsymbal@siemens.com, martin.huber@siemens.com

Abstract. To emphasize gene interactions in the classification algorithms, a new representation is proposed, comprising gene-pairs and not single genes. Each pair is represented by L_1 difference in the corresponding expression values. The novel representation is evaluated on benchmark datasets and is shown to often increase classification accuracy for genetic datasets. Exploiting the gene-pair representation and the Gene Ontology (GO), the semantic similarity of gene pairs can be incorporated to pre-select pairs with a high similarity value. The GO-based feature selection approach is compared to the plain data driven selection and is shown to often increase classification accuracy.

1 Introduction

Since the human genome has been sequenced in 2003 [13], biological research is becoming more interested in the genetic cause of diseases. Microbiologists try to find responsible genes for the disease under study by analyzing the expression values of genes. To extract useful information from the genetic experiments and to get a better understanding of the disease, usually a computational analysis is performed [22].

In addition to the commonly used plain *k*-Nearest Neighbour (*k*-NN) algorithm, there are other more sophisticated approaches that use a distance function for classification and are suitable for processing gene expression data. Recently, a growing body of work has addressed the problem of supervised or semi-supervised learning of customized distance functions [12]. In particular, two different approaches, learning from equivalence constraints and the intrinsic Random Forest similarity have been recently introduced and shown to perform well in particular with image data [5,14,26]. Many characteristics of gene expression data are similar to those of image data. Both imaging and genetic data usually have a big number of features where many of them are redundant, irrelevant and noisy.

Another often used possibility to improve the analysis of genetic data is to exploit available external biological knowledge [27,16,21,7]. The Gene Ontology (GO) [3] is a valuable source of biological knowledge that can be incorporated into the process of classification or clustering of genetic data. In addition, the content of GO is

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 217–226, 2010.

periodically refined and constantly growing, becoming a more complete and reliable source of knowledge. The number of entries in the GO, for example, has increased from 27,867 in 2009 to 30,716 in 2010, which is a more than 10% increase.

In this paper, a novel data representation for learning from gene expression data is introduced, which is aimed at emphasizing *gene-gene* interactions in learning. With this representation, the data simply comprise differences in the expression values of gene pairs and not the expression values themselves. An important benefit of this representation, except the better sensitivity to gene interactions, is the opportunity to incorporate external knowledge in the form of *semantic similarity*.

This paper is organized as follows. In Section 2 we review related work and the datasets used; in Section 3 we introduce the gene-pair representation and the application of the semantic similarity for guiding feature selection. Section 4 presents our empirical study and Section 5 concludes the paper with a summary and directions for future work.

2 Material

2.1 Related Work

Much work has been done in the area of machine learning over the last few decades and as bioinformatics is gaining more attention, different techniques are being developed and are applied to process genetic data [17]. In addition to the usually small sample of cases (conditions) making it difficult to extract statistically valid patterns from the data, the large amount of gene expression values clearly complicates the learning too. For that reason, a feature selection step is normally conducted before classification to determine useful discriminative genes and eliminate redundancy in the dataset. Feature selection has thus become under active study in bioinformatics and is currently one of the clear focuses in bioinformatics research [24].

Based on the controlled vocabulary of the GO, two genes can be semantically associated and further a similarity can be calculated based on their annotations in the GO. For example, Sevilla et al. [25] have successfully applied semantic similarity which is well known in the field of lexical taxonomies, AI and psychology to the GO by calculating the information content of each term in the ontology based on several methods including those of Resnik [23], Jiang and Conrath [15] or Lin [18]. Sevilla et al. computed correlation coefficients to compare the physical inter-gene similarity with the GO semantic similarity. The experiments demonstrated a benefit for the similarity measure of Resnik [23], which resulted in higher correlations.

Later Wang and Azuaje [28] have integrated the similarity information from the GO into the clustering of gene expression data. They have shown that this method not only ensures competitive results in terms of clustering accuracy, but also has the ability to detect new biological dependencies.

Another approach to include the knowledge from the GO into machine learning is to use it for feature selection. Thus, Qi and Tang [21] have introduced a novel method to select genes not only by their individual discriminative power, but also by integrating the GO annotations. The algorithm corrects invalid information by ranking the genes based on their GO annotations and was shown to boost accuracy. Chen and

Tang [7] further investigated this idea, suggesting a novel approach to aggregate semantic similarities and integrated it into traditional redundancy evaluation for feature selection. This resulted in higher or comparable classification accuracies with using less features compared to plain feature selection. The approach we introduce here is inspired by these attempts but is different in that the semantic similarity is applied directly to the corresponding gene pair in the novel gene-pair representation.

2.2 Benchmark Datasets

For evaluation, ten benchmark datasets have been used. The datasets are associated with different clinical problems and have been received from various sources. *Colon* [2], *Embrional Tumours* [20], *Leukemia* [8] and *Lymphoma* [1] datasets have been obtained from the Bioinformatics Research Group, Spain (*http://www.upo.es/eps/aguilar/datasets.html*), *Arcene* [10] is available at the UCI repository [6], *Lupus* [4] from The Human-Computer Interaction Lab at the University of Maryland, USA, *Breast Cancer* [27] from the University of Minnesota and *Lung Cancer* [9] from the Division of Thoracic Surgery at Brigham and Women's Hospital, USA. The *Mesh* dataset was generated from cardiac aortic valve images, see [14]. The last dataset, *HeC Brain Tumours* is obtained from the hospitals participating in the EU FP6 Health-e-Child consortium, see *www.health-e-child.org*. The datasets are public and often used in research studies, except the last two which are not publicly available. All the datasets represent binary classification problems and are different in the number of features and cases, although one important commonality is that the number of features (from 2,000 in Colon to 10,000 in Arcene) significantly exceeds the number of cases (from 45 in Lymphoma to 72 in Leukemia, to 200 in Arcene). Moreover, many features are redundant, irrelevant and/or noisy, which is typical for biomedical and in particular for gene expression data.

3 Methods

3.1 The Gene-Pair Representation and Experimental Setting

Genetic datasets similar to those considered in Section 2.3 normally contain gene expression values, where each feature is the expression of a single gene. In biology, however, the influence of a gene on a certain disease often depends not only on its own expression, but also on the expression of some other genes, interacting with it. Thus, at a certain disease, the higher expression of gene A might only be influencing the etiopathology if another gene B is over- or under-expressed, too. By training a classifier with microarray data with the usual single gene representation, these dependencies are often neglected or are at least more difficult to grasp and thus require strong adaptive learners. To better consider these co-operations, the data can be transformed into another representation. A normalization of each gene's expression with respect to the other genes is needed.

Another motivation for the new representation is the incorporation of the semantic similarity into classification. The GO provides similarity measures between two genes A and B while most learning algorithms when the usual representation is used, consider differences between patients with the given gene expression values. The GO

provides information about gene-gene interactions and the classifiers normally use patient-patient relations for classification. There is no obvious way how to incorporate the semantic similarity between two genes to improve the classification of patients, when the plain representation is used.

For these reasons, the original datasets can be transformed into a new representation with gene pairs instead of single genes. The simplest yet most robust solution also accepted by us here is to use L_1 difference in the corresponding gene expression values. First, all possible pairs of single genes available in the dataset are generated. Training a classification model may not finish in a limited period of time if all the pairs are used. Moreover, most of these pairs will be useless for classification. Therefore, feature selection has to be done to select the discriminative pairs in advance.

Our preliminary experiments with this representation have demonstrated that the classifiers usually perform well already with 100 pair-features, if a proper feature selector is used. Selecting more gene pairs, in general, does not increase accuracy, and requires considerable time.

Arcene and *Mesh* are non-genetic, but they contain enough features to get comparable results and were used as non-genetic reference datasets. As the base for our experiments, reduced datasets have been used, containing not more than 200 single gene features (if necessary, pre-selection was conducted with *GainRatio* feature filter). The final number of gene pairs to select was set to 100 with 400 gene-pairs preselected by *ReliefF*. *CFS,* Correlation-based Feature Selection for subset evaluation, together with the greedy stepwise search with forward inclusion available in Weka [11] was used for the final selection of 100 pairs from the set of 400. For datasets containing more than 90 cases, 30 iterations of 10-fold cross validation has always been used in our experiments here and further and leave-one-out cross validation for the others. Each representation was evaluated on all datasets with four classifiers, plain *k-Nearest-Neighbour (kNN)* classification, *Random Forest (RF), kNN with learning from equivalence constraints (EC)* and *kNN with intrinsic Random Forest Similarity (iRF).* To measure the robustness of the gene-pair representation to noise, the same tests have been conducted also with 10% and 20% of class noise.

3.2 Integration of GO Semantic Similarity and Experimental Setting

With the novel representation of features as the difference in the expression values of a pair of two genes, it becomes possible to incorporate the external biological knowledge into classification. There are several reasons that motivate us to use semantic similarity to guide the classification. First, two genes that are known to interact (that is the genes whose semantic similarity is expected to be relatively high, annotations of which are closer to each other in the GO so that the genes are more functionally related), might be more useful for classification than two genes that are not associated with each other, because difference in their expression values can be more clinically significant than difference in the expression values of two genes which are not related. Second, the usually big number of features in gene expression data makes it difficult for common feature selection methods not to ignore some important features. The support by the semantic similarity might guide the selection process and help to consider important pairs that would otherwise be neglected.

A simple approach to use the semantic similarity to support classification is to pre-select pair-features based on the corresponding GO similarity values such as GIC [19]; a certain number of pairs with the greater semantic similarities according to the GO are initially selected. In our experimental setting, first, 400 pairs with the greater GO similarities are selected, followed by the CFS feature selection to select the 100 most discriminative gene-pairs out of 400. To identify the best matching semantic similarity calculation technique for genetic data, a set of similarity calculation methods have been tested.

In a set of preliminary experiments with different semantic similarity measures (the results of these experiments are not included here due to the space restrictions), the GIC, Graph Information Content [19], semantic similarity was shown to demonstrate the highest correlation with the physical gene expression similarity for two genes and be the best guide for feature selection (in particular, the GIC values in the group of "best" most discriminative pairs was shown to be 71% higher than for all the pairs, while this increase was much lower or even absent for the other techniques). Thus, GIC was used in our experiments with the GO similarity.

4 Experimental Results

4.1 Gene-Pair Representation

The gene-pair representation was compared with the plain representation and evaluated with four classifiers. The following configurations have been used for the experiments.

- 400 preselected pairs by *ReliefF*.
- 100 pairs selected by *CFS* and used for training the models.
- Four classifiers as follows:
  RF: Random Forest with 25 trees.
  kNN: *k*NN with $k = 7$ and case weighting inversely proportional to distance.
  EC: *k*NN with learning from equivalence constraints with RF in difference space.
  iRF: *k*NN with intrinsic Random Forest similarity, 25 trees.

It was shown that k=7 and case weighting inversely proportional to distance is the most robust parameter choice for *kNN* in our preliminary tests, which thus used in all nearest neighbour classifiers in our experiments. The experiments have been implemented and conducted based on the Weka machine learning library in Java [11] and default parameter values were always used for classifiers and feature selectors unless otherwise stated here.

Main experimental results are presented in Table 1, where the two rows of each dataset correspond to the original representation and the gene-pair representation, respectively. Each column includes results for one learning algorithm and the average over all four classifiers is presented in the last column. First, results for the genetic datasets and averages over them and then results for the non-genetic datasets and the averages are presented.

Table 1. Classification accuracies for the two representations

Data Set	*RF*	*kNN*	*EC*	*iRF*	*Average*
Breast Cancer	74.11	81.22	75.56	72.44	75.83
	73.11	78.56	77.00	72.44	75.28
Colon	80.65	87.10	87.10	83.87	84.68
	87.10	83.87	90.32	88.71	87.40
Embrional Tumours	76.67	75.0	73.33	75.00	75.00
	80.00	78.33	78.33	76.67	78.33
HeC Brain Tumours	92.65	86.76	92.65	92.65	91.18
	94.12	94.12	92.65	95.59	94.12
Leukemia	95.83	95.83	95.83	94.44	95.48
	97.22	97.22	97.22	97.22	97.22
Lung Cancer	98.48	95.52	98.91	98.48	97.85
	98.12	98.79	98.85	97.94	98.43
Lupus	78.57	77.26	78.45	77.38	77.92
	78.69	76.67	77.98	77.74	77.77
Lymphoma	95.56	100	88.89	93.33	94.45
	95.56	100	97.78	95.56	97.23
Average Genetic	86.57	87.34	86.34	85.95	**86.55**
	87.99	88.45	88.77	87.73	**88.22**
Arcene	86.06	84.89	87.00	85.44	85.85
	84.17	83.94	85.72	83.89	84.43
Mesh	92.06	87.30	88.89	90.48	89.68
	85.71	85.71	76.19	84.13	82.96
Average Non-genetic	89.06	86.10	87.95	87.96	**87.77**
	84.94	84.83	80.96	84.01	**83.70**

For six out of eight genetic datasets the novel representation outperforms the original one (according to the average performance and most particular accuracies). Only for *Breast* and *Lupus* the original representation reached better results on average, but with less than 0.56% difference each. The gene-pair representation could increase the average accuracy over all classifiers and datasets by 1.67%. Moreover, all the four classifiers demonstrate a better average performance with the new representation, where distance learning from equivalence constraints appears to have the biggest accuracy increase.

The experimental results show a clear benefit of the new representation, which is motivated by the fact that genes often depend on each other. To validate the assumption that this is the main reason for its better performance, two similar non-genetic datasets have also been tested. Table 1 shows how the gene-pair representation may fail for non-genetic data. The results indicate that the benefit of the gene-pair representation presumably relies on the interactions of genes indeed.

From the results of our experiments with noisy data (these results are not included in this paper for the sake of brevity) it could be seen that the novel representation is unfortunately less robust to noise with respect to the original representation. For no noise the pairs outperform the single genes by 1.67% while with 10% class noise they

perform almost equal and with more noise the gene-pair representation performed worse. With the gene-pair representation it is easier to overfit noise; one thus needs to be cautious and take this into consideration when using it.

The statistics of selected pairs show a tendency to select a big number of pairs where the same feature is included. In some cases this feature was highly ranked in the old representation as well. However some features being part of many pairs have not been ranked high for the plain representation. A deeper and more thorough analysis of these statistics is a direction for future work. In addition, it is also important to note that selected gene pairs will not only be useful for learning algorithms but can also provide precious information about gene interactions having important influence on the disease under study.

4.2 Incorporation of the GO Semantic Similarity

For the experiments with the GO semantic similarity only six datasets could be used as it is necessary to have genetic data where the gene-names are known. As *Leukemia* includes only 38 features that can be matched to genes, this dataset has thus been excluded from the experiments. As the GO is not complete and some features could not be matched to genes in the GO, the datasets have thus been reduced in their number of features. Some discriminative features may have been deleted and therefore the accuracies reported for these experiments are not comparable with the results of the pair representation evaluation.

In the main experiments, 400 pairs with the highest GIC semantic similarity have been pre-selected for each dataset followed by a CFS feature selection to reduce the number of pairs to 100. The combination with a feature subset selection method (CFS) is needed to eliminate redundant and irrelevant features. It must be noted that for some of highly semantically similar gene pairs expression values are also identical or strongly correlated without any deviations. This makes these pairs useless for classification and thus necessitates their removal. The tests have been performed under the same conditions as described in the previous section. The main results can be found in Table 2.

Table 2. Accuracies with and without the guidance of the GO-based semantic similarity

Dataset/ GIC	*RF*	*kNN*	*iRF*	*Average*
Breast/ no GIC	69.00	70.22	68.00	69.07
Breast/ GIC	74.22	73.56	72.33	**73.37**
Colon/ no GIC	88.71	88.71	87.10	**88.17**
Colon/ GIC	83.87	88.71	85.48	86.02
Embr./ no GIC	75.00	73.33	73.33	73.89
Embr./ GIC	81.67	81.67	78.33	**80.56**
HeC/ no GIC	89.71	91.18	89.71	90.20
HeC/ GIC	92.65	94.12	92.65	**93.14**
Lupus/ no GIC	80.60	79.17	79.76	**79.84**
Lupus/ GIC	78.21	79.05	78.10	78.45

As can be seen from Table 2, the GIC guidance often results in an increase in accuracy. The biggest increase was reached with the Embryonal Tumours dataset, 6.67%. This is the best accuracy ever reached in all experiments conducted for this study. The overall average accuracy with no GIC guidance is 80.23%. The GIC guidance improves this number by 2.08% to 82.31%. It could be noted that for datasets where the average semantic similarity of selected pairs was much higher than for all the pairs, the GIC-based pre-selection could also improve accuracy.

The bad performance for Lupus is not surprising as the gene-pair representation failed for this dataset too. One reason for this might be that for this dataset gene interactions are not so important or not reflected well in the data. Another clearly affecting factor here is the fact that for many genes which might be discriminative information is still absent in the GO and they were thus excluded from the experiments (both for Lupus and Colon a relatively bigger number of genes are not yet included in the GO).

5 Conclusions

A new representation for genetic datasets has been proposed, which emphasizes interaction between genes. The new representation was shown to increase accuracy on genetic data by 1.67%. The assumption that this increase is caused by the reflection of gene-gene interactions could be validated by testing the gene-pair representation on non-genetic datasets where worse results were obtained. The gene-pair representation increased the accuracy for six out of eight genetic datasets. There is a clear tendency, although validating the statistical significance of this finding is rather difficult (all known tests for significance are insensitive to small samples). Beside the increase in accuracy and the possibility of the incorporation of semantic similarity, the new representation may lead to the discovery of new important domain knowledge.

GO-based feature selection was shown to perform well in combination with a common data-driven feature selection method (CFS) and the use of GIC similarity and could improve the classification accuracy by 2.08% over the tested datasets in comparison with the use of the plain feature selection only. More thoroughly analyzing dependencies between the data driven feature importance measure and the semantic similarity of the pairs can provide a better understanding of how to better use the GO similarity for feature selection. A histogram of dependencies between the semantic similarity of the pairs for Colon and the GainRatio feature merit of the gene-pairs is shown in Figure 1. The pairs are divided into 100 bins based on their semantic similarity values, where each bin contains the same number of pairs. The histogram shows that the first groups, the groups with the highest semantic similarity, are ranked low on average (this trend holds true for all datasets, at least for all those included in our study). Notice that the low value is an average value over the GainRatio of the pairs included in the bin. There are also gene-pairs of high importance within the first bins. As expected, the gene pairs with low semantic similarity values are clearly useless for classification. To better analyze these trends and to use the knowledge that can be derived from these correlations for better feature selection is a promising direction for future work.

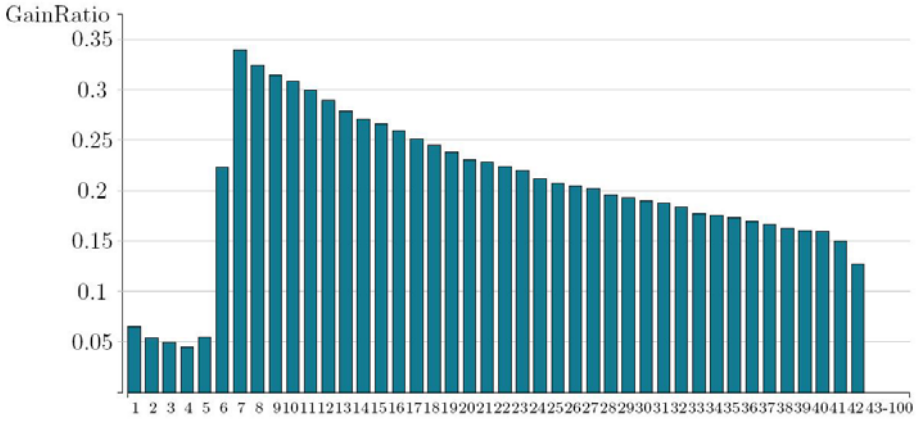


Fig. 1. Average GainRatio for gene-pair sets with different semantic similarity for Colon data

Acknowledgements. This work has been partially funded by the EU project Health-e-Child (IST 2004-027749).

References

1. Alizadeh, A., Eisen, M., Davis, R., et al.: Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403(3), 503–511 (2000)
2. Alon, U., Barkai, N., Notterman, D.A., et al.: Broad patterns of gene expression revealed by clustering of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Natl. Acad. Sci. USA 96(12), 6745–6750 (1999)
3. Ashburner, M., Ball, C., Blake, J., et al.: Gene ontology: tool for the unification of biology. Nature Genetics 25(1), 25–29 (2000)
4. Baechler, E., Batliwalla, F., Karypis, G., et al.: Interferon-inducible gene expression signature in peripheral blood cells of patients with severe lupus. Proc. Natl. Acad. Sci. 100(5) (2003)
5. Bar-Hillel, A.: Learning from weak representations using distance functions and generative models. PhD thesis, The Hebrew University of Jerusalem (2006)
6. Blake, C., Merz, C.J.: UCI repository of machine learning databases (1998), `http://archive.ics.uci.edu/ml/`
7. Chen, Z., Tang, J.: Using gene ontology to enhance effectiveness of similarity measures for microarray data. In: IEEE Inter. Conf. on Bioinformatics and Biomedicine, pp. 66–71 (2008)
8. Golub, T.R., Slonim, D.K., Tamayo, P., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286, 531–537 (1999)
9. Gordon, G., Jensen, R., Hsiao, L., et al.: Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. Cancer Research 62(17), 4963–4967 (2002)
10. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A.: Feature Extraction, Foundations and Applications. Springer, Heidelberg (2006)

11. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations 11(1) (2009)
12. Hertz, T.: Learning Distance Functions: Algorithms and Applications. PhD thesis, The Hebrew University of Jerusalem (2006)
13. International Human Genome Sequencing Consortium: Finishing the euchromatic sequence of the human genome. Nature 431(7011), 931–945 (2004)
14. Ionasec, R.I., Tsymbal, A., Vitanovski, D., Georgescu, B., Zhou, S.K., Navab, N., Comaniciu, D.: Shape-based diagnosis of the aortic valve. In: Proc. SPIE Medical Imaging (2009)
15. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Int. Conf. Research on Computational Linguistics (1997)
16. Kustra, R., Zagdanski, A.: Incorporating Gene Ontology in clustering gene expression data. In: Proc.19th IEEE Symposium on Computer-Based Medical Systems, CBMS 2006 (2006)
17. Larranaga, P., Calvo, B., Santana, R., et al.: Machine learning in bioinformatics. Brief Bioinform. 7(1), 86–112 (2006)
18. Lin, D.: An information-theoretic definition of similarity. Morgan Kaufmann, San Francisco (1998)
19. Pesquita, C., Faria, D., Bastos, H., Falcão, A.O., Couto, F.M.: Evaluating GO-based semantic similarity measures. In: Proc. 10th Annual Bio-Ontologies Meeting (2007)
20. Pomeroy, S., Tamayo, P., Gaasenbeek, M., et al.: Prediction of central nervous system embryonal tumour outcome based on gene expression. Nature 415, 436–442 (2002)
21. Qi, J., Tang, J.: Integrating gene ontology into discriminative powers of genes for feature selection in microarray data. In: Proc. ACM Symposium on Applied Computing (2007)
22. Quackenbush, J.: Computational analysis of microarray data. Nature Reviews Genetics 2(6), 418–427 (2001)
23. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: Proc. 14th Int. Joint Conf. on Artificial Intelligence (1995)
24. Saeys, Y., Inza, I., Larranaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics 23(19), 2507–2517 (2007)
25. Sevilla, J., Segura, V., Podhorski, A., et al.: Correlation between gene expression and GO semantic similarity. IEEE/ACM Trans. Comput. Biol. Bioinformatics 2(4), 330–338 (2005)
26. Tsymbal, A., Huber, M., Zhou, K.: Neighbourhood graph and learning discriminative distance functions for clinical decision support. In: Proc. IEEE Eng. Med. Biol. Soc. Conf. (2009)
27. van 't Veer, L.J., Dai, H., van de Vijver, M.J., et al.: Gene expression profiling predicts clinical outcome of breast cancer. Nature 415, 530–536 (2002)
28. Wang, H., Azuaje, F.: An ontology-driven clustering method for supporting gene expression analysis. In: Proc. 18th IEEE Symposium on Computer-Based Medical Systems, CBMS 2005, pp. 389–394. IEEE Computer Society, Los Alamitos (2005)

A Fuzzy View on Rough Satisfiability

Anna Gomolińska

Białystok University, Department of Mathematics,
Akademicka 2, 15267 Białystok, Poland
anna.gom@math.uwb.edu.pl

Abstract. In the paper, several notions of rough satisfiability of formulas are recalled and discussed from the standpoint of fuzzy set theory. By doing so we aim to better understand what rough satisfiability really means and to look for schemata describing it.

Keywords: fuzzy set, satisfiability of formulas, descriptor language, approximation space.

To the memory of my Father Kazimierz

1 Introduction

A wise judgement of satisfiability under imperfect information is a useful faculty an intelligent system can possess. Such a system would judge in a reasonable, flexible way, e.g. about satisfiability of conditions for application of rules and execution of actions and about satisfiability of specifications when constructing complex objects.

Descriptor languages for Pawlak information systems are taken as specification languages whose formulas are judged under satisfiability by objects in rough approximation spaces. Let a non-empty, finite set of objects U, information about elements of U in terms of attribute values, and – possibly – a domain knowledge be given. U can be viewed as merely a sample of the actual universe. The available information and knowledge are by necessity imperfect. A learning system S is supposed to discover – with help of an expert – how to judge satisfiability of formulas of a descriptor language $\mathcal{L}$ by objects of the universe. Briefly speaking, S's goal is to learn a suitable concept of satisfiability of formulas of $\mathcal{L}$ by objects. The extent of an expert's help may vary from the case where S is given some number of examples of satisfiability judgement only to the case where S is provided with a nearly 'ready-to-use' satisfiability notion.

We consider several rough satisfiability notions two of which are new, whereas the rest was described in [4,5]. Each of the notions may be proposed to S which, in turn, should discover (e.g., by inductive learning) parameter values, an approximation space, and the language which the best fit a considered judgemental situation. A general discussion of satisfibility judgement is out of the scope of this article, and we rather aim at finding one or more schemata defining rough

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 227–236, 2010.

satisfiability if there are any. It seems that fuzzy set theory is a framework suitable for this purpose. By re-interpreting rough satisfiability concepts in fuzzy set terms we hope to shed a new light on the problem of satisfiability of formulas in approximation spaces.

Fuzzy sets, viewed as a bunch of theories, is one of the main mathematical approaches to model vagueness and uncertainty. In the fuzzy set framework, vague concepts (e.g., 'tall man', 'nice weather', 'safe driving'), commonly used by humans, are represented by fuzzy sets. The idea of a fuzzy set, introduced by Zadeh [29], gave rise to a research field intensively exploited both by theoreticians and practitioners. At first glance, the fuzzy and rough set approaches seemed to be competitors. As turned out they rather complement each other. Note that combination of fuzzy sets with rough sets resulted in two hybrid approaches: rough-fuzzy sets and fuzzy-rough sets (see, e.g., [3,12,13]).

The rest of the article is organized as follows. In Sect. 2, approximation spaces are overviewed in a nutshell. A Pawlak information system and a descriptor language for it are recalled in Sect. 3. The main section is Sect. 4 where we present several rough satisfiability models, partly known from the author's earlier works, and we give them a fuzzy interpretation. The results are summarized in Sect. 5.

2 Approximation Spaces

Consider $U \neq \emptyset$ whose elements and subsets are referred to as objects and concepts, respectively. U is covered by clumps of objects drawn together on the basis of similarity. After Zadeh [30] these clusters are called information granules (infogranules for short). Indistinguishability of objects is treated as a limit case of similarity. Equivalence relations serve as mathematical models of indistinguishability, whereas reflexive relations are used to model similarity.

An approximation space provides a frame and tools for approximation of concepts. Since its origin [15], approximation spaces have been generalized in several ways, so nowadays it is an umbrella term for a number of structures within which one can approximate concepts [23,24,25,28,32]. In our approach, an approximation space is a structure of the form $M = (U, \varrho, \kappa)$ where U is as earlier, ϱ is a reflexive relation on U referred to as a similarity relation, and κ is a weak quasi-rough inclusion function (q-RIF) upon U, i.e., a mapping $\kappa : (\wp U)^2 \mapsto [0, 1]$ fulfilling rif_0 and rif_2:

$$\mathrm{rif}_0(\kappa) \overset{\mathrm{def}}{\Leftrightarrow} \forall X, Y \subseteq U.(X \subseteq Y \;\Rightarrow\; \kappa(X, Y) = 1),$$
$$\mathrm{rif}_2(\kappa) \overset{\mathrm{def}}{\Leftrightarrow} \forall X, Y, Z \subseteq U.(Y \subseteq Z \;\Rightarrow\; \kappa(X, Y) \leq \kappa(X, Z)).$$

2.1 Elementary Infogranules

With reading '$(u, u') \in \varrho$' as 'u is similar to u'', the counter-image of $\{u\}$ given by ϱ, $\varrho^{\leftarrow}\{u\}$, consists of all objects similar to u. Such counter-images are viewed as elementary infogranules here.

By a granulation mapping we mean any mapping assigning infogranules to objects. Among them are uncertainty mappings introduced by Skowron and Stepaniuk [23]. An example of such a mapping is $\Gamma : U \mapsto \wp U$ such that $\Gamma u \stackrel{\text{def}}{=} \varrho^{\leftarrow}\{u\}$.

2.2 Approximation Operators

There are many possibilities for concept approximation. In line with the classical Pawlak approach [15,17], the lower and upper P-approximation operators $\text{low}^P, \text{upp}^P : \wp U \mapsto \wp U$ may be given by

$$\text{low}^P X \stackrel{\text{def}}{=} \{u \in U \mid \Gamma u \subseteq X\} \;\&\; \text{upp}^P X \stackrel{\text{def}}{=} \{u \in U \mid \Gamma u \cap X \neq \emptyset\}. \quad (1)$$

A concept X is said to be P-exact if $\text{upp}^P X = \text{low}^P X$; otherwise, X is P-rough. In Skowron and Stepaniuk's framework [23], the lower and upper S-approximation operators $\text{low}^S, \text{upp}^S : \wp U \mapsto \wp U$ are defined by

$$\text{low}^S X \stackrel{\text{def}}{=} \{u \in U \mid \kappa(\Gamma u, X) = 1\} \;\&\; \text{upp}^S X \stackrel{\text{def}}{=} \{u \in U \mid \kappa(\Gamma u, X) > 0\}. \quad (2)$$

Here is X called S-exact if $\text{upp}^S X = \text{low}^S X$; otherwise, X is S-rough. In general, P- and S-approximation operators are different[1]. Let $t \in [0, 1]$. The t-positive and t-negative region operators $\text{pos}_t, \text{neg}_t : \wp U \mapsto \wp U$ may be defined as follows[2]:

$$\text{pos}_t X \stackrel{\text{def}}{=} \{u \in U \mid \kappa(\Gamma u, X) \geq t\} \;\&\; \text{neg}_t X \stackrel{\text{def}}{=} \{u \in U \mid \kappa(\Gamma u, X) \leq t\}. \quad (3)$$

Note that[3] $\text{low}^S X = \text{pos}_1 X$ and $\text{upp}^S X = U - \text{neg}_0 X$.

2.3 Rough Inclusion

When dealing with a granulated universe of objects, the usual set-theoretical inclusion may not suffice. Therefore graded inclusions of which rough inclusion is a prominent representative are of use [6,7,19,26,27]. Rough mereology by Polkowski and Skowron [21], extending Leśniewski's mereology, may be viewed as a formal theory of rough inclusion. Realizations of rough inclusion are rough inclusion functions (RIFs for short). A RIF upon U is mapping $\kappa : (\wp U)^2 \mapsto [0, 1]$ such that $\text{rif}_1(\kappa)$ and $\text{rif}_2^*(\kappa)$ hold[4] where

$$\text{rif}_1(\kappa) \stackrel{\text{def}}{\Leftrightarrow} \forall X, Y \subseteq U.(\kappa(X, Y) = 1 \;\Leftrightarrow\; X \subseteq Y),$$
$$\text{rif}_2^*(\kappa) \stackrel{\text{def}}{\Leftrightarrow} \forall X, Y, Z \subseteq U.(\kappa(Y, Z) = 1 \;\Rightarrow\; \kappa(X, Y) \leq \kappa(X, Z)).$$

[1] The lower approximation operators are equal if κ is a RIF which means that $\text{rif}_0^{-1}(\kappa)$ holds where $\text{rif}_0^{-1}(\kappa) \stackrel{\text{def}}{\Leftrightarrow} \forall X, Y \subseteq U.(\kappa(X, Y) = 1 \;\Rightarrow\; X \subseteq Y)$. A sufficient condition for the upper approximation operators to be equal is that $\text{rif}_5(\kappa)$ holds where $\text{rif}_5(\kappa) \stackrel{\text{def}}{\Leftrightarrow} \forall X, Y \subseteq U.(X \neq \emptyset \;\Rightarrow\; (\kappa(X, Y) = 0 \Leftrightarrow X \cap Y = \emptyset))$.

[2] They are generalized variants of the early Ziarko operations [31].

[3] We have that $\text{neg}_t X = \text{pos}_{1-t}(U - X)$ if $\text{rif}_6(\kappa)$ holds where $\text{rif}_6(\kappa) \stackrel{\text{def}}{\Leftrightarrow} \forall X, Y \subseteq U.(X \neq \emptyset \;\Rightarrow\; \kappa(X, Y) + \kappa(X, U - Y) = 1)$.

[4] The latter postulate may be replaced by rif_2.

Mappings $\kappa^{\pounds}$ (the standard RIF whose idea goes back to Jan Łukasiewicz), κ_1, and κ_2 (the latter mentioned in [2]) as below are examples of different, yet mutually definable RIFs [6]. Assume for a while that U is finite.

$$\kappa^{\pounds}(X,Y) \stackrel{\text{def}}{=} \begin{cases} \frac{\#(X\cap Y)}{\#X} & \text{if } X \neq \emptyset, \\ 1 & \text{otherwise,} \end{cases}$$

$$\kappa_1(X,Y) \stackrel{\text{def}}{=} \begin{cases} \frac{\#Y}{\#(X\cup Y)} & \text{if } X \cup Y \neq \emptyset, \\ 1 & \text{otherwise,} \end{cases}$$

$$\kappa_2(X,Y) \stackrel{\text{def}}{=} \frac{\#((U-X)\cup Y)}{\#U}. \tag{4}$$

As turned out, functions which do not fully satisfy postulates for RIFs, e.g. weak q-RIFs can also prove useful in measuring the degree of inclusion. For any $t \in [0,1]^2$, by $\pi_i(t)$ we denote the i-th element of t $(i = 1, 2)$. Given a RIF $\kappa : (\wp U)^2 \mapsto [0,1]$ and $t \in [0,1]^2$ where $\pi_1(t) < \pi_2(t)$, a mapping $\kappa_t : (\wp U)^2 \mapsto [0,1]$ defined below is a weak q-RIF (but not a RIF) [26][5]:

$$\kappa_t(X,Y) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } \kappa(X,Y) \leq \pi_1(t), \\ \frac{\kappa(X,Y)-\pi_1(t)}{\pi_2(t)-\pi_1(t)} & \text{if } \pi_1(t) < \kappa(X,Y) < \pi_2(t), \\ 1 & \text{if } \kappa(X,Y) \geq \pi_2(t). \end{cases} \tag{5}$$

3 Descriptor Languages for Pawlak Information Systems

Assume that U is finite and its objects are only known by descriptions in terms of attribute values. Objects, attributes, and attribute values are denoted by u, a, and v, with sub/superscripts whenever needed. Each attribute a is viewed as a mapping from U into $V_a \cup \{*\}$ where V_a is a set of values of a and $*$ is to symbolize the lack of information about attribute values (for the treatment of missing values see, e.g., [8,9,11]). Let A denote a non-empty set of attributes and $V = \bigcup_{a\in A} V_a$. It is assumed for practical reasons that $\#U \geq 2$ and each of A's distinguishes at least two objects. Pairs of the form $\mathrm{IS} = (U, A)$ are called Pawlak information systems (infosystems for short) [14,17]. Since object descriptions can be indistinguishable from one another, or similar in some respect, the universe U is perceived as covered by infogranules. Every infosystem IS as above gives rise to a family of approximation spaces with the same universe U.

A number of properties of objects and concepts of U can be expressed by formulas of the descriptor language L_{IS}[6] whose primitives are symbols denoting attributes and their values and propositional connectives $\wedge, \vee, \neg$. Pairs of the form (a, v), where $v \in V_a$, are called descriptors being the atomic formulas here. Formulas, obtained as usual, are denoted by α, β with sub/superscripts if needed, and their set is denoted by FOR.

[5] More about weak q-RIFs and their stronger versions can be found in [7].

[6] Descriptor languages originally proposed by Pawlak were both specification and query languages.

The well-known notion of crisp satisfiability is a starting point to define rough satisfiability. The crisp satisfiability relation, $\models_c$, is defined as follows, for any descriptor (a, v), $\alpha, \beta \in \mathrm{FOR}$, and $u \in U$:

$$\begin{aligned} u &\models_c (a, v) \Leftrightarrow a(u) = v, \\ u &\models_c \alpha \wedge \beta \Leftrightarrow u \models_c \alpha \ \& \ u \models_c \beta, \\ u &\models_c \alpha \vee \beta \Leftrightarrow u \models_c \alpha \text{ or } u \models_c \beta, \\ u &\models_c \neg\alpha \Leftrightarrow u \not\models_c \alpha. \end{aligned} \tag{6}$$

By the crisp extension of α we mean the infogranule of objects crisply satisfying α, i.e., $\mathrm{Sat}_c(\alpha) \stackrel{\mathrm{def}}{=} \{u \in U \mid u \models_c \alpha\}$. Hence, $\mathrm{Sat}_c(a, v) = \{u \in U \mid a(u) = v\}$, $\mathrm{Sat}_c(\alpha \wedge \beta) = \mathrm{Sat}_c(\alpha) \cap \mathrm{Sat}_c(\beta)$, $\mathrm{Sat}_c(\alpha \vee \beta) = \mathrm{Sat}_c(\alpha) \cup \mathrm{Sat}_c(\beta)$, and $\mathrm{Sat}_c(\neg\alpha) = U - \mathrm{Sat}_c(\alpha)$.

4 Rough Satisfiability: A Fuzzy Perspective

Given an approximation space $M = (U, \varrho, \kappa)$, based on $\mathrm{IS} = (U, A)$, and the descriptor language L_{IS}. Every $X \subseteq U$ is uniquely represented by its membership function $\mu_X : U \mapsto \{0, 1\}$ such that for any $u \in U$, $\mu_X(u) = 1$ if and only if $u \in X$. Fuzzy sets [29] are obtained by allowing any reals from $[0, 1]$ as values of membership functions. Let X be a fuzzy set with a membership function $\mu_X : U \mapsto [0, 1]$ and $t \in [0, 1]$. Sets

$$X_t \stackrel{\mathrm{def}}{=} \{u \in U \mid \mu_X(u) \geq t\} \ \& \ X_t^{>} \stackrel{\mathrm{def}}{=} \{u \in U \mid \mu_X(u) > t\} \tag{7}$$

are called the t-cut and the strong t-cut of X, respectively. The 1-cut and the strong 0-cut of X (i.e., X_1 and $X_0^{>}$) are referred to as the core and the support of X, respectively [10]. Among generalizations of fuzzy sets are L-fuzzy sets with membership values in the universe of some lattice. The notions of the t-cut and the strong t-cut are adapted accordingly.

There are infinitely many possibilities of defining operations of fuzzy intersection, union, and complementation. There is an aggreement upon that intersections and unions should be computed by means of triangular norms and co-norms, respectively. Every associative, commutative, and monotone mapping $f : [0, 1]^2 \mapsto [0, 1]$ such that for any $x \in [0, 1]$, $f(1, x) = x$ (resp., $f(0, x) = x$) is called a triangular norm (co-norm). The intersection of fuzzy sets X, Y upon U induced by a triangular norm f, $X \cap_f Y$, and the union of X, Y induced by a triangular co-norm g, $X \cup_g Y$, are defined by

$$\mu_{X \cap_f Y}(u) \stackrel{\mathrm{def}}{=} f(\mu_X(u), \mu_Y(u)) \ \& \ \mu_{X \cup_g Y}(u) \stackrel{\mathrm{def}}{=} g(\mu_X(u), \mu_Y(u)). \tag{8}$$

The standard fuzzy intersections and unions are induced by the functions of minimum and maximum, respectively [10]. A complement of X can be obtained according to the formula

$$\mu_{-_f X}(u) \stackrel{\mathrm{def}}{=} f(\mu_X(u)) \tag{9}$$

where $f : [0,1] \mapsto [0,1]$, a complementation function, is co-monotone, $f(0) = 1$, and $f(1) = 0$. The complement of X obtained for $f(x) = 1 - x$ is called standard [10] and denoted by $-_{\mathrm{st}}X$.

When information is imperfect, satisfiability of formulas may be viewed as a vague concept with unsharp boundaries. In particular, an extension of α, $\mathrm{Sat}(\alpha)$, may be seen as a fuzzy set whose membership function assigns to every $u \in U$, a degree of satisfiability of α by u. In such an approach, we need a method to compute membership values for objects of U in extensions of descriptors first. Next, a triangular norm f, a triangular co-norm g, and a complementation function h should be chosen suitably[7]. Finally, the induced fuzzy semantics for compound formulas can be derived, generalizing the crisp semantics:

$$\begin{aligned}
&\mu_{\mathrm{Sat}(\alpha\wedge\beta)}(u) \stackrel{\mathrm{def}}{=} f(\mu_{\mathrm{Sat}(\alpha)}(u), \mu_{\mathrm{Sat}(\beta)}(u)),\\
&\mu_{\mathrm{Sat}(\alpha\vee\beta)}(u) \stackrel{\mathrm{def}}{=} g(\mu_{\mathrm{Sat}(\alpha)}(u), \mu_{\mathrm{Sat}(\beta)}(u)),\\
&\mu_{\mathrm{Sat}(\neg\alpha)}(u) \stackrel{\mathrm{def}}{=} h(\mu_{\mathrm{Sat}(\alpha)}(u)).
\end{aligned} \tag{10}$$

Notice that despite its mathematical attractiveness, the approach may cause practical problems because such important factors as membership functions of extensions of descriptors, a triangular norm, and a fuzzy complementation function have to be discovered here.

Rough satisfiability I. In this case, rough satisfiability is modelled as a family of relations $\{\models_t\}_{t\in[0,1]}$ where $\models_t$, a relation of satisfiability to a degree t, is defined as follows [4,5], for any $\alpha \in \mathrm{FOR}$ and $u \in U$:

$$u \models_t \alpha \stackrel{\mathrm{def}}{\Leftrightarrow} \kappa(\Gamma u, \mathrm{Sat}_c(\alpha)) \geq t. \tag{11}$$

Thus, u satisfies α to a degree t, $u \models_t \alpha$, if and only if the infogranule of objects similar to u is included to a degree at least t in the infogranule of all objects satisfying α crisply[8]. The corresponding t-extension of α is given by

$$\mathrm{Sat}_t(\alpha) \stackrel{\mathrm{def}}{=} \{u \in U \mid u \models_t \alpha\}. \tag{12}$$

Observe that $\mathrm{Sat}_t(\alpha) = \mathrm{pos}_t(\mathrm{Sat}_c(\alpha))$ meaning that the t-extension of α is the t-positive region of the crisp extension of α.

From the fuzzy point of view, the vague notion of extension of α may be formalized as a fuzzy set upon U denoted by $\mathrm{Sat}^b(\alpha)$ whose membership function is defined by

$$\mu_{\mathrm{Sat}^b(\alpha)}(u) \stackrel{\mathrm{def}}{=} \kappa(\Gamma u, \mathrm{Sat}_c(\alpha)). \tag{13}$$

Thus, the t-extension of α is the t-cut of $\mathrm{Sat}^b(\alpha)$[9].

[7] For instance, g can be defined as dual to f.
[8] A similar idea can be found in [20].
[9] The right-hand side of (13) defines the degree of rough membership of u in $\mathrm{Sat}_c(\alpha)$ [18].

Rough satisfiability II. The first model can be enhanced by considering the family $\{\models_t^+\}_{t\in[0,1]}$ where $\models_t^+ \stackrel{\text{def}}{=} \models_t \cap \models_c$ [5]. The t-extension of α, $\mathrm{Sat}_t^+(\alpha)$, is defined by

$$\mathrm{Sat}_t^+(\alpha) \stackrel{\text{def}}{=} \{u \in U \mid u \models_t^+ \alpha\}, \tag{14}$$

so $\mathrm{Sat}_t^+(\alpha) = \mathrm{pos}_t(\mathrm{Sat}_c(\alpha)) \cap \mathrm{Sat}_c(\alpha)$. Let $\mathrm{Sat}^+(\alpha)$ denote the fuzzy extension of α where

$$\mu_{\mathrm{Sat}^+(\alpha)}(u) \stackrel{\text{def}}{=} \min\{\mu_{\mathrm{Sat}^b(\alpha)}(u), \mu_{\mathrm{Sat}_c(\alpha)}(u)\}. \tag{15}$$

In view of our remarks on fuzzy intersection, one can see that $\mathrm{Sat}^+(\alpha)$ is the intersection of $\mathrm{Sat}^b(\alpha)$ and $\mathrm{Sat}_c(\alpha)$ induced by the minimum function. Moreover, the t-extension of α is the t-cut of $\mathrm{Sat}^+(\alpha)$ if $t > 0$.

Rough satisfiability III. The first two approaches take into account 'positive examples' only. Now, 'negative examples' come into play, too. The pair $L = ([0,1]^2, \preceq)$, where $\preceq$ is the coordinate-wise ordering on $[0,1]^2$, is a lattice with $(0,0), (1,1)$ as the zero and the unit elements. Rough satisfiability is modelled as $\{\models_t^{\mathrm{np}}\}_{t\in[0,1]^2}$ where

$$u \models_t^{\mathrm{np}} \alpha \stackrel{\text{def}}{\Leftrightarrow} \kappa(\Gamma u, \mathrm{Sat}_c(\neg\alpha)) \leq \pi_1(t) \;\&\; \kappa(\Gamma u, \mathrm{Sat}_c(\alpha)) \geq \pi_2(t), \tag{16}$$

i.e., u satisfies α to a degree t if and only if the infogranule of objects similar to u is included to a degree not greater than $\pi_1(t)$ in the infogranule of objects crisply satisfying $\neg\alpha$ and to a degree at least $\pi_2(t)$ in the infogranule of objects crisply satisfying α [5]. The t-extension of α, $\mathrm{Sat}_t^{\mathrm{np}}(\alpha)$, is defined along the standard lines by

$$\mathrm{Sat}_t^{\mathrm{np}}(\alpha) \stackrel{\text{def}}{=} \{u \in U \mid u \models_t^{\mathrm{np}} \alpha\}, \tag{17}$$

which results in $\mathrm{Sat}_t^{\mathrm{np}}(\alpha) = \mathrm{neg}_{\pi_1(t)}(U - \mathrm{Sat}_c(\alpha)) \cap \mathrm{pos}_{\pi_2(t)}(\mathrm{Sat}_c(\alpha))$[10].

Let $\mathrm{Sat}^{\mathrm{np}}(\alpha)$ denote the L-fuzzy extension of α where

$$\mu_{\mathrm{Sat}^{\mathrm{np}}(\alpha)}(u) \stackrel{\text{def}}{=} (\mu_{-_{\mathrm{st}}\mathrm{Sat}^b(\neg\alpha)}(u), \mu_{\mathrm{Sat}^b(\alpha)}(u)). \tag{18}$$

Since $\kappa(\Gamma u, \mathrm{Sat}_c(\neg\alpha)) \leq \pi_1(t)$ if and only if $\mu_{-_{\mathrm{st}}\mathrm{Sat}^b(\neg\alpha)}(u) \geq 1 - \pi_1(t)$, the t-extension of α is the $(1-\pi_1(t), \pi_2(t))$-cut of $\mathrm{Sat}^{\mathrm{np}}(\alpha)$[11].

Rough satisfiability IV. In this approach, rough satisfiability is modelled as a relation $\models^P$ such that

$$u \models^P \alpha \stackrel{\text{def}}{\Leftrightarrow} \Gamma u \cap \mathrm{Sat}_c(\alpha) \neq \emptyset. \tag{19}$$

The P-extension of α, $\mathrm{Sat}^P(\alpha)$, defined along the standard lines by

$$\mathrm{Sat}^P(\alpha) \stackrel{\text{def}}{=} \{u \in U \mid u \models^P \alpha\}, \tag{20}$$

[10] When $\mathrm{rif}_6(\kappa)$ holds as in the case $\kappa = \kappa^{\pounds}$, the equality will be simplified to $\mathrm{Sat}_t^{\mathrm{np}}(\alpha) = \mathrm{pos}_{t_0}(\mathrm{Sat}_c(\alpha))$ where $t_0 = \max\{1-\pi_1(t), \pi_2(t)\}$.

[11] It is also the intersection of the $(1-\pi_1(t))$-cut of $-_{\mathrm{st}}\mathrm{Sat}^b(\neg\alpha)$ and the $\pi_2(t)$-cut of $\mathrm{Sat}^b(\alpha)$.

is the upper P-approximation of the crisp extension of α [5][12]. The P-extension of α is the support of every fuzzy set X upon U whose membership function satisfies

$$\mu_X(u) = 0 \;\Leftrightarrow\; \Gamma u \cap \mathrm{Sat}_c(\alpha) = \emptyset. \tag{21}$$

Rough satisfiability V. In this case, rough satisfiability is modelled as a relation $\models^S$ such that

$$u \models^S \alpha \overset{\mathrm{def}}{\Leftrightarrow} \kappa(\Gamma u, \mathrm{Sat}_c(\alpha)) > 0. \tag{22}$$

The S-extension of α, $\mathrm{Sat}^S(\alpha)$, is defined by

$$\mathrm{Sat}^S(\alpha) \overset{\mathrm{def}}{=} \{u \in U \mid u \models^S \alpha\}, \tag{23}$$

so it is the upper S-approximation of the crisp extension of α. Notice that the S-extension of α is just the support of $\mathrm{Sat}^b(\alpha)$.

Rough satisfiability VI. In the last approach presented here, rough satisfiability is modelled as $\{\models_t^{\mathrm{gW}}\}_{t\in[0,1]^2}$ where

$$u \models_t^{\mathrm{gW}} \alpha \overset{\mathrm{def}}{\Leftrightarrow} \kappa(\mathrm{pos}_{\pi_1(t)}(\Gamma u), \mathrm{Sat}_c(\alpha)) \geq \pi_2(t), \tag{24}$$

i.e., u satisfies α to a degree t if and only if the $\pi_1(t)$-positive region of the infogranule of objects similar to u is included to a degree at least $\pi_2(t)$ in the crisp extension of α[13]. The t-extension of α, $\mathrm{Sat}_t^{\mathrm{gW}}(\alpha)$, is defined by

$$\mathrm{Sat}_t^{\mathrm{gW}}(\alpha) \overset{\mathrm{def}}{=} \{u \in U \mid u \models_t^{\mathrm{gW}} \alpha\}. \tag{25}$$

For any $t' \in [0,1]$, let $\mathrm{Sat}(\alpha; \mathrm{pos}_{t'})$ be a fuzzy set such that

$$\mu_{\mathrm{Sat}(\alpha;\mathrm{pos}_{t'})}(u) \overset{\mathrm{def}}{=} \kappa(\mathrm{pos}_{t'}(\Gamma u), \mathrm{Sat}_c(\alpha)). \tag{26}$$

Thus, the t-extension of α is the $\pi_2(t)$-cut of $\mathrm{Sat}(\alpha; \mathrm{pos}_{\pi_1(t)})$[14].

5 Conclusions

We described six, in general different models of rough satisfiability. Models I–IV (without their fuzzy interpretation) were presented in [4,5], the last two are new. In each case except for the fourth one, a corresponding (L-)fuzzy extension of a formula was defined. The rough extension was either the support of the fuzzy extension an in cases IV and V or a family of cuts of the (L-)fuzzy extension.

[12] Notice a close relationship with Pawlak's notion of rough truth [1,16].

[13] In [5] we studied a special case, proposed by Wolski in a private communication, where $\pi_2(t) = 1$. There, u satisfied α to a degree t if and only if $\mathrm{pos}_t(\Gamma u) \subseteq \mathrm{Sat}_c(\alpha)$.

[14] In the 'Wolski' case, the t-extension of α is the core of $\mathrm{Sat}(\alpha; \mathrm{pos}_{\pi_1(t)})$.

Except for case III where L-fuzzy sets are employed, the cases discussed are instances of the following approach. Let $G_1 : U \mapsto \wp U$, $G_2 : \mathrm{FOR} \mapsto \wp U$ be granulation mappings and κ be a weak q-RIF. A fuzzy extension of α, $\mathrm{Sat}(\alpha)$, is given by $\mu_{\mathrm{Sat}(\alpha)}(u) \stackrel{\mathrm{def}}{=} \kappa(G_1(u), G_2(\alpha))$. A rough extension is defined as the support or a family of cuts of $\mathrm{Sat}(\alpha)$. This scheme can cover a large number of cases but more general schemata can also be proposed.

Although the definitions of rough satisfiability apply both to descriptors and to compound formulas, one may also use them in the case of descriptors only, searching for suitable triangular norms and conorms, and complementation functions to define satisfiability of compound formulas.

The questions what rough satisfiability is and how to define it are still open, yet a step forward has been made by capturing several cases uniformly, taking fuzzy set theory as a background.

References

1. Banerjee, M.: Rough belief change. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, pp. 25–38. Springer, Heidelberg (2006)
2. 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. Symp. Intelligent Information Systems (IIS 1998), Malbork, Poland, pp. 392–395. PAS Institute of Computer Science, Warsaw (1998)
3. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. Int. Journal of General Systems 17(2), 191–209 (1990)
4. Gomolińska, A.: A graded meaning of formulas in approximation spaces. Fundamenta Informaticae 60(1-4), 159–172 (2004)
5. Gomolińska, A.: Satisfiability and meaning of formulas and sets of formulas in approximation spaces. Fundamenta Informaticae 67(1-3), 77–92 (2005)
6. Gomolińska, A.: On certain rough inclusion functions. In: Peters, J.F., Skowron, A., Rybiński, H. (eds.) Transactions on Rough Sets IX. LNCS, vol. 5390, pp. 35–55. Springer, Heidelberg (2008)
7. Gomolińska, A.: Rough approximation based on weak q-RIFs. In: Peters, J.F., Skowron, A., Wolski, M., Chakraborty, M.K., Wu, W.-Z. (eds.) Transactions on Rough Sets X. LNCS, vol. 5656, pp. 117–135. Springer, Heidelberg (2009)
8. Greco, S., Matarazzo, B., Słowiński, R.: Handling missing values in rough set analysis of multi-attribute and multi-criteria decision problems. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) RSFDGrC 1999. LNCS (LNAI), vol. 1711, pp. 146–157. Springer, Heidelberg (1999)
9. Grzymała-Busse, J.W.: Characteristic relations for incomplete data: A generalization of the indiscernibility relation. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets IV. LNCS, vol. 3700, pp. 58–68. Springer, Heidelberg (2005)
10. Klir, G.J., Wierman, M.J.: Uncertainty-based Information: Elements of Generalized Information Theory. Physica-Verlag, Heidelberg (1998)
11. Kryszkiewicz, M.: Rough set approach to incomplete information system. Information Sciences 112, 39–49 (1998)
12. Nakamura, A.: Fuzzy rough sets. Note on Multiple-Valued Logic in Japan 9(8), 1–8 (1988)

13. Pal, S.K., Skowron, A. (eds.): Rough-Fuzzy Hybridization: A New Trend in Decision Making. Springer, Singapore (1999)
14. Pawlak, Z.: Information systems – theoretical foundations. Information systems 6(3), 205–218 (1981)
15. Pawlak, Z.: Rough sets. Int. J. Computer and Information Sciences 11, 341–356 (1982)
16. Pawlak, Z.: Rough logic. Bull. Polish Acad. Sci. Tech. 35, 253–258 (1987)
17. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer, Dordrecht (1991)
18. Pawlak, Z., Skowron, A.: Rough membership functions. In: Zadeh, L.A., Kacprzyk, J. (eds.) Fuzzy Logic for the Management of Uncertainty, pp. 251–271. John Wiley & Sons, New York (1994)
19. Polkowski, L.: Rough mereology in analysis of vagueness. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 197–204. Springer, Heidelberg (2008)
20. Polkowski, L., Semeniuk-Polkowska, M.: On intensional aspects of concepts defined in rough set theory. In: Czaja, L., Szczuka, M. (eds.) Proc. 18th Workshop on Concurrency, Specification and Programming (CS& P 2009), Kraków Przegorzały, September 2009, vol. 2, pp. 486–497. Warsaw University (2009)
21. Polkowski, L., Skowron, A.: Rough mereology: A new paradigm for approximate reasoning. Int. J. Approximated Reasoning 15(4), 333–365 (1996)
22. Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.): Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems. Physica-Verlag, Heidelberg (2001)
23. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
24. Słowiński, R., Greco, S., Matarazzo, B.: Dominance-based rough set approach to reasoning about ordinal data. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 5–11. Springer, Heidelberg (2007)
25. Słowiński, R., Vanderpooten, D.: Similarity relation as a basis for rough approximations. In: Wang, P.P. (ed.) Advances in Machine Intelligence and Soft Computing, vol. 4, pp. 17–33. Duke University Press, Durham (1997)
26. Stepaniuk, J.: Knowledge discovery by application of rough set models. In: [26], pp. 137–233 (2001)
27. Xu, Z.B., Liang, J.Y., Dang, C.Y., Chin, K.S.: Inclusion degree: A perspective on measures for rough set data analysis. Information Sciences 141, 227–236 (2002)
28. Yao, Y.Y., Wong, S.K.M.: A decision theoretic framework for approximating concepts. Int. J. of Man–Machine Studies 37(6), 793–809 (1992)
29. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)
30. Zadeh, L.A.: Outline of a new approach to the analysis of complex system and decision processes. IEEE Trans. on Systems, Man, and Cybernetics 3, 28–44 (1973)
31. Ziarko, W.: Variable precision rough set model. J. Computer and System Sciences 46(1), 39–59 (1993)
32. Ziarko, W.: Probabilistic rough sets. In: Ślęzak, D., Wang, G., Szczuka, M.S., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 283–293. Springer, Heidelberg (2005)

Rough Sets in Terms of Discrete Dynamical Systems

Marcin Wolski

Department of Logic and Philosophy of Science
Maria Curie-Skłodowska University
marcin.wolski@umcs.lublin.pl

Abstract. In the paper we consider a topological approximation space (U, τ) (induced by a given information system $\mathcal{I}$) as a discrete dynamical system; that is, we are concerned with a finite approximation space U whose topology τ is induced by a function $f : U \to U$. Our aim is to characterise these type of approximation spaces by means of orbits which represent the evolution of points of U with respect to the process f. Apart from topological considerations we also provide some algebraic characterisation of orbits. Due to the finiteness condition imposed by $\mathcal{I}$, any point $a \in U$ is eventually cyclic. In consequence, as we demonstrate, orbits are algebraically close to rough sets, e.g. they induce a Łukasiewicz algebra of order two, where the lower approximation operator may be interpreted as the action of retriving a cycle from a given orbit and the upper approximation operator may be interpreted as the action of making a given orbit cyclic.

1 Introduction

Rough set theory (RST), introduced by Pawlak in the early 80's [9,10], has been a hugely successful field of research. Not only there has been found a number of fields of application but also the mathematical foundations of rough sets have been described from the standpoint of diverse branches of mathematics. However, new areas of application of RST appear continuously and in turn new mathematical approaches to rough sets are needed. Among these areas of application one can find the task of discovering process from data or discovering and approximating interactions between (or among) objects. These tasks require a new formulation of RST which would take into account some sort of dynamics.

Our first contribution to this topic is a translation of RST into the theory of discrete dynamical systems; these systems has been already investigated in the context of computer science in [3,5]. As is well-known, a complete information system $\mathcal{I}$ induces a topological approximation space (U, τ) whose topology is in turn induced by the indiscernability relation E of $\mathcal{I}$. In the case of an incomplete system, the corresponding topology τ is produced by means of the specialisation preorder of $\mathcal{I}$. Assume that there is a process defined on U which is represented as a function $f : U \to U$ and that attributes from $\mathcal{I}$ reflects some essential properties of f. To be more precise, we assume that topology τ on U reflects f in the sense that τ is induced by f; these type of topologies are called functional. Since topologies in the scope of our interest come from information systems, we are actually interested in finite functional topologies.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 237–246, 2010.

Of course, a finite functional topology is a topology and one could apply all well-known results concerning rough sets and topology. However, our aim is to focus on the process f. Therefore we are interested in the evolution of points rather than points alone; furthermore, we would also like to view this evolution from a granular perspective which underlies RST. To this end, some new concepts are induced: e.g. a granular orbit. Starting with granular orbits of fixed length one can build a De Morgan Lattice which can be augmented by the lower and upper approximation operators in order to form a Łukasiewicz algebra. In this case, lattice operations are interpreted as kinds of interactions between objects whereas the lower approximation operator may be interpreted as the action of retriving a cycle from a given orbit and the upper approximation operator may be interpreted as the action of making a given orbit cyclic.

2 Rough Sets: Point Set Topology

In this section we introduce basic concepts from rough set theory (RST) [9,10,11] and interpret them in terms of topological spaces. Since the main aim of the paper is to relate RST with dynamical systems we also recall how an information system can be represented by means of continuous real functions.

Definition 1 (Information System). *A quadruple $\mathcal{I} = (U, Att, Val, i)$ is called an* information system*, where:*

- *U is a non–empty finite set of objects;*
- *A is a non–empty finite set of attributes;*
- *$V = \bigcup_{A \in Att} Val_A$, where Val_A is the value–domain of the attribute A, and $Val_A \cap Val_B = \emptyset$, for all $A, B \in Att$;*
- *$i : U \times Att \to Val$ is an information function, such that for all $A \in Att$ and $a \in U$ it holds that $i(a, A) \in Val_A$.*

If i is a total function, i.e. $i(a, A)$ is defined for all $a \in U$ and $A \in Att$, then the information system $\mathcal{I}$ is called complete*; otherwise, it is called* incomplete*.*

In other words, in an incomplete information system some attributes are affected by missing values.

An information system $\mathcal{I}$ can be augmented by an information ordering $\precsim_i$ – which has obvious affinities with a specialisation preorder (in topology) [1,6] or an information quantum relation (in RST) [7,8] – defined as:

$$a \precsim_i b \text{ iff } i(a, A) = U \text{ implies } i(b, A) = U, \tag{1}$$

for all attributes A from $\mathcal{I}$, such that $i(a, A)$ is defined and equals U.

Proposition 1. *For a complete information system $\mathcal{I} = (U, Att, Val, i)$, its information order $\precsim_i$ is an equivalence relation.*

Customarily, for a complete information system the relation $\precsim_i$ is called an *indiscernability relation* and it is often written as $IND(Att)$; the partition induced by the relation $IND(Att)$ is denoted by $U/IND(Att)$, and $[a]_{IND(Att)}$ denotes the equivalence class of $IND(Att)$ defined by $a \in U$. The simple generalisation of $(U, IND(Att))$ is given by the concept of an approximation space:

Definition 2 (Approximation Space). *The pair (U, E), where U is a non-empty set and E is an equivalence relation on U, is called an* approximation space. *A subset $A \subseteq U$ is called* definable *if $A = \bigcup \mathcal{B}$ for some $\mathcal{B} \subseteq U/E$, where U/E is the family of equivalence classes of E.*

Definition 3 (Approximation Operators). *Let (U, E) be an approximation space. For every concept $A \subseteq U$, its E-lower and E-upper* approximations*, are defined as follows, respectively:*

$$\underline{A} = \{a \in U : [a]_E \subseteq A\},$$
$$\overline{A} = \{a \in U : [a]_E \cap A \neq \emptyset\}.$$

The main idea of RST is to approximate any set A by means of two definable sets: $\underline{A}$ and $\overline{A}$. The lower approximation $\underline{A}$ consists of objects which necessarily belong to A, whereas the upper approximation $\overline{A}$ consists of objects which possibly belong to A.

Let $\mathcal{P}(U)$ denote the powerset of U. By the usual abuse of language and notation, the operator $\underline{\quad} : \mathcal{P}(U) \to \mathcal{P}(U)$ sending A to $\underline{A}$ will be called the *lower approximation operator*, whereas the operator $\overline{\quad} : \mathcal{P}(U) \to \mathcal{P}(U)$ sending A to $\overline{A}$ will be called the *upper approximation operator*.

An approximation space (U, E) may be converted into a topological space (U, τ_E) called an approximation topological space [12]. Customarily, Int and Cl will denote a topological interior and closure operators, respectively.

Definition 4 (Approximation Topological Space). *A topological space (U, τ_E) where U/E, the family of all equivalence classes of E, is the minimal basis of τ_E and Int is given by*

$$Int(A) = \bigcup \{[a]_E \in U/E : a \in U \& [a]_E \subseteq A\}$$

is called an approximation topological space.

Proposition 2. *For an information system $\mathcal{I} = (U, Att, Val, i)$ its information order $\precsim_i$ is a preorder, i.e. a reflexive and transitive relation.*

As is well-known, there is a one-to-one correspondence between preorders and Alexandroff topologies. This correspondence was proved by Naturman [6] and later Arenas made an important contribution to this topic [1].

For a topological space (U, τ) one can convert the relation of set inclusion on τ into a preorder $\preceq$ defined on elements of U, which is called the *specialisation preorder*:

$$a \preceq b \text{ iff } Cl(\{a\}) \subseteq Cl(\{b\}).$$

For an arbitrary preordered set $(U, \preceq)$ there is always a topology τ whose specialisation preorder is $\preceq$ and there are many of them in general.

Definition 5 (Specialisation Topology). *Let $\mathcal{U} = (U, \preceq)$ be a preordered set. A* specialisation topology *on $\mathcal{U}$ is a topology τ with a specialisation preorder $\preceq$ such that every automorphism of $\mathcal{U}$ is a homeomorphism of (U, τ).*

Definition 6 (Alexandroff Space). *A topological space (U, τ) whose topology τ is closed under arbitrary intersections and arbitrary unions is called an* Alexandroff space.

The Alexandroff topology is actually the largest specialisation topology induced by $\preceq$; furthermore, Alexandroff spaces and preordered sets regarded as categories are dually isomorphic and we may identify them.

Proposition 3 (Correspondence). *There exists a one-to-one correspondence between Alexandroff topologies on a set U and preorders on U.*

In the case of Alexandroff spaces, each $a \in U$ has the smallest neighbourhood defined as follows:

$$\nabla(a) = \bigcap \{A \in \tau : a \in A\}.$$

Furthermore, the following sets

$$\nabla'(a) = \{b \in U : a \preceq b\},$$

for all $a \in U$, form a subbasis. One can also prove that $\nabla(a) = \nabla'(a)$, for any a.

In what follows for a preordered set $(U, \preceq)$ we shall denote the corresponding Alexandroff topological space by $(U, \tau_{\preceq})$.

Of course for an information system $\mathcal{I} = (U, Att, Val, i)$, the corresponding approximation topological space has an Alexandroff topology.

Proposition 4. *Let be given an information system $\mathcal{I} = (U, Att, Val, i)$ equipped with its information order $\precsim_i$. Then $(U, \tau_{\precsim_i})$ is an Alexandroff topological space.*

Alexandroff spaces induced by information systems may be also characterised by means of continuous real functions.

Definition 7. *(U, τ) is a completely regular space if and only if, given any closed set F and any point a that does not belong to F, there is a continuous function f from U to the real line $\mathbb{R}$ such that $f(a) = 0$ and $f(b) = 1$ for every b in F.*

Corollary 1. *Let $\mathcal{I} = (U, Att, Val, i)$ a complete information system equipped with its information order $\precsim_i$. Then $(U, \tau_{\precsim_i})$ is a completely regular Alexandroff topological space.*

Thus complete information systems can be linked with continuous real functions. In the next section we shall be interested in processes on U rather than real functions, that is we shall be interested in functions from U to U.

3 Rough Sets: Discrete Dynamical Systems

In this section we introduce basic concepts from the theory of dynamical systems and apply them to rough set theoretic structures. Since we start with the concept of an information system, our interest is restricted to discrete dynamical systems. These systems has been already investigated in the context of computer science in [3,5].

Definition 8 (Discrete Dynamical System). *A* discrete dynamical system *is a pair (U, f) where U is a set and $f : U \to U$ is simply a function from U into itself.*

The standard definition of a discrete dynamical system assumes that the set U is supplied with a metric or topology (e.g. [4]). In this paper we are interested in the topology which is induced by f. Therefore, we do not assume any prior topology or metric defined on U; sometimes systems of this type are called set theoretic discrete dynamical systems.

Definition 9. *Let U be a set and $f : U \to U$. Define $f^0 = id_U$ and for all $k \geq 1$ define $f^k = f \circ f^{k-1}$.*

A key concept in the study of discrete dynamical systems is the orbit of a point.

Definition 10 (Orbit). *Let $f : U \to U$. The* f-orbit *of $a \in U$, often called the f-trajectory of a, is defined as a sequence:*

$$(f^0(a), f^1(a), f^2(a), f^3(a), \ldots)$$

The usual interpretation of orbit is that iterations of $f^n(a)$ describe the evolution of a in discrete time n. Of course, any orbit $\mathcal{O}_f(a)$ can be converted into a set $\mathcal{O}s_f(a)$:

$$\mathcal{O}s_f(a) = \{b \in U : b = f^n(a), \text{ for some } n\} \tag{2}$$

for all $a \in U$.

Definition 11 (Periodic Point). *Let $f : U \to U$ and $a \in U$. The f-orbit of a is* cyclic *if $f^n(a) = a$, for some $n \geq 1$. We also say that a is a cyclic point (or a* periodic point*) for f.*

Definition 12 (Eventually Periodic Point). *Let $f : U \to U$ and $a \in U$. The f-orbit of a is* eventually cyclic *if $f^n(a) = f^m(a)$ for some n, m with $n \neq m$. In this case we also say that a is an eventually cyclic point (or an* eventually periodic point*) for f.*

Definition 13. *Let U be a set and $f : U \to U$ a function. Define:*

$$\tau_f = \{A \subseteq U : f(A) \subseteq A\}.$$

It is easy to observe that these sets form a topology [3,5]:

Proposition 5. *Let (U, f) be a discrete dynamical system, then (U, τ_f) is an Alexandroff topological space.*

Of course not every Alexandroff topology may be obtained in this way.

Definition 14 (Functional Topology). *Let (U, σ) be a topological space. Then we say that the topology σ is a* functional topology *if there is a map $f : U \to U$ such that $\sigma = \tau_f$.*

Of course it may happen that two different functions f and g induce the same topology $\tau_f = \tau_g$. However, as it was observed by Monks [5], if every f-cyclic point a is a fixed point then from $\tau_f = \tau_g$ it follows that $f = g$. Furthermore, many concepts such as specialisation preorder can be redefined by means of f-orbits.

Proposition 6. *Let (U, τ_f) be a functional topological space. Then $\leq$ defined by*

$$a \leq b \text{ iff } b \in \mathcal{O}s_f(a),$$

for all $a, b \in U$, is the specialisation preorder of (U, τ_f).

Corollary 2. *Let (U, τ_f) be a functional topological space. Then $\mathcal{O}s_f(a)$ is the minimal open neighbourhood of $a \in U$.*

Corollary 3. *Let (U, τ_f) be a functional topological space. Then the set $\{\mathcal{O}s_f(a) : a \in U\}$ forms a minimal basis of τ_f.*

In what follows, we would like to focus our attention on information systems which describe some process on a set U.

Definition 15 (Functional Informaltion System). *Let $\mathcal{I} = (U, Att, Val, i)$ be an information system and let $\precsim_i$ denote its information order. Then $\mathcal{I}$ is called* functional *if the corresponding topological space $(U, \tau_{\precsim_i})$ is functional. If $\tau_f = \tau_{\precsim_i}$ for some $f : U \to U$ then we say that the information system $\mathcal{I} = (U, Att, Val, i)$ describes the process f.*

On the basis of Corollary 2, it easy to observe that:

Corollary 4. *Let $\mathcal{I} = (U, Att, Val, i)$ be a functional information system describing a process f. Then*

$$\underline{A} = \{a \in U : \mathcal{O}s_f(a) \subseteq A\},$$
$$\overline{A} = \{a \in U : \mathcal{O}s_f(a) \cap A \neq \emptyset\},$$

for all $a \in U$.

Corollary 5. *Let $\mathcal{I} = (U, Att, Val, i)$ be a functional information system describing a process f. Then every $a \in U$ is an eventually cyclic point for f.*

In other words, for a functional information system $\mathcal{I} = (U, Att, Val, i)$ and $a \in U$, we can represent $\mathcal{O}_f(a)$ as a finite sequence:

$$(f^0(a), f^1(a), f^2(a), f^3(a), \ldots, f^m(a)) \tag{3}$$

where m is the smallest number for which there exists $n < m$ such that $f^n(a) = f^m(a)$. In this case m will be called the length of $\mathcal{O}_f(a)$.

In order to emphasise differences among already introduced types of information systems we shall introduce Euclidean-like relations:

Definition 16 (Euclidean Relations). *Let U be a set and R a binary relation on U.*

1. *R is called an* Euclidean *relation if aRb and aRc then bRc,*
2. *R is called an* almost Euclidean *relation if aRb and aRc then bRc or cRb,*

for all $a, b, c \in U$.

Proposition 7. *Let $\mathcal{I} = (U, Att, Val, i)$ be an information system, then:*

1. *if $\mathcal{I}$ is functional then $\precsim_i$ is almost Euclidean,*
2. *if $\mathcal{I}$ is complete then $\precsim_i$ $(= IND(Att))$ is Euclidean.*

Corollary 6. *Let $\mathcal{I} = (U, Att, Val, i)$ be a functional information system (for a process f), then $\precsim_i$ is Euclidean on all f-periodic points.*

By the very definition $\precsim_i$ is a reflexive and transitive relation (see also Proposition 6). Now suppose that $a \precsim_i b$ for two periodic points $a, b \in U$. By definition of the specialisation preorder, it means that $b \in \mathcal{O}s_f(a)$. But given that b is periodic, it follows that $\mathcal{O}s_f(b) = \mathcal{O}s_f(a)$. Thus, for periodic points $\precsim_i$ is also symmetric. In consequence, $\precsim_i$ is an equivalence relation, which in turn (as is well-known) is a relation that is Euclidean and reflexive.

Corollary 7. *Let $\mathcal{I} = (U, Att, Val, i)$ be a complete information system, then $\mathcal{I}$ is a functional system (for some f) and all points of U are f-periodic.*

Assume a complete information system $\mathcal{I} = (U, Att, Val, i)$; let $IND(Att)$ denote its indiscernability relation. With each equivalence class $[a]_{IND(Att)}$ we can associate a permutation $s_{[a]} : [a]_{IND(Att)} \rightarrow [a]_{IND(Att)}$.

Corollary 8. *Let $\mathcal{I} = (U, Att, Val, i)$ be a complete information system. Then the space $(U, \tau_{IND(Att)})$ is a functional topological space induced by $f : U \rightarrow U$ defined by:*

$$f(b) = s_{[a]}(b) \text{ iff } b \in [a]_{IND(Att)}, \text{ for all } b \in U.$$

Thus complete information systems are in actual fact systems describing some permutations of U (preserving the membership to a given equivalence class), and in this sense, they are less interesting than functional information systems.

4 Granular Approach to Dynamical Systems

For a functional information system $\mathcal{I} = (U, Att, Val, i)$ and $a \in U$, $\mathcal{O}_f(a)$ represents the smallest information granule containing a. That is, whenever we have any piece of information about a it applies also to any $b \in \mathcal{O}s_f(a)$. So let us replace any $a \in U$ with $\mathcal{O}s_f(a)$ in Eq. 3.

Definition 17 (Granular Orbit). *Let $\mathcal{I} = (U, Att, Val, i)$ be a functional information system (for some f). A* granular orbit $\mathcal{GO}_f(a)$ *of a is defined as a sequence:*

$$(\mathcal{O}s_f(f^0(a)), \mathcal{O}s_f(f^1(a)), \mathcal{O}s_f(f^2(a)), \ldots, \mathcal{O}s_f(f^m(a)))$$

Since every point of $a \in U$ is eventually f-cyclic we can easily prove what follows:

Proposition 8. *Let $\mathcal{I} = (U, Att, Val, i)$ be a functional information system (for some f) and $a \in U$. Then for all f-periodic points $b, c \in \mathcal{GO}_f(a)$ it holds that $\mathcal{O}s_f(b) = \mathcal{O}s_f(c)$.*

Thus, for all $a \in U$ the set $\mathcal{O}s_f(f^m(a))$ is repeated at list once in $\mathcal{GO}_f(a)$. It allows one to easily change the length of $\mathcal{GO}_f(a)$ by adding more copies of $\mathcal{O}s_f(f^m(a))$. So, given a functional information system $\mathcal{I} = (U, Att, Val, i)$, one can take the maximal length $ml = max\{m : \ m \text{ is a length of } \mathcal{GO}_f(a), \ a \in U\}$ and set all other $\mathcal{GO}_f(b)$ to this length.

Let $\mathcal{I} = (U, Att, Val, i)$ be a functional information system (for some f), and let ml be the maximal length. As said above, we set the length of $\mathcal{GO}_f(a)$ to ml, for all $a \in U$. Now for two orbits of the same length

$$\mathcal{GO}_f(a) = (\mathcal{O}s_f(f^0(a)), \mathcal{O}s_f(f^1(a)), \mathcal{O}s_f(f^2(a)), \ldots, \mathcal{O}s_f(f^{ml}(a))$$

$$\mathcal{GO}_f(b) = (\mathcal{O}s_f(f^0(b)), \mathcal{O}s_f(f^1(b)), \mathcal{O}s_f(f^2(b)), \ldots, \mathcal{O}s_f(f^{ml}(b))$$

one can introduce an order $\subseteq$:

$$\mathcal{GO}_f(a) \subseteq \mathcal{GO}_f(b) \text{ iff } \mathcal{O}s_f(f^0(a)) \subseteq \mathcal{O}s_f(f^0(b)).$$

Let $\mathcal{I} = (U, Att, Val, i)$ be a functional information system (for some f), let $GO(\mathcal{I})$ denote the set of all f-robits of some fixed length ml. Of course, $(GO(\mathcal{I}), \subseteq)$ is a poset. As said earlier $GO(\mathcal{I})$ is also a basis of some topology τ_f; now we extend the granular approach on τ_f by means of lattice-theoretic operations induced by $\subseteq$:

$$\neg\mathcal{GO}_f(a) = (U \setminus \mathcal{O}s_f(f^0(a)), U \setminus \mathcal{O}s_f(f^1(a)), \ldots, U \setminus \mathcal{O}s_f(f^{ml}(a)))$$

$$\mathcal{GO}_f(a) \wedge \mathcal{GO}_f(b) = (\mathcal{O}s_f(f^0(a)) \cap \mathcal{O}s_f(f^0(b)), \ldots, \mathcal{O}s_f(f^{ml}(a)) \cap \mathcal{O}s_f(f^{ml}(b)))$$

$$\mathcal{GO}_f(a) \vee \mathcal{GO}_f(b) = (\mathcal{O}s_f(f^0(a)) \cup \mathcal{O}s_f(f^0(b)), \ldots, \mathcal{O}s_f(f^{ml}(a)) \cup \mathcal{O}s_f(f^{ml}(b)))$$

Now, a natural question arises: what kind of structure one can obtain when one close $GO(\mathcal{I})$ under $\neg$, $\wedge$, and $\vee$? Let denote this closure by $GO(\mathcal{I})_{lattice}$. Our intended interpretation of the lattice operations is given by some sorts of interactions between objects $a, b \in U$; as a result of these interactions the process f may be changed, what is expressed in terms of a new orbit which is not an original f-orbit.

Example 1. Let $a, b \in U$ be two objects such that $\mathcal{O}s_f(f^0(a)) \cap \mathcal{O}s_f(f^0(b)) \neq \emptyset$. Then $\mathcal{GO}_f(a) \wedge \mathcal{GO}_f(b)$ is an f-orbit. If $\mathcal{GO}_f(a) \subseteq \mathcal{GO}_f(b)$ then also $\mathcal{GO}_f(a) \vee \mathcal{GO}_f(b)$ is an f-orbit.

Example 2. Let $a, b \in U$ be two objects such that $\mathcal{O}s_f(f^0(a)) \cap \mathcal{O}s_f(f^0(b)) = \emptyset$. Then both $\mathcal{GO}_f(a) \wedge \mathcal{GO}_f(b)$ and $\mathcal{GO}_f(a) \vee \mathcal{GO}_f(b)$ are not f-orbits.

Thus, some interactions between elements change the process f and some do not.

Definition 18 (De Morgan Algebra). *A a structure $(U, \vee, \wedge, 0, 1, \neg)$ is called a* De Morgan algebra *if $(U, \vee, \wedge, 0, 1)$ is a bounded distributive lattice, and $\neg$ is a De Morgan involution:*

$$\neg(a \wedge b) = \neg a \vee \neg b, \quad \neg(a \vee b) = \neg a \wedge \neg b, \quad \neg\neg a = a.$$

Proposition 9. *Let $\mathcal{I} = (U, Att, Val, i)$ be a functional information system (for some f). Then $(GO(\mathcal{I})_{lattice}, \vee, \wedge, 1, 0)$, where*

$$1 = (U, U, \ldots, U) \quad \textit{and} \quad 0 = (\emptyset, \emptyset, \ldots, \emptyset),$$

is De Morgan algebra.

This algebra can be enriched to a structure which provides a representation for the lower and upper approximation operators.

Definition 19 (Łukasiewicz Algebra). *A* Łukasiewicz algebra *of order* n *is a structure* $(U, \vee, \wedge, 0, 1, \neg, \sigma_0 \ldots, \sigma_{n-1})$ *such that* $(U, \vee, \wedge, 0, 1, \neg)$ *is a De Morgan algebra, and*

1. σ_i *is a lattice homomorphism:*

$$\sigma_i(x \vee y) = \sigma_i(x) \vee \sigma_i(y) \text{ and } \sigma_i(x \wedge y) = \sigma_i(x) \wedge \sigma_i(y),$$

2. $\sigma_i(x) \vee \neg(\sigma_i(x)) = 1$ *and* $\sigma_i(x) \wedge \neg(\sigma_i(x)) = 0$,
3. $\sigma_i(\sigma_j(x)) = \sigma j(x)$ *for* $0 \leq j \leq n-1$,
4. $\sigma_i(\neg x) = \neg(\sigma_{n-i}(x))$,
5. $\sigma_i(x) \wedge \sigma_j(x) = \sigma_i(x)$ *for* $i \leq j \leq n-1$,
6. $x \vee \sigma_{n-1}(x) = \sigma_{n-1}(x)$ *and* $x \wedge \sigma_0(x) = \sigma_0(x)$,
7. $y \wedge (x \vee \neg(\sigma_i(x)) \vee \sigma_{i+1}(y)) = y$ *for* $i \neq n-1$.

These axioms are not independent; please consult e.g. [2] for more information about this class of algebras.

Proposition 10. *Let* $\mathcal{I} = (U, Att, Val, i)$ *be a functional information system (for some* f*) and* $(GO(\mathcal{I})_{lattice}, \vee, \wedge, 1, 0)$ *its De Morgan algebra. Define:*

$$U(\mathcal{GO}_f(a)) = (\mathcal{O}s_f(f^0(a)), \mathcal{O}s_f(f^0(a)), \ldots, \mathcal{O}s_f(f^0(a))),$$

$$L(\mathcal{GO}_f(a)) = (\mathcal{O}s_f(f^{ml}(a)), \mathcal{O}s_f(f^{ml}(a)), \ldots, \mathcal{O}s_f(f^{ml}(a))).$$

Then $(GO(\mathcal{I})_{lattice}, \vee, \wedge, 1, 0, L, U)$ *is a Łukasiewicz algebra of order 2.*

In other words, the lower approximation operator L retrieves from $\mathcal{GO}_f(a))$ its cycle, whereas the upper approximation operator U makes the orbit $\mathcal{GO}_f(a))$ periodic.

5 Summary

In the paper we have described basic concepts of rough set theory in terms of dynamical systems. Our attention has been focused on functional information system which describe some process represented by a function f. Each complete information system is functional, but not otherwise. We have described the differences between these systems in terms of Euclidean relations. The last part of the paper is devoted to granular view of orbits. It turned out that starting from orbits one could obtain quite rich lattice structures such like a De Morgan Algebra or Łukasiewicz algebra of order two.

Acknowledgements. The research has been supported by the grant N N516 368334 from Ministry of Science and Higher Education of the Republic of Poland.

References

1. Arenas, F.G.: Alexandroff spaces. Acta Math. Univ. Comenianae 68, 17–25 (1999)
2. Cignoli, R., de Gallego, M.S.: The lattice structure of some Łukasiewicz algebras. Algebra Universalis 13, 315–328 (1981)
3. Geerts, F., Kuijpers, B.: Topological formulation of termination properties of iterates of functions. Information Processing Letters 89(1), 31–35 (2004)

4. Kurka, P.: Topological and Symbolic Dynamics. Socit Mathematique de France (2004), http://www.cts.cuni.cz/~kurka/studij.html
5. Monks, K.: A category of topological spaces encoding acyclic settheoretic dynamics. In: Proceedings of the International Conference on the Collatz Problem and Related Topics (2000)
6. Naturman, C.A.: Interior Algebras and Topology. Ph.D. thesis, University of Cape Town Department of Mathematics (1991)
7. Pagliani, P., Chakraborty, M.: Information quanta and approximation spaces. I: Non-classical approximation operators. In: Hu, X., Liu, Q., Skowron, A., Lin, T.S., Yager, R.R., Zhang, E.B. (eds.) Proceedings of the IEEE International Conference on Granular Computing, vol. 2, pp. 605–610. IEEE, Los Alamitos (2005)
8. Pagliani, P., Chakraborty, M.: Information quanta and approximation spaces. II: Generalised approximation space. In: Hu, X., Liu, Q., Skowron, A., Lin, T.S., Yager, R.R., Zhang, E.B. (eds.) Proceedings of the IEEE International Conference on Granular Computing, vol. 2, pp. 611–616. IEEE, Los Alamitos (2005)
9. Pawlak, Z.: Classification of Objects by Means of Attributes. Institute for Computer Science, Polish Academy of Sciences PAS 429 (1981)
10. Pawlak, Z.: Rough sets. Int. J. Computer and Information Sci. 11, 341–356 (1982)
11. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publisher, Dordrecht (1991)
12. Rasiowa, H.: Algebraic Models of Logics. University of Warsaw (2001)

A Preference-Based Multiple-Source Rough Set Model

Md. Aquil Khan* and Mohua Banerjee**

Department of Mathematics and Statistics,
Indian Institute of Technology,
Kanpur 208 016, India
{mdaquil,mohua}@iitk.ac.in

Abstract. We propose a generalization of Pawlak's rough set model for the multi-agent situation, where information from an agent can be preferred over that of another agent of the system while deciding membership of objects. Notions of lower/upper approximations are given which depend on the knowledge base of the sources as well as on the position of the sources in the hierarchy giving the preference of sources. Some direct consequences of the definitions are presented.

1 Introduction

Pawlak's rough set model [6] is based on the simple structure of *approximation space*, consisting of an equivalence relation R over a set U of objects. As we know, the union of all equivalence classes $[x]_R$ with $[x]_R \subseteq X$ gives the lower approximation of X, denoted as $\underline{X}_R$, while the union of all equivalence classes having a non-empty intersection with X gives the upper approximation of X, denoted as $\overline{X}_R$. The set $B_R(X) := \overline{X}_R \setminus \underline{X}_R$ denotes the boundary of X. The elements of the sets $\underline{X}_R$, $(\overline{X}_R)^c$ and $B_R(X)$ are respectively the positive, negative and boundary elements of X.

With time, this simple model has seen many generalizations due to demands from different practical situations. The variable precision rough set model [11], the rough set model based on covering [7], neighborhood system [4] and tolerance relation [9], the Bayesian rough set model [10], the fuzzy rough set model [1] are a few instances of generalizations of Pawlak's rough set model. We deal with an extension to handle the multi-agent situation.

Multi-agent extensions of rough sets came into the picture at the very beginning of the development of the theory. It was present in Orłowska and Pawlak's work in [5], where each agent was assigned an equivalence relation representing an agent's knowledge base. Thus a generalized notion of Pawlak's approximation space was considered, consisting of a number of equivalence relations over the same domain. Later, Rauszer [8] continued the above study of multi-agent

* The author is presently a visiting scholar at the ILLC, Universitiet van Amsterdam. An EMECW grant financed by the European Commission supports the visit.

** The research was supported by grant NN516 368334 from the Ministry of Science and Higher Education of the Republic of Poland.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 247–256, 2010.

scenario. In addition to the knowledge base of each agent, Rauszer also considered the (strong) distributed knowledge base of the group of agents. Two objects are distinguishable with respect to the strong distributed knowledge base of a group of agents if and only if any of the agents from the group can distinguish them. The work in [5] and [8] does not address the issue of counterparts of the standard rough set concepts such as approximations of sets, definability of sets, membership functions in the multi-agent case. This issue was raised in [2,3], although the more general term 'source' is used instead of 'agent', and these notions are defined in the multiple-source context. The interest of the present work lies in the situation where a source may be preferred over another source of the system in deciding membership of an object. For instance, we could make the assumption that a source will always prefer herself (i.e. her knowledge base) over the other sources of the system. Thus with this assumption, if we find that $x \in \underline{X}_{R_1} \cap B_{R_2}(X)$ and $y \in \underline{X}_{R_2} \cap B_{R_1}(X)$, R_1, R_2 being the knowledge bases of sources 1 and 2 respectively, then source 1 will put more possibility on x to be an element of X than y. Observe that in the above conclusion, not only the knowledge base of the sources but also the preference of source 1 is playing a role. We propose a rough set model where a preference order on the set of sources will also be considered. Moreover, we shall define notions of lower/upper approximations which depend on the knowledge base of the sources as well as on the position of the sources in the hierarchy giving the preference of sources.

The remainder of this article is organized as follows. In Section 2, we present the notion of a 'multiple-source approximation system with preference' ($MSAS^P$), which is a generalization of Pawlak's approximation space as well as of the multiple-source approximation system proposed in [2]. In section 3, we investigate the notions of approximations in $MSAS^P$. Section 3.1 continues this investigation and defines notions of approximations which involve the distributed knowledge base of the group of sources. In section 4, we consider the situation where $MSAS^P$ has a number of preference lists representing the view of the sources of the system. Section 5 concludes the article.

2 Multiple-Source Approximation Systems with Preference

We recall the following basic definitions.

Definition 1 ([2]). *A* multiple-source approximation system ($MSAS$) *is a tuple* $(U, \{R_i\}_{i\in N})$*, where* N *is an initial segment of the set of positive integers, and each* $R_i, i \in N$*, is an equivalence relation on the non-empty set* U *of objects.*

N represents the set of sources, and is called the *cardinality* of the $MSAS$. Moreover, for each $i \in N$, R_i represents the knowledge base of the source i.

The following notions of lower/upper approximations were introduced in [2]. We give them in the context of a set of sources. Let $\mathfrak{F} := (U, \{R_i\}_{i\in N})$ be a $MSAS$ and $X \subseteq U$. The *strong lower approximation* $\underline{X}_{s(P)}$, *weak lower approximation* $\underline{X}_{w(P)}$, *strong upper approximation* $\overline{X}_{s(P)}$, and *weak upper approximation*

$\overline{X}_{w(P)}$ of X with respect to a non-empty $P \subseteq N$, respectively, are defined as follows.

Definition 2 ([2])

$$\underline{X}_{s(P)} := \bigcap_{i\in P}\underline{X}_{R_i};\ \ \underline{X}_{w(P)} := \bigcup_{i\in P}\underline{X}_{R_i}.$$
$$\overline{X}_{s(P)} := \bigcap_{i\in P}\overline{X}_{R_i};\ \ \overline{X}_{w(P)} := \bigcup_{i\in P}\overline{X}_{R_i}.$$

We omit the occurrence of $MSAS$ in the notation for strong/weak approximations to make the notation simple. The following relationship is obtained.

(*) $$\underline{X}_{s(P)} \subseteq \underline{X}_{w(P)} \subseteq X \subseteq \overline{X}_{s(P)} \subseteq \overline{X}_{w(P)}.$$

So, depending on the possibility of an object to be an element of a set X with respect to information provided by a group P of sources, the domain is divided into five disjoint sets, viz. $\underline{X}_{s(P)}$, $\underline{X}_{w(P)} \setminus \underline{X}_{s(P)}$, $B_{s(P)}(X) := \overline{X}_{s(P)} \setminus \underline{X}_{w(P)}$, $\overline{X}_{w(P)} \setminus \overline{X}_{s(P)}$, and $(\overline{X}_{w(P)})^c$. We term the elements of these regions as *certain positive*, *possible positive*, *certain boundary*, *possible negative* and *certain negative* element of X for the group P of sources respectively.

It is to be noted that the notions of lower/upper approximations given in Definition 2 are based on the assumption that each source is equally preferred. But, as mentioned in the Introduction, one may require to incorporate a preference ordering on the set of sources in some practical situations. Thus, we extend the notion of $MSAS$ to define the following.

Definition 3. *A $MSAS$* with preference *($MSAS^P$) is the tuple* $\mathfrak{F} := (U, \{R_i\}_{i\in N}, \{Q_i\}_{i\in N})$, *where*

- $(U, \{R_i\}_{i\in N})$ *is a $MSAS$,*
- Q_i *is a subset of N satisfying the following:*
  - **P1** $Q_1 \neq \emptyset$,
  - **P2** $Q_i = \emptyset$ *implies* $Q_j = \emptyset$ *for all* $j > i$,
  - **P3** $Q_i \cap Q_j = \emptyset$ *for* $i \neq j$.

The collection $\{Q_i\}_{i\in N}$ will be called the preference list of the sources. *The $MSAS^P$ and the preference list will be called* strict *if it satisfies the additional condition,*

P4 $|Q_i| \leq 1$ *for all* $i \in N$.

The preference list $\{Q_i\}_{i\in N}$ signifies that the sources belonging to Q_i are preferred over the sources belonging to Q_j, $j > i$. Moreover, all the sources belonging to the same Q_i are equally preferred. Depending on applications, one may wish to put different conditions on the preference list. In this article, we will consider the above two types of lists.

Observe that we have not asked for the condition $\bigcup_{i\in N} Q_i = N$. So, there may be sources in the system which do not find a place in the list. One may give two interpretations for this. It could be the case that one does not want to take some of the sources of system into consideration at all – say, due to the possibility of a serious error in their knowledge bases. Another interpretation could be that we may not have enough information to grade some of the sources.

It is natural to say that a source will not be preferred over herself – (P3) corresponds to this condition. In general, a $MSAS^P$ keeps the possibility open where two sources are equally preferred. Condition (P4) rules out this situation to give a strict $MSAS^P$.

3 Notions of Approximation in $MSAS^P$

Let us consider an $MSAS^P$ $\mathfrak{F} := (U, \{R_i\}_{i\in N}, \{Q_i\}_{i\in N})$, a subset X of the domain U and an object $x \in U$. Suppose we want to decide whether the object x is an element of the set X or not. If x falls outside the certain boundary region of the group Q_1 consisting of most preferred sources, then we will be able to take some decision here. Otherwise, we may like to use the knowledge base of other sources. In that case, instead of using the knowledge base of all the sources of the system, one may like to use only the knowledge base of sources in Q_2, the set of the second most preferred sources. If the object does not fall in the certain boundary region of X with respect to the group Q_2, then we will be able to take some decision. Otherwise, we may like to move to next preferred sources, i.e. to the sources of Q_3 and repeat the process. But as we descend in the hierarchy of sources given by preference list, faith on the decision will also keep reducing. This observation motivates us to give the following notions of approximations. Let $\mathfrak{F} := (U, \{R_i\}_{i\in N}, \{Q_i\}_{i\in N})$ be a $MSAS^P$ and $X \subseteq U$. Recall that for non-empty $P \subseteq N$, $B_{s(P)}(X)$ denotes the set $\overline{X}_{s(P)} \setminus \underline{X}_{w(P)}$ consisting of certain boundary elements of X for P.

Definition 4. *The strong and weak lower approximations of X of level n, $1 \leq n \leq |N|$, denoted as $L^s(X,n)$ and $L^w(X,n)$ respectively, are defined as follows.*

- $L^s(X,1) := \underline{X}_{s(Q_1)}$, $L^w(X,1) := \underline{X}_{w(Q_1)}$.

For $n > 1$

- $L^s(X,n) := \begin{cases} \bigcap_{i=1}^{n-1} B_{s(Q_i)}(X) \cap \underline{X}_{s(Q_n)} & \text{if } Q_n \neq \emptyset \\ \emptyset & \text{otherwise.} \end{cases}$
- $L^w(X,n) := \begin{cases} \bigcap_{i=1}^{n-1} B_{s(Q_i)}(X) \cap \underline{X}_{w(Q_n)} & \text{if } Q_n \neq \emptyset \\ \emptyset & \text{otherwise.} \end{cases}$

The notions of strong and weak upper approximation of X of level n, denoted as $U^s(X,n)$ and $U^w(X,n)$ respectively, are defined as follows.

- $U^w(X,1) := \overline{X}_{w(Q_1)}$, $U^s(X,1) := \overline{X}_{s(Q_1)}$.

For $n > 1$

- $U^w(X,n) := \begin{cases} \bigcup_{i=1}^{n-1} (B_{s(Q_i)}(X))^c \cup \overline{X}_{w(Q_n)} & \text{if } Q_n \neq \emptyset \\ U & \text{otherwise.} \end{cases}$
- $U^s(X,n) := \begin{cases} \bigcup_{i=1}^{n-1} (B_{s(Q_i)}(X))^c \cup \overline{X}_{s(Q_n)} & \text{if } Q_n \neq \emptyset \\ U & \text{otherwise.} \end{cases}$

If $x \notin U^w(X,n)$, x is negative element of X for some source belonging to Q_n, but the presence of $\bigcup_{i=1}^{n-1}(B_{s(Q_i)}(X))^c$ in the definition of $U^w(X,n)$ guarantees that x is boundary element of X for each source belonging to $\bigcup_{i=1}^{n-1} Q_i$.

We know that notions of approximations can be viewed as functions from the power set 2^U of set U of objects to 2^U. Given a $MSAS^P$ $(U, \{R_i\}_{i\in N}, \{Q_i\}_{i\in N})$, the strong lower approximation of level n, $1 \leq n \leq |N|$, is a function which maps $X(\subseteq U)$ to $L^s(X,n)$. Consider a map $f : 2^U \to 2^U$ and the following properties.

1. $f(X) \subseteq X$.
2. $f(X \cap Y) \subseteq f(X) \cap f(Y)$.
3. $f(X \cap Y) \supseteq f(X) \cap f(Y)$.
4. $f(X) \cup f(Y) \subseteq f(X \cup Y)$.
5. For $X \subseteq Y$, $f(X) \subseteq f(Y)$.
6. $f(U) = U$.
7. $f(f(X)) = f(X)$.

It is well-known that Pawlak's lower approximation satisfies all the above properties. As shown in [2], notions of strong and weak lower approximations given by Definition 2 satisfy 1-6 and 1,2,4-7 respectively. The generalized notions of strong/weak lower approximations given by Definition 4 are not very well-behaved with respect to these properties. In fact, the strong and weak lower approximations of level n, $n > 1$, satisfy only 1,3 and 1 respectively. The following proposition lists a few more properties of these approximations.

Proposition 1. *1.* $L^s(X,n) \subseteq L^w(X,n) \subseteq X \subseteq U^s(X,n) \subseteq U^w(X,n)$.
2. $L^s(X^c,n) = (U^w(X,n))^c$.
3. $L^w(X^c,n) = (U^s(X,n))^c$.
4. $L^s(X,n) \cap L^r(X,m) = \emptyset$ *for* $m \neq n$, $r \in \{s,w\}$.
5. $U^s(X,n) \cap U^r(X,m) = \emptyset$ *for* $m \neq n$, $r \in \{s,w\}$.
6. If $|Q_n| = 1$, *then* $L^s(X,n) = L^w(X,n)$ *and* $U^s(X,n) = U^w(X,n)$.

The proof is a direct consequence of the definitions. Items 2 and 3 show that U^s and U^w are the duals of L^w and L^s respectively.

Given a $MSAS^P$ $\mathfrak{F} := (U, \{R_i\}_{i\in N}, \{Q_i\}_{i\in N})$ and $X \subseteq U$, depending on the possibility of being an element of X, we name the elements of U following the nomenclature used for $MSAS$.

Definition 5. *Let* $1 \leq n \leq |N|$. $x \in U$ *is said to be a*

certain positive *element of* X *of level* n, *if* $x \in L^s(X,n)$,
possible positive *element of* X *of level* n, *if* $x \in L^w(X,n) \setminus L^s(X,n)$,
certain negative *element of* X *of level* n, *if* $x \in (U^w(X,n))^c$,
possible negative *element of* X *of level* n, *if* $x \in U^w(X,n) \setminus U^s(X,n)$,

Elements of $L^s(X,1)$ and $L^w(X,n) \setminus L^s(X,n)$ have respectively the highest and second highest possibility of being an element of X. Similarly, the elements of $(U^w(X,1))^c$ and $x \in U^w(X,n) \setminus U^s(X,n)$ have, respectively, the highest and

second highest possibility of *not* being an element of X. If $x \in \bigcap_{i=1}^{n} B_{s(Q_i)}(X)$, then we move to the $n+1$ level and again check if x is certain/possible positive or negative element of X of level $n+1$. As noted earlier, faith on the decision reduces as we move to the next level. For instance, if x is a possible positive element of X of level n and y is even a certain positive element of X of level $n+1$, we will consider x to have a greater possibility to be an element of X compared to y. Similarly, if x is a possible negative element of X of level n and y is a certain negative element of X of level $n+1$, we will take x to have a greater possibility to *not* be an element of X compared to y. Observe that if $Q_n = \emptyset$, there will be no certain/possible positive or negative element of level n. So, while descending the preference list, once we reach an empty Q_n, no more elements can be decided to be positive or negative. Thus given a $MSAS^P$ $\mathfrak{F} := (U, \{R_i\}_{i\in N}, \{Q_i\}_{i\in N})$ such that $Q_n \neq \emptyset$ and $Q_{n+1} = \emptyset$, on the basis of being an element of $X \subseteq U$, the universe is divided into $4n+1$ disjoint regions – the regions of being certain/possible positive and negative element of X of level m, $1 \leq m \leq n$, and the undecidable region $\bigcap_{i=1}^{n} B_{s(Q_i)}(X)$ of the $MSAS^P$ $\mathfrak{F}$. However, if $\mathfrak{F}$ is strict, there are no possible positive or negative elements of X of level m, $1 \leq m \leq n$, and thus in that case, the universe would be divided into $2n+1$ disjoint regions.

Example 1. Let us consider the $MSAS^P$ $\mathfrak{F} := (U, \{R_P\}_{P\subseteq N}, \{Q_i\}_{i\in N})$, where

- $N := \{1, 2, 3\}$ and $U := \{O_1, O_2, \ldots, O_5\}$,
- $U|R_1 := \{\{O_1, O_2\}, \{O_4\}, \{O_3\}, \{O_5\}\}$,
- $U|R_2 := \{\{O_1, O_4\}, \{O_2, O_3\}, \{O_5\}\}$,
- $U|R_3 := \{\{O_2\}, \{O_1, O_4\}, \{O_3, O_5\}\}$,
- $Q_1 := \{3\}$, $Q_2 := \{1, 2\}$ and $Q_i := \emptyset$ for $i > 2$.

Let $X := \{O_2, O_3, O_4\}$ of the domain U. Then, we obtain

- $\underline{X}_{R_1} = \{O_3, O_4\}$, $\underline{X}_{R_2} = \{O_2, O_3\}$, $\underline{X}_{R_3} = \{O_2\}$ and
- $\overline{X}_{R_1} = \{O_1, O_2, O_3, O_4\}$, $\overline{X}_{R_2} = \{O_1, O_2, O_3, O_4\}$, $\overline{X}_{R_3} = U$.

Therefore, the approximations given in Definition 4 are as follows.

- $L^s(X, 1) = L^w(X, 1) = \{O_2\}$,
- $U^s(X, 1) = U^w(X, 1) = U$,
- $L^s(X, 2) = \{O_3\}$, $L^w(X, 2) = \{O_3, O_4\}$,
- $U^s(X, 2) = U^w(X, 2) = \{O_1, O_2, O_3, O_4\}$.

Thus the information provided by $\mathfrak{F}$ results in the following division of the domain. O_2 is a certain positive element of X of level 1. O_3, O_4 are respectively certain and possible positive elements of X of level 1 and O_5 is a certain negative element of X of level 2. O_1 belongs to the undecidable region of the $MSAS^P$.

Remark 1. As mentioned in Section 1, in the notion of $MSAS$, there is a hidden assumption that no source is preferred over another, i.e. each source is equally preferred. Thus one can represent the $MSAS := (U, \{R_i\}_{i\in N})$ as the $MSAS^P$ $(U, \{R_i\}_{i\in N}, \{Q_i\}_{i\in N})$, where $Q_1 = N$ and $Q_i = \emptyset$ for $i > 1$. In that case,

$$L^s(X,1) = \underline{X}_{s(N)}, \quad L^w(X,1) = \underline{X}_{w(N)},$$
$$U^s(X,1) = \overline{X}_{s(N)}, \quad U^w(X,1) = \overline{X}_{w(N)}.$$

So when a $MSAS^P$ is of cardinality one, i.e. consists of a single relation, the notions of lower and upper approximations of level 1 given in Definition 4 just reduce to Pawlak's lower and upper approximations with respect to the relation of the $MSAS$. Also note that the division of the domain of a $MSAS$ on the basis of possibility of being an element of a set X given in Section 2 is obtained as a special case of the division of the domain of a $MSAS^P$ given above with $n = 1$. Moreover, the notions of certain/possible positive and negative elements of a set given in Definition 2 are obtained as a special case of Definition 5.

3.1 Notion of Approximations Involving Distributed Knowledge Base

Suppose we are given a $MSAS^P$ and it is the case that an object falls in the certain boundary of a set X for the group Q_1 of highest preferred sources. One option now is that we will use the knowledge base of other sources of the system following the method given in Section 3. But in that approach, the knowledge base of the most preferred sources is not used once we cross level one. This is indeed the case due to condition (P3), of Definition 3. One may like to have notions of approximations such that even when we cross level one, decision will still depend on the knowledge base of the most preferred sources. Such an approximation can be given using the distributed knowledge base of the sources [8]. The following definition proposes some notions of approximations involving the distributed knowledge base of a group of sources.

Let $\mathfrak{F} := (U, \{R_i\}_{i\in N}, \{Q_i\}_{i\in N})$ be a $MSAS^P$. Let us recall that for $P \subseteq N$, $R_P := \bigcap_{i\in P} R_i$ denotes the distributed knowledge base of the group P of sources.

Definition 6. *Let $X \subseteq U$ and $1 \leq n \leq |N|$. We define the lower approximation L^i and upper approximation U^i, $i = 1, 2, 3$ of X of level n as follows.*

- $L^i(X,1) = \underline{X}_{R_{Q_1}}$, $U^i(X,1) = \overline{X}_{R_{Q_1}}$, $i = 1, 2, 3$.

For $n > 1$

- $L^1(X,n) := \bigcap_{i=1}^{n-1} B_{R_{Q_i}}(X) \cap \underline{X}_{R_{Q_n}}$.
  $U^1(X,n) := \bigcup_{i=1}^{n-1} (B_{R_{Q_i}}(X))^c \cup \overline{X}_{R_{Q_n}}$.
- $L^2(X,n) := \bigcap_{i=1}^{n-1} B_{R_{Q_1\cup Q_i}}(X) \cap \underline{X}_{R_{Q_1\cup Q_n}}$.
  $U^2(X,n) := \bigcup_{i=1}^{n-1} (B_{R_{Q_1\cup Q_i}}(X))^c \cup \overline{X}_{R_{Q_1\cup Q_n}}$.
- $L^3(X,n) := B_{R_{\bigcup_{i=1}^{n-1} Q_i}}(X) \cap \underline{X}_{R_{\bigcup_{i=1}^{n} Q_i}}$.
  $U^3(X,n) := (B_{R_{\bigcup_{i=1}^{n-1} Q_i}}(X))^c \cup \overline{X}_{R_{\bigcup_{i=1}^{n} Q_i}}$.

Approximations L^3, U^3 are based on the idea that given an object x and a set X, we check if x is a positive or negative element of X with respect to the distributed knowledge base R_{Q_1}. If it is in the boundary region, then we consider the distributed knowledge base of the group $Q_1 \cup Q_2$ consisting of first

and second most preferred sources. We continue in this way and we stop our search once x falls in the decidable region. Then we put the weightage to the decision depending on how much we had to descend in the preference list to make the decision. Similarly one can interpret the other two types of approximations.

Observe that in the case of a *strict* $MSAS^P$, L^1, U^1 coincide with L^r and U^r respectively, $r \in \{w, s\}$. Also note that in the case of $MSAS$ $\mathfrak{F} := (U, \{R_i\}_{i \in N})$ viewed as a $MSAS^P$, we obtain $L^i(X,1) = \underline{X}_{R_N}$, $U^i(X,1) = \overline{X}_{R_N}$, $L^i(X,n) = \emptyset$, $U^i(X,n) = U$, for $n > 1, i \in \{1,2,3\}$.

The following proposition lists a few properties of these approximations.

Proposition 2.

1. *Let $i, j \in \{1,2,3\}$ such that $i < j$. Then for each n_1, n_2, $1 \leq n_1, n_2 \leq |N|$, there exists m_1, m_2 with $m_k \leq n_k$, $k = 1,2$ such that $L^i(X, n_1) \subseteq L^j(X, m_1)$ and $U^j(X, m_2) \subseteq U^j(X, m_1)$.*
2. *$L^k(X,n) \subseteq X \subseteq U^k(X,n)$, $k \in \{1,2,3\}$.*
3. *For $m \neq n$, $L^k(X,n) \cap L^k(X,m) = \emptyset$ and $U^k(X,n) \cap U^k(X,m) = \emptyset$.*

Given an $MSAS^P$ with $Q_n \neq \emptyset$ and $Q_{n+1} = \emptyset$ and a subset X of the domain, on the basis of possibility to be an element of X, each of the approximations given by Definition 6 divides the domain into $2n+1$ disjoint regions, namely $L^i(X,m)$, $(U^i(X,m))^c$, $1 \leq m \leq n$ and undecidable region $\bigcap_{k=1}^n B_k^i(X)$, $i = 1,2,3$, where $B_k^1(X) := B_{R_{Q_k}}(X)$, $B_k^2(X) := B_{R_{Q_1 \cup Q_k}}(X)$ and $B_k^3(X) := B_{R_{\cup_j^k Q_j}}(X)$.

4 $MSAS^P$ with Preference Lists Representing the View of Sources

The preference list in the definition of $MSAS^P$ could also be interpreted as the view of a particular source. In fact, this interpretation also leads to the generalization of $MSAS^P$ to a structure where we have preference lists corresponding to each source of the system.

Definition 7. *A (strict) $MSAS^P$* with source preference *($MSAS^{SP}$) is the tuple $\mathfrak{F} := (U, \{R_i\}_{i \in N}, \{Q_i^j\}_{j,i \in N})$, where for each $j \in N$, $(U, \{R_i\}_{i \in N}, \{Q_i^j\}_{i \in N})$ is a (strict) $MSAS^P$ and $Q_1^j := \{j\}$.*

For a fixed $j \in N$, $\{Q_i^j\}_{i \in N}$ gives the preference list of the source j. Thus the condition $Q_1^j := \{j\}$ signifies that each source prefers herself over all other sources of the system. The notions of approximations given so far can be defined in $MSAS^{SP}$ corresponding to each source of the system by using the preference list of the source. We add a subscript j to the notations of approximations to express that the approximation is with respect to the preference list of j.

Consider a strict $MSAS^{SP}$ $(U, \{R_i\}_{i \in N}, \{Q_i^j\}_{j,i \in N})$ and a source j. Let us see how the agent j makes a decision regarding the elements of a set X using approximations L_j^2 and U_j^2. Given an object x, if j fails to decide whether the object is an element of the set or not by using her own knowledge base, then she starts combining her knowledge base with the other sources of the system descending

her preference list. She stops her search when she gets a source combining her knowledge base with whom she is able to decide regarding the membership of the object. Depending on the position of the source in her preference list, she assigns weightage to her decision.

We end the article with the following example.

Example 2. Consider the $MSAS$ of Example 1 with the sources' preference lists:
Source-1: $Q_1^1 := \{1\}$, $Q_2^1 := \{2\}$, $Q_3^1 := \{3\}$;
Source-2: $Q_1^2 := \{2\}$, $Q_2^2 := \{1\}$, $Q_3^2 := \{3\}$;
Source-3: $Q_1^3 := \{3\}$, $Q_2^3 := \{2\}$, $Q_3^3 := \{1\}$.

Let $X := \{O_2, O_3, O_4\}$. As we are considering strict $MSAS^{SP}$, we have $L_j^s(X,n) = L_j^w(X,n)$ and $U_j^s(X,n) = U_j^w(X,n)$. One can verify that:
$L_1^s(X,1) = \{O_3, O_4\}$, $L_1^s(X,2) = \{O_2\}$, $L_1^s(X,3) = \emptyset$,
$U_1^s(X,1) = \{O_1, O_2, O_3, O_4\}$, $U_1^s(X,2) = U_1^s(X,3) = U$,
$L_2^s(X,1) = \{O_2, O_3\}$, $L_2^s(X,2) = \{O_4\}$, $L_2^s(X,3) = \emptyset$,
$U_2^s(X,1) = \{O_1, O_2, O_3, O_4\}$, $U_2^s(X,2) = U_2^s(X,3) = U$.

Thus source 1 considers O_5 and O_3, O_4 to be certain negative and certain positive elements of X of level 1. O_2 is a certain positive element of level 2 for source 1. Although source 2 also considers O_5 to be a certain negative element of X of level 1, it considers O_4 and O_2, O_3 to be certain positive elements of X of level 2 and 1 respectively. O_1 is an undecidable element for both the sources 1 and 2.

Next suppose source 3 wants to decide membership of elements with respect to X. As $\underline{X}_{R_3} = \{O_2\}$ and $\overline{X}_{R_3} = U$, she concludes that O_2 is a positive element of X, but she is unable to make any decision about the other objects. At this point, she may like to use one of the approximations defined in this article. For instance, suppose, she has decided to use approximations L_3^2, U_3^2. So, she combines her knowledge base with that of the source 2 as she prefers source 2 over source 1. Since $L_3^3(X,2) = \{O_3\}$, $U_3^3(X,2) = \{O_1, O_2, O_3, O_4\}$, she is able to decide that O_3 and O_5 are respectively positive and negative elements of X. But, as she had to use the knowledge base of second preferred source 2, she puts less possibility on O_3 to be an element of X compared to O_2. Moreover, she is still not able to make any decision about the objects O_1 and O_4. So, she will now have to use the knowledge base of source 1. There it is found that $O_4 \in L_3^3(X,3)$ and $O_1 \notin U(X,3)$ and so she concludes that O_4 and O_1 are respectively positive and negative elements. She puts the weightage on the decision accordingly.

5 Conclusion

In order to capture the situation where information from a source may be preferred over that of another source of the system for deciding membership of objects, the notion of multiple-source approximation system ($MSAS$) is extended to define the multiple-source approximation system with preference ($MSAS^P$). Notions of lower/upper approximations are proposed which depend on the knowledge base of the sources as well as on the positions of the sources in the hierarchy giving the preference of sources. It is observed that the notions of

weak/strong lower and upper approximations defined on $MSAS$ are obtained as a special case of these. It may be noted that the logic $LMSAS$ for $MSAS$ proposed in [2] can be extended to obtain a logic with semantics based on $MSAS^P$ and $MSAS^{SP}$, where one can express the notions of approximations defined here. But this issue is outside the scope of the current article.

Our investigation is restricted to the suitable notions of approximations for $MSAS^P$. One needs to investigate other standard rough set concepts such as definability of sets, membership functions. In this direction, one could think of generalizing the notions for $MSAS$ already studied in [2,3].

As mentioned in Section 1, there are many generalizations of Pawlak's rough set model. In the line of the current work, one may define multiple-source extensions based on these generalizations. We note that Pawlak's rough set model rules out contradictory knowledge base of the sources in the sense that a source considers an object to be a positive element of a set, but another source in the system considers the object to be a negative element of the same set. However, some generalized rough set models, such as covering and neighborhood based ones, would admit such a situation. This could be handled, for instance, by providing a *strict* preference list of the sources.

References

1. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. International Journal of General Systems 17, 191–200 (1990)
2. Khan, M.A., Banerjee, M.: Formal reasoning with rough sets in multiple-source approximation systems. International Journal of Approximate Reasoning 49(2), 466–477 (2008)
3. Khan, M.A., Banerjee, M.: Multiple-source approximation systems: membership functions and indiscernibility. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 80–87. Springer, Heidelberg (2008)
4. Lin, T.Y., Yao, Y.Y.: Neighborhoods system: measure, probability and belief functions. In: Proceedings of the 4th International Workshop on Rough Sets and Fuzzy Sets and Machine Discovery, November 1996, pp. 202–207 (1996)
5. Orłowska, E., Pawlak, Z.: Expressive power of knowledge representation systems. International Journal of Man-Machine Studies 20(5), 485–500 (1984)
6. Pawlak, Z.: Rough sets. International Journal of Computer and Information Science 11(5), 341–356 (1982)
7. Pomykała, J.A.: Approximation, similarity and rough constructions. ILLC prepublication series for computation and complexity theory CT-93-07, University of Amsterdam (1993)
8. Rauszer, C.M.: Rough logic for multiagent systems. In: Masuch, M., Polos, L. (eds.) Knowledge Representation and Reasoning under Uncertainty. LNCS (LNAI), vol. 808, pp. 161–181. Springer, Heidelberg (1994)
9. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
10. Ślęzak, D., Ziarko, W.: The investigation of the Bayesian rough set model. International Journal of Approximate Reasoning 40, 81–91 (2005)
11. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences 46, 39–59 (1993)

Classification of Dynamics in Rough Sets

Davide Ciucci

Dipartimento Di Informatica, Sistemistica e Comunicazione
Università di Milano – Bicocca, Viale Sarca 336 – U14, I–20126 Milano (Italia)
ciucci@disco.unimib.it

Abstract. A classification of the different dynamics which can arise in rough sets is given, starting from three different standpoints: information tables, approximation spaces and coverings. Dynamics is intended in two broad meanings: evolution in time and originated from different sources. Existing works on this topic are then categorized accordingly.

1 Introduction

It is a matter of fact that knowledge evolves in time (synchronic dynamics) and changes from a point of view to another (diachronic dynamics). Thus, even rough-set techniques are influenced by dynamics. Indeed, since the very beginning, attempts to deal with this issue have been carried out [8]. Then, during years there have been few works on these topics (for instance [13,14]). Recently, several authors try to deal with dynamics and multi source (or multi agent) in rough sets and they work in different directions: defining new approximations, new methods to update rules or giving a logical approach. Beyond a direct use of these new techniques, another chance of dynamics is to split a given problem in sub-problems and solve them in parallel, a major challenge for rough sets applicability. As we can read already in 1995: "It is our belief that only through the use of parallel processing will major progress and achievements be possible in real-world application of this data" [14]. Nowadays, we can assist to some works also in this direction [4].

The aim of the present paper is to classify and characterize different kinds of dynamic and point out the way in which they can be analysed. This can clarify the dynamics already under investigation and also highlight new forms of dynamics. Once made clear our framework, the known results about dynamics will be interpreted under this standpoint.

2 A Classification of Dynamics

The main distinction concerns the simultaneity or not of the events under investigation. These two branches are called by Pagliani [9] synchronic and diachronic dynamics. In the first case we are in presence of a multi-source situation: at a fixed moment in time we have different sources of information. These variety can depend mainly on two factors: different agents (humans or bots) which have

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 257–266, 2010.

some knowledge on the same concept or different sources of information. For example: two stockbrokers with different predictions, different web-sites about weather forecasts, etc.

In the case of diachronic dynamics, it is supposed that changes occur in time. These changes can regard different factors: new objects enter into the system under investigation (or also, existing objects that exit the system), new facts are taken into account, unknown facts that become known. Of course, several of these changes can appear simultaneously, for instance going from time t to time $t+1$, it may happen that n new objects enter the system and m new facts are considered. However, in this case, we can split the two events in two separate steps: from time t to $t+\frac{1}{2}$, n new objects enter the system and from time $t+\frac{1}{2}$ to time $t+1$, m new facts happen.

We note that also diachronic and synchronic changes can be mixed, that is we can have different sources of information which can evolve during time.

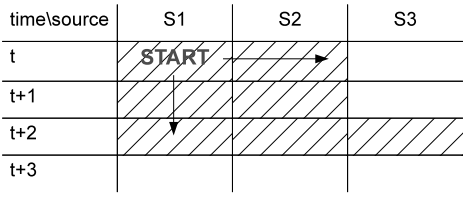


Fig. 1. A representation of synchronic and diachronic dynamics

In figure 1, these kinds of evolution are schematized. Starting at time t and source S1 we can move in two directions: horizontal to have synchronous dynamics and vertical to have asynchronous one. The dashed lines show a dynamic where at time t and t+1 we have two sources of information and at time t+2 a third source is added.

In generalized rough sets theory we have different ways to represent knowledge. Information Tables are the first developed and most used one and they represent the available knowledge in terms of attributes assumed by objects. Then, we can have Approximation Spaces, where the knowledge is available in terms of relations among objects. Finally, under the granular computing paradigm, knowledge may be represented in terms of granules generating a covering of the universe. We now analyse these three directions and classify the dynamics they give rise to, with respect to both synchronic and diachronic behaviour. In case of time evolution we point out the cases where there is an increase (decrease) of knowledge with respect to some fixed point of view.

2.1 Dynamic in Information Tables

Information Tables (or Information System) [10] are at the basis of rough sets. They have been defined to represent knowledge about objects in terms of observables (attributes). We deal with incomplete information tables, where some value can be missing.

Definition 1. *A (possibly incomplete)* Information Table *is a structure* $\mathcal{K}(X) = \langle X, A, val, F\rangle$ *where:*

- *the universe* X *is a non empty set of* objects*;*
- A *is a non empty set of* attributes*;*
- *val is the set of all* possible values *that can be observed for all attributes;*
- F *(called the* information map*) is a mapping* $F : X \times A \to val \cup \{*\}$ *which associates to any pair object* $x \in X$ *and attribute* $a \in A$*, the value* $F(x, a) \in$ *val assumed by a for the object* x*. If* $F(x, a) = *$ *it means that this particular value is unknown.*

Let us note that we do not deal with different semantics of incomplete information tables, but simply take into account that for some reason a value can be missing, i.e., $F(x, a) = *$.

In case of synchronic dynamics we assume that each source of information or agent is represented by a different information table. Thus, when comparing them we can have different degree of accordance, which we call *compatibility* among information tables.

Definition 2. *A set of information tables* $\mathcal{K}_i$ *is said to be:*

- fully compatible *if all the* $\mathcal{K}_i$ *are defined on the same set of objects, attributes and values, i.e., only the function* F_i *depends on the specific information table:* $\mathcal{K}_i = \langle X, Att(X), val(X), F_i\rangle$*;*
- value compatible *if the set of attributes and values is the same for all* $\mathcal{K}_i$*:* $\mathcal{K}_i = \langle X_i, Att(X), val(X), F_i\rangle$*;*
- attribute compatible *if the attributes are the same for all* $\mathcal{K}_i$*:* $\mathcal{K}_i = \langle X_i, Att(X), val_i(X), F_i\rangle$*;*
- object compatible *if the objects are the same for all* $\mathcal{K}_i$*:* $\mathcal{K}_i = \langle X, Att_i(X), val_i(X), F_i\rangle$.

Clearly, fully compatibility is equal to object plus value compatibility.

If we consider an information system evolving in time, it may change in terms of objects, attributes, values or information map. We want to characterize the situation where the knowledge increases as time passes. So, the information systems at time t and $t + 1$ must share some form of compatibility and either new objects enter the system or unknown values become known or new attributes be added. These different ways to increase the knowledge are formalized in the following way.

Definition 3. *Let* $\mathcal{K}^{(t_1)}(X) = \langle X_1, Att(X_1), val(X_1), F_1\rangle$ *and* $\mathcal{K}^{(t_2)}(X) = \langle X_2, Att_2(X_2), val_2(X_2), F_2\rangle$*, with* $t_1, t_2 \in \mathbb{R}$*,* $t_1 \leq t_2$ *be two information tables. We will say that there is a* monotonic increase of information *from time* t_1 *to time* t_2

- *wrt* values *iff* $\mathcal{K}^{(t_1)}$ *and* $\mathcal{K}^{(t_2)}$ *are fully compatible and* $F_1(x, a) \neq *$ *implies* $F_2(x, a) = F_1(x, a)$.

– *wrt* attributes *iff* $\mathcal{K}^{(t_1)}$ *and* $\mathcal{K}^{(t_2)}$ *are object compatible (i..e,* $X_1 = X_2$*) and* $Att_1(X_1) \subseteq Att_2(X_2)$, $val_1(X_1) \subseteq val_2(X_2)$ *and* $\forall a \in Att_1(X_1)$, $\forall x \in X_1, F_2(x,a) = F_1(x,a)$.

– *wrt* objects *iff* $\mathcal{K}^{(t_1)}$ *and* $\mathcal{K}^{(t_2)}$ *are value compatible,* $X_1 \subseteq X_2$ *and* $\forall x \in X_1$, $F_2(x,a) = F_1(x,a)$.

In all the three cases we can also define a *decrease of knowledge* when the reverse ordering holds.

Example 1. In Table 1 we can see a monotone increase of information wrt values from time t_0 to time t_1. Indeed, the only difference between time t_0 and time t_1 are the values F(Down-Town,f_3) and F(Furniture, f_4) which from missing become defined.

Table 1. Flats incomplete information systems

Observer at time t_0			
Flat	Price	Down-Town	Furniture
f_1	high	yes	*
f_2	high	yes	no
f_3	*	*	no
f_4	*	*	*

Observer at time t_1			
Flat	Price	Down-Town	Furniture
f_1	high	yes	*
f_2	high	yes	no
f_3	*	yes	no
f_4	*	*	yes

Observer at time t_2				
Flat	Price	Rooms	Down-Town	Furniture
f_1	high	2	yes	*
f_2	high	*	yes	no
f_3	*	2	yes	no
f_4	*	1	*	yes

On the other hand, from time t_1 to time t_2 we have a monotone increase of knowledge with respect to attributes, since the new attribute "Rooms" is added while the others do not change.

2.2 Dynamics in Approximation Spaces

Starting from an information table we usually define a binary relation on objects, which can be an equivalence, tolerance or also a more general relation, which is used to cluster objects in granules and define approximations. A different approach is to start directly from available relations, that is from a so called Approximation Space.

Definition 4. *An* Approximation Space *is a pair* $\mathcal{A} = (X, R)$ *with* X *a set of objects and* R *a binary relation on* X.

In this case, we have two sources of information which can vary: the set of objects X and the relation R. Thus, we can derive two notions of compatibility in approximation spaces with respect to synchronic dynamics.

Definition 5. *A set of approximation spaces $\mathcal{A}_i$ is said to be:*

- object compatible *if the objects are the same for all $\mathcal{A}_i$: $\mathcal{A}_i = \langle X, R_i \rangle$.*
- relation compatible *if the relations are defined in the same way on all common objects, i.e., let $Y = \bigcap X_i$, then for all $x, y \in Y$, $R_i(x, y)$ iff $R_j(x, y)$.*

In case of time evolution the increase of knowledge is defined as follows.

Definition 6. *Given two approximation spaces $\mathcal{A}^{(t_1)} = (X_1, R_1), \mathcal{A}^{(t_2)} = (X_2, R_2)$ with $t_1, t_2 \in \mathbb{R}$, $t_1 \leq t_2$, we have an* increase of knowledge in approximation spaces

- *wrt* objects *if $X_1 \subseteq X_2$ and $\mathcal{A}^{(t_1)}$ and $\mathcal{A}^{(t_2)}$ are relation compatible;*
- *wrt* relations *if $\mathcal{A}^{(t_1)}$ and $\mathcal{A}^{(t_2)}$ are object compatible and $R_1 \subseteq R_2$.*

That is, either we add new objects and new relations involving them without affecting the existing ones or we add new relations among the existing objects.

Example 2. Let us consider an approximation space at time t_0 represented by $X_0 = \{a, b, c, d\}, R_0 = \{(i, i), (b, c), (a, d)\}$, where i stands for any object, i.e., the relation is reflexive. Then, at time t_1 we have an increase of knowledge with respect to objects if the approximation space is updated as $X_1 = \{a, b, c, d, e\}, R_1 = \{(i, i), (b, c), (a, d), (d, e)\}$. That is, we added object e and the relations involving it. At time t_2 we have a monotone increase of knowledge with respect to relations if, for instance, the approximation space is $X_2 = \{a, b, c, d, e\}, R_2 = \{(i, i), (b, c), (a, d), (d, e), (b, d), (c, e)\}$. That is, objects are the same but new relations between b and d and between c and e are added.

2.3 Dynamics in Coverings

Finally, as said before, the binary relation definable in information systems or intrinsic in approximation spaces can be used to cluster objects. Thus, we can think to start our analysis directly on granules of objects which form a covering of the universe.

Definition 7. *Let X be a non empty set, a covering $\mathbb{C}(X)$ of X is a collection of sets $C_i \subseteq \mathcal{P}(X)$ such that $\bigcup C_i = X$.*

Also in this case, when considering knowledge coming from different sources, we can define two notions of compatibility.

Definition 8. *A collection of coverings $\mathbb{C}_i(X_i)$ is said to be:*

- object compatible *if the objects are the same for all $\mathbb{C}_i$: $\mathbb{C}_i(X)$*
- granule compatible *if for all common objects $Y = \bigcap X_i$, it happens that for all $x, y \in Y$, if x, y belong to the same set C_j for one covering $\mathbb{C}_i(X_i)$, then x, y belong to the same set in all coverings.*

Object compatibility is trivial, granules compatibility means that the objects common to all sources are classified in the same way.

In order to deal with increase of knowledge with respect to granularity, we need a notion of ordering among coverings. This is not so trivial, since different equivalent definitions of partition orderings, differ when generalized to coverings (see [1] for an overview). So, we assume to have a notion of ordering (or quasi-ordering) available, let call it $\preceq$, and define monotonicity with respect to this order as follows.

Definition 9. *Given two coverings* $\mathbb{C}^{(t_1)}(X_1), \mathbb{C}^{(t_2)}(X_2)$ *with* $t_1, t_2 \in \mathbb{R}$, $t_1 \leq t_2$, *we have an* increase of knowledge in coverings

- *wrt* objects *if* $X_1 \subseteq X_2$ *and* $\mathcal{C}^{(t_1)}$ *and* $\mathcal{C}^{(t_2)}$ *are granule compatible;*
- *wrt* granules *if* $\mathbb{C}^{(t_1)}$ *and* $\mathbb{C}^{(t_2)}$ *are object compatible and* $\mathbb{C}^{(t_1)}(X) \preceq \mathbb{C}^{(t_2)}(X)$.

Example 3. Consider the universe $X = \{a, b, c, d\}$ and the covering $\{\{a, d\}, \{b, c\}\}$. At a following time a new object e can enter the system and the new covering is $\{\{a, d\}, \{c, d\}, \{d, e\}, \{e\}\}$. That is we have an increase of knowledge in coverings wrt objects. Then, if the system is updated such that the new covering is $\{\{a, d\}, \{b, c, d\}, \{c, e\}, \{d, e\}, \{e\}\}$ we have an increase of knowledge with respect to granules if the following quasi ordering is considered:

$$\mathbb{C}_1(X) \preceq \mathbb{C}_2(X) \quad \text{iff} \quad \forall C_i \in \mathbb{C}_1(X)\ \exists D_j \in \mathbb{C}_2(X) \quad \text{such that} \quad C_i \subseteq D_j$$

2.4 Dependencies among the Three Approaches

Of course, information tables, approximation spaces and coverings are not independent tools. We want now to see how these dependencies reflect on dynamics.

Given an information table and a set of attributes $D \subseteq A$, two objects $x, y \in X$ are called *indiscernible* with respect to D, and we write xI_Dy, iff $\forall a \in D$, $F(a, x) = F(a, y)$. It can be easily verified that I_D is an equivalence relation and so it partitions the universe X in disjoint classes (granules) $E_D(x)$ defined as $E_D(x) := \{y \in X : xI_Dy\}$. Thus, for any set of attributes D, the pair $\langle X, I_D\rangle$ is an approximation space. About the relationship of compatibility in information tables and approximation spaces, we can trivially see that

- two information tables are object compatible iff the corresponding approximation spaces are object compatible.

Moreover, there is no correspondence between other notions of compatibility. Indeed, suppose for instance that two information tables generate two relation compatible approximation spaces. Nothing assures that the attributes do not change from one information table to another, nor the values, nor the information map. Vice versa, fully (value, attribute) compatibility can result in non compatible relations since F_i changes.

About the time evolution, we can say that

- there is a monotone increase of information wrt objects in an information table iff there is also in the induced approximation space.

Of course, this approach can be generalized by defining on objects a general (i.e., not necessarily equivalence) binary relation.

Now, given an approximation space (X, R), we can define the *successor* and *predecessor neighborhood* [15] of an element x respectively as $N_s(x) := \{y|xRy\}$ and $N_p(x) := \{y|yRx\}$. Under the (sufficient) condition that R is reflexive, the collection of all neighborhood is a covering of X. About the relationship between compatibility in approximation spaces and in covering we can say that

- object compatibility in approximation spaces induces object compatibility in coverings;
- relation compatibility in approximation spaces induces granule compatibility in coverings.

Similarly, when considering diachronic dynamics we have that

- an increase of knowledge wrt objects in approximation spaces induces an increase of knowledge wrt objects in coverings;

Since the increase of knowledge wrt to granules depends on a given ordering, it is not so immediate to give a general result. However, looking at example 3, it can be obtained by the approximation spaces in example 2 using the successor neighborhood. Thus, in this case, an increase of knowledge wrt relations in approximation spaces induces an increase of knowledge wrt granules in coverings.

3 Analysis of Dynamics

The most important instruments of rough sets are (lower and upper) approximations, reducts and rules. Thus, it is fundamental to understand how they are involved in dynamics. That is, how to put together approximations and rules coming from different sources, how the quality of a rule evolves in time, and so on. Another important and widely studied aspect is how to represent and manipulate knowledge from a formal-logical standpoint. Also this has to be (and in effect it is) investigated in dynamic environments.

As a consequence we can envisage four main streamlines to investigate:

1. Lower and upper approximations. Of course this aspect involves also boundary and exterior region;
2. Reducts and rules;
3. Quality indexes, both of approximations and rules;
4. Formal logic, that is how to represent and manipulate dynamics form a formal language point of view.

Clearly, the analysis varies according to the kind of dynamics (synchronic or diachronic) we are dealing with. In asynchronous dynamics, the main issue to face is how these topics change in time. So, questions which can arise are: are the new approximations closer to the set to approximate than the old ones? Are the reducts simpler than the older ones? Do the new rules perform better or not?

In synchronous dynamics, we have to deal with how the "opinion" of the different agents can be put together. This can lead to the definition of new kind of constructs (as was done for approximations in [6]) which account for the fact that different agents can partially agree, completely agree or do not agree at all. Just to give an example, let us think to three information tables (for the sake of simplicity, fully compatible) producing each three rules with the same antecedent. Now, it can happen that the three rules return the same decision (totally agree), only two are equal (partially agree) or they return three different results (totally disagree). Finally, in case of quality-indexes analysis, some overall measures (such as computing the average) should be defined.

3.1 Survey of Known Results

The dynamic topic in rough sets has been touched in several paper, more or less explicitly. Here we relate them to our framework and in the following table a summary of this classification is outlined.

	Approximations	Rules	Indexes	Logic
Synchronic	[13] [6] [12] [11]	[12] [11]		[13] [6]
Diachronic	[3]	[14] [2] [5]	[7] [5]	[8]

In time order and to the best of our knowledge, the first work about dynamics in rough sets is [8]. Here, by dynamic information system is intended a collection of *fully compatible information systems* where the information map depends on a parameter t interpreted as time and thus leading to an asynchronous dynamics. The logic DIL is introduced as the "linguistic counterpart" of these information systems. Clearly, if we interpret the parameter t as an agent, we can have a multi-source information system. Indeed, following this line [13] studies the ways of interaction of different agents whose knowledge is given by different partitions. In particular, the notion of "weak common knowledge" defines a new partition which is *object and granule compatible* with the existing starting ones. Properties of approximations in new partitions with respect to the existing one are studied and a multi-modal logic is introduced to model different agents. Also [6] tackles this problem and defines multiple-source approximation systems based on the same set of objects but on different equivalence relations and introduces a formal logic to model them. This approximation system can be viewed as a collection of *object compatible approximation spaces* where all the relations are equivalence ones. In order to take into account the different relations, two lower and two upper approximations are considered, which result from the intersection or union of the approximations of the relations taken separately. Some results about these approximations are proved. Further a definition of Knowledge representation system is given which corresponds to a collection of *object compatible information systems*.

Finally, on the synchronic side, we have [12] which is not properly intended for dynamics but can be interpreted in this sense. Indeed, the authors consider

one complete information system and use different subsets of attributes to generate different partitions of the universe. Thus, we can see this approach as a collection of *object compatible information systems* which have some common attributes with identical values on all objects. A new kind of approximation is introduced where instead of the AND of single attributes, OR is considered. Also "approximate reduct" is defined as the smallest collection of attributes which preserves the approximations in all the information systems. A similar study is done in [11] wrt incomplete information systems and tolerance relations.

When considering the diachronic dynamics we can see that [14] and [2] studied in different situations how to update rules during time. In [14] one new object is added to an information system and the rules are updated. This situation can be viewed as two value-compatible information tables with a *monotone increase of information wrt objects*. Also in [2] we have a *monotone increase of information wrt objects* since a new object is added to the system. However here an incremental induction of rules is proposed in the Dominance-based Rough Set Approach (DRSA) with missing values.

Time evolution and rules are also treated in [5] where *a monotonic decrease of knowledge wrt values* is studied: at each time step, some values are considered as missing (here, with three different semantics) and the resulting rules are then tested for performance. The result is that the rules obtained in the case of less information (more missing values) have better performances.

Evolution of indexes in time are also analyzed in [7] where *a monotonic increase and decrease of knowledge wrt objects in information tables* is considered. The coverage and accuracy measure of the rules at time t are then updated with the new event occured at time $t + 1$.

Finally, in [3] we studied the evolution of approximations in tolerance and preclusive rough sets in presence of a *monotonic increase of information wrt values* and *wrt attributes* in information tables. The result is that approximations become better when acquiring new knowledge.

4 Conclusion

A classification of dynamics has been presented both in the synchronic and diachronic case. Three directions have been followed as starting point of the analysis: information tables, approximation systems and coverings. Since these approaches are not independent, their relationship has been analyzed. Finally, the existing works on dynamics have been classified according to our framework. This survey is not exhaustive since for lack of space it does not give the details for all the three starting points (information tables, approximation spaces and coverings) nor it pretends to suggest which approach is better than others (note also that often the approaches are not comparable since they solve different problems). Nevertheless, the recent studies [6], [11,12], [7] seem promising in the three different aspects of logic, rules and indexes. Further, we can see that some problems have not been investigated yet. For instance, if one compares all the classifications of section 2 with the existing works it can be seen that there are

no works which start directly from a covering (which is not a partition as in [13]). Finally, we note that even if the two dynamics (synchronic and diachronic) are often studied separately, it seems that the results of one field can be translated into the other.

References

1. Bianucci, D., Cattaneo, G.: Information entropy and granulation co-entropy of partitions and coverings: A summary. In: Peters, J.F., Skowron, A., Wolski, M., Chakraborty, M.K., Wu, W.-Z. (eds.) Transactions on Rough Sets X. LNCS, vol. 5656, pp. 15–66. Springer, Heidelberg (2009)
2. Blaszczynski, J., Slowinski, R.: Incremental induction of decision rules from dominance-based rough approximations. Electr. Notes Theor. Comput. Sci. 82(4), 40–51 (2003)
3. 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)
4. Geng, Z., Zhu, Q.: A multi-agent method for parallel mining based on rough sets. In: Proceedings of the 6th World Congress on Intelligent Control and Automation, pp. 5977–5980 (2006)
5. Grzymala-Busse, J., Grzymala-Busse, W.: Inducing better rule sets by adding missing attribute values. In: Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) RSCTC 2008. LNCS (LNAI), vol. 5306, pp. 160–169. Springer, Heidelberg (2008)
6. Khan, M., Banerjee, M.: Formal reasoning with rough sets in multiple-source approximation systems. International Journal of Approximate Reasoning 49, 466–477 (2008)
7. Liu, D., Li, T., Ruan, D., Zou, W.: An incremental approach for inducing knowledge from dynamic information systems. Fundamenta Informaticae 94, 245–260 (2009)
8. Orlowska, E.: Dynamic information systems. Fundamenta Informaticae 5, 101–118 (1982)
9. Pagliani, P.: Pretopologies and dynamic spaces. Fundamenta Informaticae 59(2-3), 221–239 (2004)
10. Pawlak, Z.: Information systems - theoretical foundations. Information Systems 6, 205–218 (1981)
11. Qian, Y., Liang, J., Dang, C.: Incomplete multigranulation rough set. IEEE Transactions on Systems, Man and Cybernetics - PART A 40, 420–431 (2010)
12. Qian, Y., Liang, J., Yao, Y., Dang, C.: MGRS: A multi-granulation rough set. Information Sciences 180, 949–970 (2010)
13. Rauszer, C.: Rough logic for multi-agent systems. In: Logic at Work 1992. LNCS (LNAI), vol. 808, pp. 161–181. Springer, Heidelberg (1994)
14. Shan, N., Ziarko, W.: Data-based acquisition and incremental modification of classification. Computational Intelligence 11, 357–370 (1995)
15. Yao, Y.: Relational interpretations of neighborhood operators and rough set approximation operators. Information Sciences 111, 239–259 (1998)

Relational Granularity for Hypergraphs

John G. Stell*

Schoool of Computing, University of Leeds, LS2 9JT, U.K.
j.g.stell@leeds.ac.uk

Abstract. A set of subsets of a set may be seen as granules that allow arbitrary subsets to be approximated in terms of these granules. In the simplest case of rough set theory, the set of granules is required to partition the underlying set, but granulations based on relations more general than equivalence relations are well-known within rough set theory. The operations of dilation and erosion from mathematical morphology, together with their converse forms, can be used to organize different techniques of granular approximation for subsets of a set with respect to an arbitrary relation. The extension of this approach to granulations of sets with structure is examined here for the case of hypergraphs. A novel notion of relation on a hypergraph is presented, and the application of these relations to a theory of granularity for hypergraphs is discussed.

1 Introduction

The theory of rough sets [Paw82, PS07] provides, in its most basic form, a way of approximating arbitrary subsets of a fixed universal set U in terms of the equivalence classes of an equivalence relation on U. These equivalence classes can be thought of as granules which represent a coarser, or less detailed, view of U in which we cannot detect individual elements – all we can see are the granules. This initial starting point of the theory has been extended, [SS96, Lin98], to more general relations on U, including the case of an arbitrary binary relation [Yao98, Zhu07]. For a relation R on U, the granules are the neighbourhoods, that is subsets of the form $R(x) = \{y \in U \mid x \; R \; y\}$. More generally still, a binary relation between two sets can be used, and the numerous links that this reveals between rough sets and other topics including formal concept analysis, Chu spaces, modal logic, and formal topology are discussed in detail by Pagliani and Chakraborty [PC08] in their monograph on rough set theory. The general importance of granularity in information processing has been discussed by numerous authors including [Zad97, Yao01].

Rough set theory represents a substantial body of knowledge which encompasses both practical techniques for data analysis and theoretical results in logic and algebra. As its name indicates, the fundamental concern of the theory is with sets, that is with collections of entities having no additional structure. However, granularity presents significant challenges in other contexts, and the focus in

* Supported by EPSRC (EP/F036019/1) and Ordnance Survey project *Ontological Granularity for Dynamic Geo-Networks*.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 267–276, 2010.

this paper is on the case that we do not merely have a set of entities where we need to approximate subsets, but where we have a graph (or more generally a hypergraph) and we need to give approximate descriptions of subgraphs. We can regard this as an extension of rough set theory since a set is just a special kind of graph in which there are only nodes (or vertices) and no edges (or arcs). For a simple example we can consider a graph as modelling a railway network and we can imagine a process of granulation in which we take a less-detailed view of the network. The need for such a granulation might arise from incomplete knowledge of some event affecting the network, such as an accident or a terrorist incident. It might also arise from the particular needs of users: a passenger requires a view of the network which is different from that of an engineer working to maintain part of the system. Hypergraphs generalize graphs by allowing edges that may be incident with arbitrary sets of edges rather than with just one or two edges. In giving a granular view of a railway network we might need to indicate that it is possible to travel between any two stations in some set without specifying exactly what station-to-station links exist. This kind of scenario is one reason why it is appropriate to consider hypergraphs and not just graphs.

In order to understand what rough hypergraph theory might be, this paper considers how a relation on a hypergraph can be used to define approximation operators which generalize the operators defined for a relation on a set by Yao [Yao98]. It turns out that the approximation operators defined by Yao can be related to well-known constructions in mathematical morphology, and this can be used as a way of generalizing the operators to ones on hypergraphs. Mathematical morphology [Ser82] originated in image processing but the most basic aspects of the body of techniques it provides (erosions, dilations, openings and closings) can all be presented [BHR07] in terms of binary relations. Although connections between rough sets and mathematical morphology have studied [Blo00, Ste07], there appears to be potential for this topic to contribute further to a general understanding of granularity.

It is not immediately clear what we should mean by a relation on a hypergraph. One possibility would be two separate relations, one for edges and one for nodes, subject to some compatibility condition. The disadvantage of adopting this approach is that we find such relations do not correspond to the sup-preserving operations on the lattice of sub-hypergraphs. This is significant, because the well-known fact that relations on a set, U, are equivalent to sup-preserving operations on the powerset $\mathscr{P}U$ is an essential ingredient in mathematical morphology. If we are to take advantage of the way mathematical morphology provides operations for granular approximation, we need the appropriate definition of hypergraph relations.

Section 2 describes how existing rough set approximation operators can be described using morphological operators. Hypergraphs are introduced formally in Section 3, where it emerges that the hypergraph relations we need must allow edges to be related to nodes as well as to edges, and dually nodes may be related to edges as well as to nodes. The main technical challenge solved by the paper concerns the converse of a hypergraph relation. Although our relations can be

modelled as sets of arrows, simply reversing the direction of the arrows fails to give a valid relation. Section 4 shows how to construct the converse and shows that it performs the same role with respect to sub-hypergraphs as the converse of a usual relation does with respect to subsets. Limitations of space mean that proofs have been omitted, but it is hoped that that the inclusion of examples of the approximation operators obtained in Section 5 will allow the reader to appreciate the main features of the ideas introduced. Conclusions and further work appear in Section 6.

2 Approximation Operators

The purpose of this section is to recall the six approximation operators described in [Yao98] and to relate them to operators from mathematical morphology. This will be used later as the means of seeing how to generalize these operators when we have a relation on a hypergraph rather than a relation on a set.

Suppose we have a set U and a subset $A \subseteq U$. To give a granular, or less detailed, description of A is to describe A not in terms of the elements it contains, but in terms of certain subsets of U called granules. These granules can be thought of as arising from some notion of indistinguishability on the elements of U. From this viewpoint, a granule clumps together elements of U that are not distinguished from each other. Granules often arise from a binary relation on U.

Definition 1. *Let R be a relation on U, then the* ***granules*** *(with respect to R) are the subsets $R(x) = \{y \in U \mid x\ R\ y\}$ where $x \in U$.*

When there are no restrictions on R, an element of U may belong to many granules or none. Given a relation R, each subset $A \subseteq U$ can be described in terms of the granules. These arise from two ways in which a set of elements gives rise to a set of granules, and two ways in which a set of granules gives rise to a set of elements. From a set of elements $A \subseteq U$ we can take the granules that intersect A, or the granules that are subsets of A. From a set of granules G we can take the elements where at least one of their granules is present in G, or we can take the elements all of whose granules lie in G. These possibilities yield four approximations to A, and I use the notation for these used in [Yao98]. If the relation R is not clear from the context, we can write $\underline{apr}'_R(A)$ etc.

$$\begin{aligned}
\underline{apr}'(A) &= \{x \in U \mid \exists y \in U(x \in R(y) \wedge R(y) \subseteq A)\} \\
\underline{apr}''(A) &= \{x \in U \mid \forall y \in U(x \in R(y) \Rightarrow R(y) \subseteq A)\} \\
\overline{apr}'(A) &= \{x \in U \mid \forall y \in U(x \in R(y) \Rightarrow R(y) \cap A \neq \varnothing)\} \\
\overline{apr}''(A) &= \{x \in U \mid \exists y \in U(x \in R(y) \wedge R(y) \cap A \neq \varnothing)\}
\end{aligned}$$

In addition to these four operators, there are two further ones [Yao98, p246]:

$$\begin{aligned}
\underline{apr}(A) &= \{x \in U \mid \forall y \in U(y \in R(x) \Rightarrow y \in A)\}, \\
\overline{apr}(A) &= \{x \in U \mid \exists y \in U(y \in R(x) \wedge y \in A)\}.
\end{aligned}$$

These six operators can be represented in terms of dilations and erosions as used in mathematical morphology. The relation $R : U \to U$ has an associated function, known as a dilation, $\_ \oplus R : \mathscr{P}U \to \mathscr{P}U$ defined by $A \oplus R = \{x \in U \mid \exists y \in U(y\ R\ x \wedge y \in A)\}$. The dilation is a sup-preserving a mapping between complete lattices, so has a right adjoint $R \ominus \_ : \mathscr{P}U \to \mathscr{P}U$ which can be described by $R \ominus A = \{x \in U \mid \forall y \in U(x\ R\ y \Rightarrow y \in A)\}$.

It is necessary here to assume some knowledge of adjunctions on posets (or Galois connections), but details can be found in [Tay99]. The notation $f \dashv g$ will be used when f is left adjoint to g, and the idea [Tay99, p152], of viewing $\dashv$ as an arrow (with the horizontal dash as the shaft of the arrow, and the vertical dash as the head of the arrow) proceeding from the left adjoint to the right adjoint is also adopted. This leads to diagrams of the form $\overrightarrow{\ \perp\ }$ and various rotated forms in Section 4 below.

Although writing dilations on the right and erosions on the left is contrary to established practice in mathematical morphology, it is adopted here since if we have a second relation S then using ; to denote composition in the 'diagrammatic' order $(R\,;S) \ominus A = R \ominus (S \ominus A)$ and $A \oplus (R\,;S) = (A \oplus R) \oplus S$. The relation R has a converse R^{-1} and dilation and erosion by R^{-1} yield operations (note the sides on which these act) $R \oplus^{-1} \_ : \mathscr{P}U \to \mathscr{P}U$ and $\_ \ominus^{-1} R : \mathscr{P}U \to \mathscr{P}U$.

The following result is stated without proof, as it follows by routine techniques. The approximations $\underline{apr}'$ and $\overline{apr}''$ are particularly well-known in mathematical morphology as the opening and as the closing by the converse.

Theorem 1. *For any relation R on U and any $A \subseteq U$,*

$$\begin{aligned} \underline{apr}(A) &= R \ominus A, & \overline{apr}(A) &= R \oplus^{-1} A, \\ \underline{apr}'(A) &= (R \ominus A) \oplus R, & \overline{apr}'(A) &= (R \oplus^{-1} A) \ominus^{-1} R, \\ \underline{apr}''(A) &= (R \ominus A) \ominus^{-1} R, & \overline{apr}''(A) &= (R \oplus^{-1} A) \oplus R. \end{aligned}$$

3 Relations on Hypergraphs

A hypergraph [Ber89] is a generalization of the concept of undirected graph in which an edge (or rather a 'hyperedge') may be incident with more than two nodes. As with graphs there are many variants of the basic idea. In the present work hypergraphs are permitted to have two distinct hyperedges incident with the same set of nodes, and hyperedges incident with an empty set of nodes are also allowed. Formally, a hypergraph consists of two sets E (the hyperedges) and N (the nodes) together with an arbitrary function from E to the powerset $\mathscr{P}N$.

A hypergraph may be drawn as in Figure 1 with each hyperedge as a closed curve containing the nodes with which it is incident. The example includes a hyperedge, f, incident with no nodes, and a node, z, to which no hyperedges are incident. The hyperedge e is incident with exactly one node, a situation that would correspond to a loop on the node y in an ordinary graph.

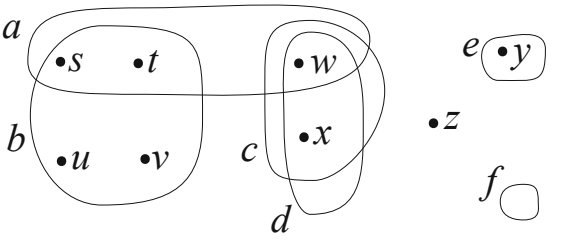


Fig. 1. Example of a hypergraph with hyperedges $\{a, b, c, d, e, f\}$ and nodes $\{s, t, u, v, w, x, y, z\}$

The idea of a hypergraph as having two disjoint sets of hyperedges and nodes is useful, but it turns out to be not the most appropriate for our purposes. Instead we need a definition based on the approach to graphs found in [BMSW06] and used in [Ste07]. This means we have a single set of elements comprising both edges and nodes and a relation associating nodes to themselves and edges to their incident edges.

Definition 2. *A* ***hypergraph*** (H, φ) *is a set* H *together with a relation* $\varphi : H \to H$ *such that for all* $x, y, z \in H$ *if* $x \varphi y$ *then* $y \varphi z$ *if and only if* $y = z$.

Figure 2 shows the same hypergraph as in Figure 1 visualized as a binary relation.

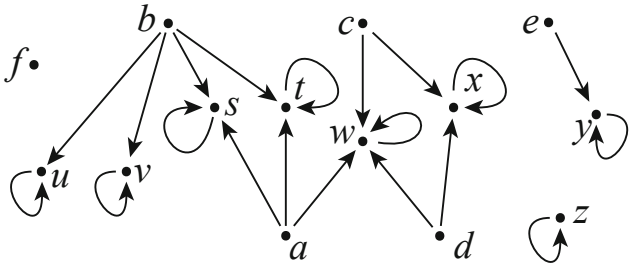


Fig. 2. The hypergraph in Figure 1 as a binary relation

Moving to consider relations on hypergraphs, we start with the definition of sub-hypergraph, which is essentially the requirement that whenever a hyperedge is present then all nodes with which it is incident are present too.

Definition 3. *A* ***sub-hypergraph*** *of* (H, φ) *is a subset* $K \subseteq H$ *such that* $K \oplus \varphi \subseteq K$.

It may be checked that the sub-hypergraphs form a complete lattice which is a sub-lattice of the powerset $\mathscr{P}H$. This lattice of sub-hypergraphs will be denoted

$\mathscr{L}\varphi$. The inclusion of $\mathscr{L}\varphi$ in $\mathscr{P}H$ has both left and right adjoints constructed as in the following definition.

Definition 4. *Let $A \subseteq H$, then we define the two ways to make A into a hypergraph* $\uparrow A = \bigcap\{K \in \mathscr{L}\varphi \mid A \subseteq K\}$ *and* $\downarrow A = \bigcup\{K \in \mathscr{L}\varphi \mid K \subseteq A\}$.

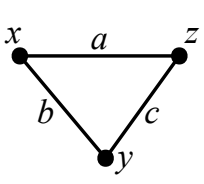
Given a hypergraph (H, φ), let I_φ be the relation $I_H \cup \varphi$ on the set H, where I_H denotes the identity relation on the set H.

Definition 5. *A **hypergraph relation** on (H, φ) is a relation R on the set H for which $R = I_\varphi \,;\, R \,;\, I_\varphi$.*

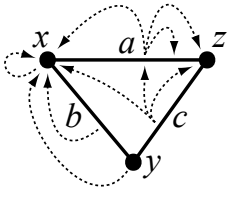
These relations play the same role with respect to the lattice $\mathscr{L}\varphi$ as the ordinary relations on the set H do with respect to the lattice $\mathscr{P}H$.

Theorem 2. *The hypergraph relations on (H, φ) form a quantale [Ros90] under composition of relations (with unit I_φ) which is isomorphic to the quantale of sup-preserving mappings on $\mathscr{L}\varphi$.*

An example of a hypergraph relation is shown in figure 3. In this example the hypergraph is actually a graph. The relation is shown by the dashed lines.



Graph G with edges $\{a, b, c\}$ and nodes $\{x, y, z\}$.



Relation R on G. $R = \{(a,a),(a,x),(a,z),(b,x),(c,a),(c,x),(c,z),(x,x),(y,x)\}$

Fig. 3. A Relation on a graph with three nodes and three edges

4 Converse for Hypergraph Relations

When R is a hypergraph relation, R^{-1} (the converse in the usual sense) need not be a hypergraph relation. Converse relations appear in the the approximation operators described in Theorem 1 and the notion of equivalence relation depends on symmetry and thus on the converse. To generalize these concepts to hypergraph relations requires that we can construct converses.

First recall one way of characterizing the converse of a relation on a set. Consider the set relation R as sup-preserving mapping $R : \mathscr{P}H \to \mathscr{P}H$ with right adjoint Σ. The converse can be obtained by defining $R^{-1}(A) = -(\Sigma(-A))$ where $-$ is the set-theoretic complement. This situation is summarised in the

diagram on the left of Figure 4. In the diagram the powerset $\mathscr{P}H$ is distinguished from from its opposite, $(\mathscr{P}H)^{\mathrm{op}}$ which has the same elements but with the reversed partial order. The mapping $(R^{-1})^{\mathrm{op}}$ has the identical effect on elements as R^{-1} but the distinction is important for the adjoints.

To generalize the notion of converse to hypergraph relations we replace the complement operation in $\mathscr{P}H$ by the corresponding construction for $\mathscr{L}\varphi$. The lattice $\mathscr{L}\varphi$ is not in general complemented, but there are two weaker operations.

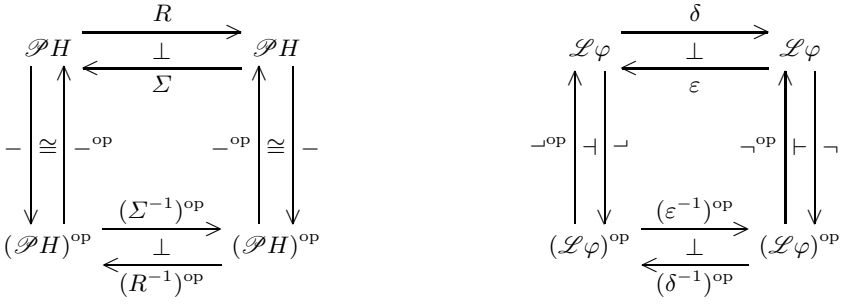
Definition 6. *Let $K \in \mathscr{L}\varphi$. Then the pseudocomplement $\neg K$ and the dual pseudocomplement $\multimap K$ are given by $\neg K = \downarrow(-K)$ and $\multimap K = \uparrow(-K)$.*

The complement operation $-$ provides an isomorphism between $\mathscr{P}H$ and its opposite. The pseudocomplement and its dual are not in general isomorphisms, but they do satisfy the weaker property of being adjoint to their opposites. That is, for $\neg, \multimap : \mathscr{L}\varphi \to (\mathscr{L}\varphi)^{\mathrm{op}}$, we have $\neg \dashv \neg^{\mathrm{op}}$ and $\multimap^{\mathrm{op}} \dashv \multimap$.

We now come to the definition of the converse of a hypergraph relation. For a relation R the notation R^c is used since R will also have a distinct converse, i.e. R^{-1}, as a relation on the set H.

Definition 7. *Let R be a hypergraph relation on (H, φ) and $\delta : \mathscr{L}\varphi \to \mathscr{L}\varphi$ its corresponding dilation. Then the converse of R is the hypergraph relation R^c corresponding to $\delta^c : \mathscr{L}\varphi \to \mathscr{L}\varphi$ where $\delta^c(K) = \multimap\varepsilon\neg(K)$ and $\delta \dashv \varepsilon$.*

The situation is summarised in the diagram on the right of Figure 4. The next theorem gives a practical means of computing converses as the composition $I_\varphi \,;\, R^{-1} \,;\, I_\varphi$ is more easily calculated than the expression given in Definition 7.



The set case

The hypergraph case

Fig. 4. Converse via complementation and adjoints

Theorem 3. *For any hypergraph relation R with associated dilation $\delta : \mathscr{L}\varphi \to \mathscr{L}\varphi$ the converse dilation satisfies $\delta^c K = I_\varphi(R^{-1}K)$ for every subgraph K, and the hypergraph relation representing δ^c is $I_\varphi \,;\, R^{-1} \,;\, I_\varphi$.*

5 Examples

The six approximation operators for subsets summarized in section 2 above can now be generalized to operators on sub-hypergraphs by interpreting the descriptions in Theorem 1 using dilations and erosions on the lattice $\mathscr{L}\varphi$ in place of the powerset lattice, and the construction of Theorem 3 for the converse. Examples of these approximations are given in Figures 5, 6, and 7.

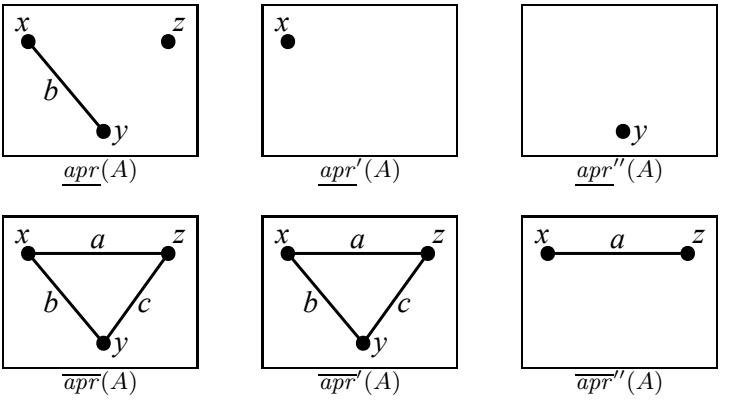
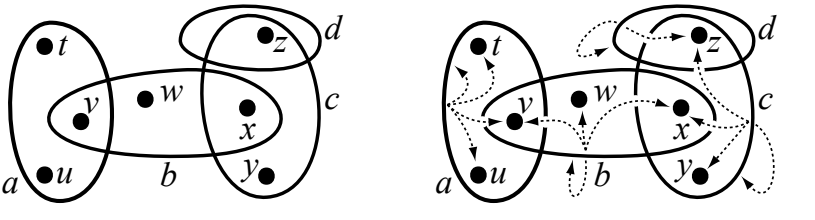


Fig. 5. Approximations of the subgraph $A = \{b, c, x, y, z\}$ of G using relation R from Figure 3



Hypergraph K with edges $\{a, b, c, d\}$ and nodes $\{t, u, v, w, x, y, z\}$

Relation $R = \{(a, a), (a, t), (a, v), (a, u), (b, b), (b, v), (b, w), (b, x), (c, c), (c, x), (c, y), (c, z), (d, d), (d, z)\}$

Fig. 6. Second example of a hypergraph relation

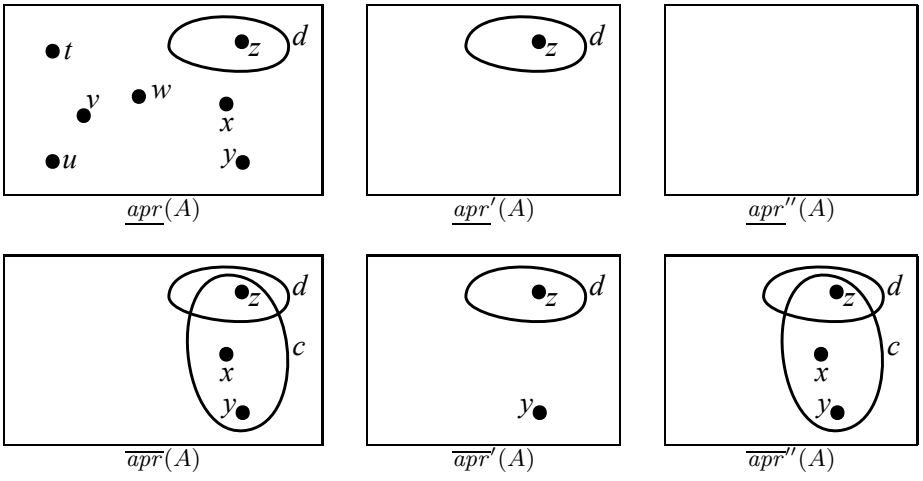


Fig. 7. Approximations of the subgraph $A = \{d, z\}$ under the relation R from Figure 6

6 Conclusions and Further Work

This paper has presented a novel approach to granularity for hypergraphs using a view of mathematical morphology as a theory of granularity in order to generalize six approximation operators from sets to hypergraphs. To define these operators on hypergraphs it was necessary to establish appropriate definitions for relations on hypergraphs and for the converse of a relation on a hypergraph. The definition of hypergraph relation has been justified by its equivalence to the notion of sup-preserving mapping on the lattice of sub-hypergraphs. The principal technical achievement in the paper has been the description of the converse of a hypergraph relation.

This work provides a starting point from which it should be possible to develop a full account of rough graphs and hypergraphs which generalizes the existing theory of rough sets. While the six kinds of approximation can all be applied to hypergraphs now that we have established appropriate generalizations of converse dilations and erosions, the properties of these constructions are not necessarily the same as in the set case. The study of these constructions thus presents one direction for further work. Other areas include extending the analysis using a relation on a single hypergraph to relations between distinct hypergraphs, and an investigation of an analogue of equivalence relations on hypergraphs. This latter issue is not straightforward as the notion of symmetry for hypergraph relations appears to have very weak properties related to the properties of the converse operation – in general $(R^c)^c \neq R$ unlike the familiar $(R^{-1})^{-1} = R$.

Acknowledgements

I am grateful to three anonymous reviewers for their helpful comments, and in one particular case for much detailed advice about the existing literature.

References

[Ber89] Berge, C.: Hypergraphs: Combinatorics of Finite Sets. North-Holland Mathematical Library, vol. 45. North-Holland, Amsterdam (1989)

[BHR07] Bloch, I., Heijmans, H.J.A.M., Ronse, C.: Mathematical morphology. In: Aiello, M., Pratt-Hartmann, I., van Benthem, J. (eds.) Handbook of Spatial Logics, ch. 14, pp. 857–944. Springer, Heidelberg (2007)

[Blo00] Bloch, I.: On links between mathematical morphology and rough sets. Pattern Recognition 33, 1487–1496 (2000)

[BMSW06] Brown, R., Morris, I., Shrimpton, J., Wensley, C.D.: Graphs of Morphisms of Graphs. Bangor Mathematics Preprint 06.04, Mathematics Department, University of Wales, Bangor (2006)

[Lin98] Lin, T.Y.: Granular computing on binary relations I, II. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery, pp. 107–140. Physica-Verlag, Heidelberg (1998)

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

[PC08] Pagliani, P., Chakraborty, M.: A Geometry of Approximation. Rough Set Theory: Logic, Algebra and Topology of Conceptual Patterns. Trends in Logic, vol. 27. Springer, Heidelberg (2008)

[PS07] Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177, 3–27 (2007)

[Ros90] Rosenthal, K.I.: Quantales and their applications. Pitman Research Notes in Mathematics, vol. 234. Longman, Harlow (1990)

[Ser82] Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, London (1982)

[SS96] Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)

[Ste07] Stell, J.G.: Relations in Mathematical Morphology with Applications to Graphs and Rough Sets. In: Winter, S., Duckham, M., Kulik, L., Kuipers, B., et al. (eds.) COSIT 2007. LNCS, vol. 4736, pp. 438–454. Springer, Heidelberg (2007)

[Tay99] Taylor, P.: Practical Foundations of Mathematics. Cambridge University Press, Cambridge (1999)

[Yao98] Yao, Y.Y.: Relational interpretations of neighborhood operators and rough set approximation operators. Information Sciences 111, 239–259 (1998)

[Yao01] Yao, Y.Y.: Information granulation and rough set approximation. International Journal of Intelligent Systems 16, 87–104 (2001)

[Zad97] 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)

[Zhu07] Zhu, W.: Generalised rough sets based on relations. Information Sciences 177, 4997–5011 (2007)

Perceptual Tolerance Intersection

Piotr Wasilewski[1], James F. Peters[1], and Sheela Ramanna[2]

[1] Computational Intelligence Laboratory,
Department of Electrical & Computer Engineering, Univ. of Manitoba,
75A Chancellor's Circle, Winnipeg, Manitoba R3T 5V6, Canada
{piotr,jfpeters}@ee.umanitoba.ca
[2] Department of Applied Computer Science
University of Winnipeg
Winnipeg, Manitoba R3B 2E9, Canada
s.ramanna@uwinnipeg.ca

Abstract. This paper introduces a perceptual tolerance intersection of sets as an example of near set operations. Such operations are motivated by the need to consider similarities between digital images viewed as disjoint sets of points. The proposed approach is in keeping with work by E.C. Zeeman on tolerance spaces and visual perception and J.H. Poincaré on sets of similar sensations used to define representative (aka *tolerance*) spaces such as visual, tactile and motile spaces. Perceptual tolerance intersection of sets is a direct consequence of recent work on near sets and a solution to the problem of how one goes about discovering affinities between digital images. The main contribution of this article is a description-based approach to assessing the resemblances between digital images.

Keywords: description, near sets, perceptual granules, set operations, similarity, tolerance.

1 Introduction

This paper introduces a perceptual tolerance intersection of sets as an example of near set operations useful in the study of similarities between digital images. The proposed set operations considered in the context of tolerance spaces is directly related to work on sets of similar objects, starting with J.H. Poincaré [10] and E.C. Zeeman [14], followed by more recent studies of similarity and tolerance relations [9,12,7,6,13,8]. In general, sets are considered near each other in the case where the sets contain objects with descriptions that are similar.

The paper is divided into two parts. In the first part, general facts about tolerance relations are presented together with Zeeman's indistinguishability of sets relation (section 2) and a tolerance intersection of sets operation is introduced and investigated (section 3). In the second part, perceptual tolerance relations and perceptual tolerance intersection of sets operation are discussed (section 4). The paper ends by proposing a postulate on similarity measures between images which are based on perceptual tolerance relations.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 277–286, 2010.

2 Tolerance Relations

A relation $\tau \subseteq O \times O$ is a *tolerance on a set* O (shortly: *tolerance*, if O is understood) iff τ is reflexive and symmetric[1]. Then a pair $\langle O, \tau \rangle$ is a *tolerance space*. We denote the family of all tolerances on a set O by $Tol(O)$. Transitive tolerances are equivalence relations, i.e. $Eq(O) \subseteq Tol(O)$, where $Eq(O)$ denotes the family of all equivalences on O. An image of a set $X \subseteq O$ by a relation τ on O we denote by $\tau(X)$ (*i.e.* $\tau(X) := \{y \in O : \text{ there is } x \in X, (x, y) \in \tau\}$) with a simplifying convention where $\tau(\{x\}) = \tau(x)$. A tolerance image operator $\tau(\ )$ has some useful properties presented by the following lemma:

Lemma 1. *Let $\langle O, \tau \rangle$ be a tolerance space. The following conditions hold for arbitrary $X, Y \subseteq O$:*

(1) $X \subseteq \tau(X)$, (Extensivity)
(2) $X \subseteq Y \Rightarrow \tau(X) \subseteq \tau(Y)$, (Monotonicity)
(3) $X \subseteq Y \Rightarrow X \subseteq \tau(Y)$,
(4) $\tau(X) = \bigcup_{x \in X} \tau(x)$.

Every tolerance generates some specific coverings of a space. Two of them are mainly used. A set $A \subseteq O$ is a *τ-preclass* (or briefly *preclass* when τ is understood) if and only if for any $x, y \in A$, $(x, y) \in \tau$. The family of all preclasses of a tolerance space is naturally ordered by set inclusion and preclasses that are maximal with respect to a set inclusion are called *τ-classes* or just *classes*, when τ is understood. The family of all classes of the space $\langle O, \tau \rangle$ is particularly interesting and is denoted by $H_\tau(O)$. The family $H_\tau(O)$ is a covering of O. However, the elements of $H_\tau(O)$ do not have to be mutually disjoint. The elements of $H_\tau(O)$ are mutually disjoint when a given tolerance τ is transitive, *i.e.*, τ is an equivalence relation. Hence, the notion of a family of tolerance classes is a natural generalization of the partition of a space.

A tolerance space $\langle O, \tau \rangle$ determines, as any relation, another family of sets, namely the family of images of elements of the space via a given tolerance relation: $\{\tau(x) : x \in O\}$. Clearly, since τ is reflexive, the family $\{\tau(x) : x \in O\}$ is a covering of a space O but it does not have to be a partition of O (analogously with the family $H_\tau(O)$, it is a partition of O when τ is transitive). However, families consisting of images of elements via tolerance relations are not natural generalizations of partitions of a space, since, for every intransitive tolerance τ on O, there is $x \in O$ and there are $a, b \in \tau(x)$ such that $(a, b) \notin \tau$. Thus one can see that images of elements with respect to a given tolerance relation are not in general tolerance classes. This holds only in the case of transitive tolerance relations, *i.e.* in the case of equivalence relations.

Often an image of an element x, $\tau(x)$, is called a *neighbourhood of x* while x itself is called the *centre* of $\tau(x)$. Toward the end of this section, it will become

[1] In universal algebra or lattice theory reflexive and symmetric relations compatible with operations from a given algebra are called *tolerances*, i.e. they are generalizations of congruence relations (see e.g. [2]). We refer to such relations as *algebraic tolerances* or *algebraic tolerance relations*.

apparent that there is some reason underlying this convention. One should also note that a neighbourhood $\tau(x)$ of the element x is uniquely determined by its centre, while an element x can belong to more than one tolerance class. There is also another difference between tolerance neighbourhoods and classes that is interesting from a mathematical point of view. Families of tolerance images of elements exist for any set, finite or infinite, but this does not hold in the case of tolerance classes. If a set is finite, then, by its finiteness, every tolerance preclass is contained in some class and, in the case of an infinite set, that condition is equivalent to the Axiom of Choice in set theory [13] (in the case of algebraic tolerances on semilattices it was shown in [2] that the Axiom of Choice is equivalent to the existence of a single tolerance class). So, in general, tolerance neighbourhoods and tolerance classes are different entities.

E.C. Zeeman pointed out [14] that any tolerance relation determines in a natural way another tolerance on the subsets of the space.

Definition 1. [14] *Let $\langle O, \tau\rangle$ be a tolerance space. A relation $\sim_\tau$ on $\mathcal{P}(O)$ is defined as follows:*

$$X \sim_\tau Y \Leftrightarrow X \subseteq \tau(Y) \text{ and } Y \subseteq \tau(X)$$

X is said to be indistinguishable from Y. We refer to the relation $\sim_\tau$ as Zeeman's tolerance *or* Zeeman's indistinguishability of sets.

If a tolerance τ is treated as a formal model of similarity, then the basic intuitive interpretation given to a relation $\sim_\tau$ is that sets standing in this relation are indistinguishable with respect to a tolerance τ, as containing only mutually similar elements.

Corollary 1. *Let $\langle O, \tau\rangle$ be a tolerance space. If τ is transitive, so $\tau \in Eq(O)$, then:*

$$X \sim_\tau Y \Leftrightarrow \tau(X) = \tau(Y),$$

i.e. *Zeeman's indistinguishability of sets is Z. Pawlak's upper rough equality of sets from rough set theory [3,4].*

Proof. Equation $\tau(X) = \tau(Y)$ together with extensivity of the operator $\tau(\ )$ directly implies $X \sim_\tau Y$ so it is enough to prove implication $\Rightarrow$. Let $\tau \in Eq(O)$, so $\langle O, \tau\rangle$ is an approximation space while $\tau(\ )$ is an upper approximation operator [3,4]. Let $X \sim_\tau Y$, so $X \subseteq \tau(Y)$ and by monotonicity $\tau(X) \subseteq \tau(\tau(Y))$ thus $\tau(X) \subseteq \tau(Y)$ by $\tau(Y) = \tau(\tau(Y))$, one of the properties of an upper approximation operator [3,4]. Analogically for $\tau(Y) \subseteq \tau(X)$, therefore, $\tau(X) = \tau(Y)$.

After the manner of E.C. Zeeman [14], every pseudometric space in a quite natural way determines tolerance relations with respect to some positive real threshold as shown by Example 1.

Example 1. Let $\langle O, p\rangle$ be pseudometric space and let $\epsilon \in (0, +\infty)$. A relation $\tau_{p,\epsilon}$ is defined for $x, y \in O$ in the following way:

$$(x, y) \in \tau_{p,\epsilon} \iff p(x, y) < \epsilon,$$

is a tolerance relation on O. Such relations we call *distance tolerance relations.*

One can show that

Proposition 1. *Let $\tau_{p,\epsilon}$ be a distance tolerance relation determined by a pseudometric space $\langle O, p\rangle$. Then, for any $x \in U$,*

$$\tau_{p,\epsilon}(x) = B_p(x, \epsilon),$$

i.e., a $\tau_{p,\epsilon}$ neighbourhood of x is just an open ball in the pseudometric space $\langle O, p\rangle$ with the centre x and radius ϵ, $B_p(x, \epsilon) := \{y \in X : p(x, y) \leq \varepsilon\}$.

Proposition 1 justifies referring to an image of the element x by any tolerance τ (not necessarily a distance tolerance) as a neighbourhood with a centre x, since in topology a named *neighbourhood of* x denotes an open ball or, as in [1], an open set containing element x.

3 Tolerance Intersection of Sets

Assuming that tolerance is a formal model of similarity, then, for any two subsets (possibly disjoint) of a tolerance space, one can ask whether the subsets contain some mutually similar elements. This motivates introducing an operation on subsets of tolerance spaces.

Definition 2. *Let $\langle O, \tau\rangle$ be a tolerance space. A* tolerance intersection *of sets is denoted by $\Cap_\tau$ and defined for $X, Y \subseteq O$ as follows:*

$$X \Cap_\tau Y := (X \cap \tau(Y)) \cup (Y \cap \tau(X)).$$

Let us note that disjoint sets can have a non-empty tolerance intersection as it is shown by the following example:

Example 2. Let $\langle O, \tau\rangle$ denote a tolerance space, where $O = \{a_1, a_2, b_1, b_2, c, d\}$ and $\tau := \Delta_O \cup \{(a_1, b_2), (b_2, a_1), (a_2, b_1), (b_1, a_2), (a_1, c), (c, a_1), (b_1, d), (d, b_1)\}$. Let also $A := \{a_1, a_2\}$, $B := \{b_1, b_2\}$, where Δ_O denotes the diagonal of a set O, i.e. $\Delta_O := \{(x, x) : x \in O\}$. Then, by straightforward calculations, the following equations hold:

$$\tau(A) = \{a_1, a_2, b_1, b_2, c\}, \qquad \tau(B) = \{a_1, a_2, b_1, b_2, d\}.$$

Thus $A \subseteq \tau(B)$ and $B \subseteq \tau(A)$. Therefore $A \sim_\tau B$ and $A \Cap_\tau B = \{a_1, a_2, b_1, b_2\}$ but $A \cap B = \emptyset$.

Example 2 shows also that disjoint sets can be indistinguishable in Zeeman's sense. Of course, indistinguishability of disjoint sets is not equivalent to having a non-empty tolerance intersection and one can easily find a counterexample to such claim on the basis of Example 2. Let us compare tolerance intersection of sets to ordinary intersection and union of sets.

Proposition 2. *Let $\langle O, \tau\rangle$ be a tolerance space and let $X, Y \subseteq O$. Then the following conditions hold:*

1. $X \cap Y \subseteq X \Cap_\tau Y$,
2. $X \Cap_\tau Y \subseteq X \cup Y$.

Proof (1) From definition and extensivity, $Y \subseteq \tau(Y)$, we get $X \cap Y \subseteq X \cap \tau(Y)$ so $X \cap Y \subseteq (X \cap \tau(Y)) \cup (Y \cap \tau(X)) = X \Cap_\tau Y$. Thus $X \cap Y \subseteq X \Cap_\tau Y$.
(2) Since $X \cap \tau(Y) \subseteq X \cup Y$ and $Y \cap \tau(X) \subseteq X \cup Y$, thus $(X \cap \tau(Y)) \cup (Y \cap \tau(X)) \subseteq X \cup Y$ and by definition $X \Cap_\tau Y \subseteq X \cup Y$.

Lemma 2. *Let $\langle O, \tau \rangle$ be a tolerance space and let $X, Y \subseteq O$. Then*

$$X \Cap_\tau Y \subseteq \tau(X) \cap \tau(Y).$$

Considering whether a tolerance intersection of sets coincides with the ordinary intersection or union of sets, $X \Cap_\tau Y = X \cap Y$, $X \Cap_\tau Y = X \cup Y$, respectively, leads to a number of interesting observations given in Prop. 3.

Proposition 3. *Let $\langle O, \tau \rangle$ be a tolerance space and let $X, Y \subseteq O$. If $X \Cap_\tau Y = X \cap Y$, then the following conditions hold:*

1. $X \cap \tau(Y) = X \cap Y$,
2. $Y \cap \tau(X) = X \cap Y$,
3. $X \cap \tau(Y) = Y \cap \tau(X)$.

Proof. Let $X \Cap_\tau Y = X \cap Y$, so $X \cap \tau(Y) \subseteq X \cap Y$, always $X \cap Y \subseteq X \cap \tau(Y)$, thus $X \cap \tau(Y) = X \cap Y$. Analogously $Y \cap \tau(X) = X \cap Y$. 1 and 2 implies 3.

Proposition 4. *Let $\langle O, \tau \rangle$ be a tolerance space and let $X, Y \subseteq O$. Then the following condition hold:*

$$\textit{If } X = \tau(X) \textit{ and } Y = \tau(Y), \textit{ then } X \Cap_\tau Y = X \cap Y,$$

i.e., on the family of sets closed w.r.t. the operator $\tau(\ )$ (Pawlak's definable sets in rough set theory [3,4], when τ is transitive) a tolerance intersection of sets coincides with ordinary intersection of sets.

Proposition 5. *Let $\langle O, \tau \rangle$ be a tolerance space and let $X, Y \subseteq O$. Then the following conditions are equivalent:*

1. $X \Cap_\tau Y = X \cup Y$,
2. $X \sim_\tau Y$.

i.e. *only on the families of mutually indistinguishable sets in Zeeman's sense (maximal preclasses of the tolerance $\sim_\tau$) a tolerance intersection of sets coincides with the union of sets.*

Proof. ($\Rightarrow$). If $X \Cap_\tau Y = X \cup Y$, then by lemma 2 we get $X \Cap_\tau Y \subseteq \tau(X) \cap \tau(Y)$. Thus we get that $X \cup Y \subseteq \tau(X) \cap \tau(Y)$. Thus $X \subseteq \tau(Y)$ and $Y \subseteq \tau(X)$, so $X \sim_\tau Y$.
($\Leftarrow$) Let $X \sim_\tau Y$, so $X \subseteq \tau(Y)$ and $Y \subseteq \tau(X)$. $X \subseteq \tau(Y)$ implies $X \subseteq X \cap \tau(Y)$ and so $X \subseteq X \Cap_\tau Y$. Analogically for $Y \subseteq X \Cap_\tau Y$. Thus $X \cup Y \subseteq X \Cap_\tau Y$. By proposition 2 we get $X \Cap_\tau Y \subseteq X \cup Y$. Therefore $X \Cap_\tau Y = X \cup Y$.

Prop. 6 presents some basic properties of the tolerance intersection operation.

Proposition 6. *Let $\langle O, \tau \rangle$ be a tolerance space and let $X, Y \subseteq O$. Then the following conditions hold:*

1. $X \mathbin{\Cap}_\tau Y = Y \mathbin{\Cap}_\tau X$,
2. $(X \cap Y) \mathbin{\Cap}_\tau (X \cap Z) \subseteq X \cap (Y \mathbin{\Cap}_\tau Z)$,
3. $X \cup (Y \mathbin{\Cap}_\tau Z) \subseteq (X \cup Y) \mathbin{\Cap}_\tau (X \cup Z)$.

Proof. (1) From the definition and commutativity of the union of sets.
(2) By monotonicity $\tau(X \cap Z) \subseteq \tau(Z)$. So $(X \cap Y) \cap \tau(X \cap Z) \subseteq (X \cap Y) \cap \tau(Z) = X \cap (Y \cap \tau(Z))$. Analogically for $(X \cap Z) \cap \tau(X \cap Y) \subseteq X \cap (Z \cap \tau(Y))$. Thus $(X \cap Y) \cap \tau(X \cap Z), (X \cap Z) \cap \tau(X \cap Y) \subseteq (X \cap (Y \cap \tau(Z))) \cup (X \cap (Z \cap \tau(Y))) = X \cap ((Y \cap \tau(Z)) \cup (Z \cap \tau(Y)))$ and so $((X \cap Y) \cap \tau(X \cap Z)) \cup ((X \cap Z) \cap \tau(X \cap Y)) \subseteq X \cap ((Y \cap \tau(Z)) \cup (Z \cap \tau(Y)))$. Therefore by the definition of a perceptual intersection one can show that $(X \cap Y) \mathbin{\Cap}_\tau (X \cap Z) \subseteq X \cap (Y \mathbin{\Cap}_\tau Z)$.
(3) By monotonicity $\tau(Z) \subseteq \tau(X \cup Z)$ and so $Y \cap \tau(Z) \subseteq \tau(X \cup Z)$. By extensivity $X \subseteq \tau(X \cup Z)$, thus $X \cup (Y \cap \tau(Z)) \subseteq \tau(X \cup Z)$. It also holds that $X \cup (Y \cap \tau(Z)) \subseteq X \cup Y$. Therefore $X \cup (Y \cap \tau(Z)) \subseteq (X \cup Y) \cap \tau(X \cup Z)$. Analogically one can show that $X \cup (Z \cap \tau(Y)) \subseteq (X \cup Z) \cap \tau(X \cup Y)$. Thus $[X \cup (Y \cap \tau(Z))] \cup [X \cup (Z \cap \tau(Y))] \subseteq [(X \cup Y) \cap \tau(X \cup Z)] \cup [(X \cup Z) \cap \tau(X \cup Y)]$ so $X \cup [(Y \cap \tau(Z)) \cup (Z \cap \tau(Y))] \subseteq [(X \cup Y) \cap \tau(X \cup Z)] \cup [(X \cup Z) \cap \tau(X \cup Y)]$. and $X \cup [(Y \cap \tau(Z)) \cup (Z \cap \tau(Y))] \subseteq (X \cup Y) \cap [\tau(X \cup Z) \cup \tau(X \cup Y)]$ by distributivity of set theoretical operations. Therefore by definition of a perceptual intersection it follows that $X \cup (Y \mathbin{\Cap}_\tau Z) \subseteq (X \cup Y) \mathbin{\Cap}_\tau (X \cup Z)$.

Now, keeping in mind the similarity interpretation of tolerance relations, we can introduce a tolerance intersection measure for finite subsets of a tolerance space. The family of all finite subsetets of a set O is denoted by $\mathcal{P}_{fin}(O)$.

Definition 3. *Let $\langle O, \tau \rangle$ be a tolerance space and let $X, Y \in \mathcal{P}_{fin}(O)$ and at least one of them is non-empty. A* tolerance intersection measure *is denoted by* pi_τ *and defined as follows:*

$$\mathrm{pi}_\tau(\mathrm{X}, \mathrm{Y}) := \frac{|\mathrm{X} \mathbin{\Cap}_\tau \mathrm{Y}|}{|\mathrm{X} \cup \mathrm{Y}|}.$$

Theorem 1. *Let $\langle O, \tau \rangle$ be a tolerance space and let $X, Y \in \mathcal{P}_{fin}(O)$ and $X \neq \emptyset$ or $Y \neq \emptyset$. Then the following conditions are equivalent:*

1. $X \sim_\tau Y$,
2. $X \mathbin{\Cap}_\tau Y = X \cup Y$,
3. $\mathrm{pi}_\tau(\mathrm{X}, \mathrm{Y}) = 1$.

Proof. Because of Proposition 5 and the fact that implication $2 \Rightarrow 3$ follows directly for definition it is enough to show $3 \Rightarrow 2$. Let $\mathrm{pi}_\tau(\mathrm{X}, \mathrm{Y}) = 1$, thus $|X \mathbin{\Cap}_\tau Y| = |X \cup Y|$. Since $X \mathbin{\Cap}_\tau Y \subseteq X \cup Y$, then by finiteness of sets X and Y it follows that $X \mathbin{\Cap}_\tau Y = X \cup Y$.

In the light of Theorem 1, we see that a tolerance intersection measure is a measure of tolerance distinguishability of sets in Zeeman's sense.

4 Near Sets and Perceptual Tolerance Intersection

Perceptual systems in near set theory [5,6,8] reflect Poincaré's idea of perception. A *perceptual system* is a pair $\langle O, \mathbb{F} \rangle$, where O is a non-empty set of *perceptual objects* and $\mathbb{F}$ is a non-empty set of real valued functions defined on O, *i.e.*, $\mathbb{F} := \{\phi \mid \phi : O \to \mathbb{R}\}$, where ϕ is called a *probe function*. Perceptual objects spring directly from the perception of physical objects derived from sets of sensations in Poincaré's view of the physical continuum [11]. A probe function $\phi \in \mathbb{F}$ is viewed as a representation of a feature in the description of sets of sensations. So, for example, a digital image Im can be seen as a set of perceptual objects, *i.e.*, $Im \subseteq O$, where every perceptual object is described with vectors of probe function values.

A family of probe functions $\mathbb{F}$ can be infinite[2]. In applications such as image analysis, from a possibly infinite family of probe functions, we always select a finite number of probe functions, $\mathcal{B} \subseteq \mathbb{F}$ and $|\mathcal{B}| < \aleph_0$, in order to describe perceptual objects (usually pixels or pixel windows in digital images). Thus, every perceptual object $x \in O$ can be described by a vector $\boldsymbol{\phi}_{\mathcal{B}}(x)$ of real values of probe functions in a space $\mathbb{R}^n$ *i.e.*

$$\boldsymbol{\phi}_{\mathcal{B}}(x) = (\phi_1(x), \phi_2(x), \ \ldots \ , \phi_n(x)),$$

where $\mathcal{B} := \{\phi_1, \ \ldots \ , \phi_n\}$ for $\mathcal{B} \subseteq \mathbb{F}$.

4.1 Perceptual Tolerance Relations

With object descriptions, one can compare objects with respect to various metric or pseudometric distances defined on $\mathbb{R}^n$. More generally, one can introduce on the set O different topologies based on topologies determined on the space $\mathbb{R}^n$ (note that such topologies are not necessarily induced from $\mathbb{R}^n$). For example, consider a natural topology on $\mathbb{R}^n$ determined by Euclidean distance, denoted here by d. Using d one can define the distance measure on O in the following way:

$$p_{\mathcal{B}}(x, y) := d(\boldsymbol{\phi}_{\mathcal{B}}(x), \boldsymbol{\phi}_{\mathcal{B}}(y)) = \sqrt{\sum_{i=1}^{n} (\phi_i(x) - \phi_i(y))^2},$$

where $\mathcal{B} \subseteq \mathbb{F}$ and $\mathcal{B} := \{\phi_1, \ \ldots \ , \phi_n\}$. Notice that d is a metric on $\mathbb{R}^n$ but $p_{\mathcal{B}}$ is not necessarily a metric on O, since it is possible that there are $x, y \in O$ such that $p_{\mathcal{B}}(x, y) = 0$ but $x \neq y$, *i.e.*, two different perceptual objects can have exactly the same description over a family of probe functions. Moreover, similarly to the case of the transitivity of distance tolerances, the condition $p_{\mathcal{B}}(x, y) = 0 \Leftrightarrow x = y$ is neither implied nor excluded by the definition of $p_{\mathcal{B}}$.

[2] From a digital image analysis perspective, the potential for a countable number of probe functions has a sound interpretation, *i.e.*, the number of image probe functions is finite but unbounded, since new probe functions can be created over an indefinitely long timespan and added to the set of existing probes.

When the set O only consists of objects with mutually different descriptions, the function $p_{\mathcal{B}}$ is a metric on O.

From Example 1, for a perceptual system and some pseudometric one can define for a real threshold $\epsilon \in (0, +\infty)$ a distance tolerance relation

Definition 4. *Let $\langle O, \mathbb{F}\rangle$ be a perceptual system, $\langle O, p_{\mathcal{B}}\rangle$ be a pseudometric space where $\mathcal{B} := \{\phi_i(x)\}_{i=1}^n \subseteq \mathbb{F}$. A relation $\cong_{\mathcal{B},\epsilon}$ is defined for any $x, y \in O$ as follows:*

$$(x, y) \in \cong_{\mathcal{B},\epsilon} :\Leftrightarrow p_{\mathcal{B}}(x, y) < \epsilon.$$

A relation $\cong_{\mathcal{B},\epsilon}$ is a distance tolerance relation and we call it perceptual tolerance relation.

$\cong_{\mathcal{B},\epsilon}$ reflects Poincaré's idea, *i.e.*, sensations are similar if their descriptions are close enough in a space $\mathbb{R}^n$. Note that a relation $\cong_{\mathcal{B},\epsilon}$ depends not only on a choice of a threshold but also on the choice of a family of probe function. For the same threshold and for two different families of probe functions one can get two distinct perceptual tolerance relations. As a direct consequence of Prop. 1, one can infer:

Corollary 2. *Let $\langle O, \mathbb{F}\rangle$ be a perceptual system, $\langle O, p_{\mathcal{B}}\rangle$ be a pseudometric space where $\mathcal{B} := \{\phi_i(x)\}_{i=1}^n \subseteq \mathbb{F}$. Then for any $x \in O$ holds that*

$$\cong_{\mathcal{B},\epsilon} (x) = B_{p_{\mathcal{B}}}(x, \epsilon),$$

i.e., *a $\cong_{\mathcal{B},\epsilon}$ neighbourhood of x is just an open ball in the pseudometric space $\langle O, p_{\mathcal{B}}\rangle$ with centre x and radius ϵ, where a centre x can be identified with an equivalence class $x_{/\theta_{p_{\mathcal{B}}}}$, where $(x, y) \in \theta_{p_{\mathcal{B}}} :\Leftrightarrow p_{\mathcal{B}}(x, y)$ for $x, y \in O$.*

This corresponds to Poincaré's idea that sensations are identified with particular sets of sensations that are very similar. It can also be observed that when sensations $x, y \in O$ are close enough, they become indistinguishable. In a near set approach, the indistinguishability of sensations results from sensations that have the same descriptions over a selected family of probe functions $\mathcal{B} \subseteq \mathbb{F}$, *i.e.*, the pseudometric distance $p_{\mathcal{B}}$ between x and y is equal to 0. From a near set perspective, in the light of corollary 2 it can be also observed that similarity between perceptual objects depends on three independent factors: a choice of a finite family of probe functions as a basis of object descriptions, a choice of a pseudometric distance function for a set of perceptual objects and a choice of a positive real threshold. Since probe functions represent results of perception (interaction of sensors with the environment), then the selected family of probe functions corresponds to a frame of sensors. The selected positive real threshold can represent a sensitivity of perceptual machinery interacting with the environment. Corollary 2 reflects also the fact that a process of perception (interaction of sensors with the environment) results in the first granularization, *perceptual granularization* of the set of sensations. Mathematically it is represented by a pseudometric space $\langle O, p_{\mathcal{B}}\rangle$ derived from a perceptual system $\langle O, \mathbb{F}\rangle$ on the basis of a finite family $\mathcal{B} \subseteq \mathbb{F}$, where the set of perceptual objects O is divided by the equivalence relation $\theta_{p_{\mathcal{B}}}$ into classes consisting of objects indistinguishable w.r.t. sensitivity of sensors interacting with the environment.

4.2 Perceptual Intersection of Sets and Perceptual Similarity Measures

On the basis of perceptual tolerance relations we can introduce perceptual intersection of sets being a particular form of tolerance intersections of sets.

Definition 5. *Let $\langle O, \mathbb{F}\rangle$ be a perceptual system and let $\langle O, \cong_{\mathcal{B},\epsilon}\rangle$ be a perceptual tolerance space where $\mathcal{B} \subseteq \mathbb{F}$ and $\epsilon \in (0, +\infty)$. A* perceptual intersection of sets based on $\langle O, \cong_{\mathcal{B},\epsilon}\rangle$ *(or shortly* perceptual intersection of sets *when a perceptual tolerance space is understood) is denoted by $\Cap_{\mathcal{B},\epsilon}$ and defined for $X, Y \subseteq O$ as follows:*

$$X \Cap_{\mathcal{B},\epsilon} Y := (X \cap \cong_{\mathcal{B},\epsilon} (Y)) \cup (Y \cap \cong_{\mathcal{B},\epsilon} (X)).$$

That $\Cap_{\mathcal{B},\epsilon}$ perceptually originated from of tolerance intersection can be seen in its similarity nature. Sets $Im_1, Im_2 \subseteq O$, where $\langle O, \mathbb{F}\rangle$ is a perceptual system, can be digital images. The perceptual intersection of Im_1 and Im_2 consists of those perceptual objects belonging to Im_1 or Im_2 which have similar 'cousins' in the other image. On the basis of perceptual intersection of sets, we can now introduce a perceptual intersection measure of the similarity of sets.

Definition 6. *Let $\langle O, \mathbb{F}\rangle$ be a perceptual system and let $\langle O, \cong_{\mathcal{B},\epsilon}\rangle$ be a perceptual tolerance space where $\mathcal{B} \subseteq \mathbb{F}$ and $\epsilon \in (0, +\infty)$. A* perceptual intersection measure *is denoted by $\mathrm{p}_{\Cap_{\mathcal{B},\epsilon}}$ and defined for any $X, Y \in \mathcal{P}_{fin}(O)$, where $X \neq \emptyset$ or $Y \neq \emptyset$.*

$$\mathrm{p}_{\Cap_{\mathcal{B},\epsilon}}(\mathrm{X}, \mathrm{Y}) := \frac{|\mathrm{X} \Cap_{\mathcal{B},\epsilon} \mathrm{Y}|}{|\mathrm{X} \cup \mathrm{Y}|}.$$

Since a perceptual intersection measure is a particular form of a tolerance intersection measure, so Theorem 1 also applies to it. Additionally, one can note that when a tolerance τ is a perceptual tolerance and sets X and Y are images in some perceptual system, then Zeeman's tolerance $\sim_\tau$ becomes a perceptual indistinguishability of images. Thus a perceptual intersection measure is a measure of perceptual indistinguishability being a kind of similarity measures between images. To conclude the paper, taking into account a direct connection of a perceptual intersection measure to a perceptual form of the Zeeman's tolerance we formulate a postulate on similarity measures between images which are based on perceptual tolerance relations:

Postulate
Every similarity measure μ_ρ derived from a perceptual system $\langle O, \mathbb{F}\rangle$ on the basis of some perceptual tolerance relation ρ should fulfill the following condition for $X, Y \subseteq O$:

$$\mu_\rho(X, Y) = 1 \;\; if \; and \; only \; if \;\; X \sim_\rho Y.$$

5 Conclusion

In this paper, tolerance intersection of sets and a tolerance intersection measure together with their perceptual forms derived from perceptual tolerances and

perceptual systems have been introduced . The properties of the proposed set operations and measures and their connections to Zeeman's indistinguishability of sets together with their perceptual applications and implications to similarity measurement in solving the digital image correspondence problem are also given.

Acknowledgements. This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grants 185986, 194376, Canadian Arthritis Network grant SRI-BIO-05, Manitoba Centre of Excellence Fund (MCEF) grant T277 and grants N N516 077837, N N516 368334 from the Ministry of Science and Higher Education of the Republic of Poland.

References

1. Engelking, R.: General Topology, Revised & completed edition. Heldermann Verlag, Berlin (1989)
2. Gratzer, G., Wenzel, G.: Tolerances, covering systems, and the axiom of choice. Archivum Mathematicum 25(1-2), 27–34 (1989)
3. Pawlak, Z.: Rough sets. International J. Comp. Inform. Science 11, 341–356 (1981)
4. Pawlak, Z.: Rough sets. Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, Dordrecht (1991)
5. Peters, J.F.: Near sets. Special theory about nearness of objects. Fundamenta Informaticae 75(1-4), 407–433 (2007)
6. Peters, J., Ramanna, S.: Affinities between perceptual granules: Foundations and perspectives. In: Bargiela, A., Pedrycz, W. (eds.) Human-Centric Information Processing Through Granular Modelling. SCI, vol. 182, pp. 49–66. Springer, Berlin (2009)
7. Peters, J., Skowron, A., Stepaniuk, J.: Nearness of objects: Extension of approximation space model. Fundamenta Informaticae 79(3-4), 497–512 (2007)
8. Peters, J.F., Wasilewski, P.: Foundations of near sets. Information Science 179(18), 3091–3109 (2009)
9. Pogonowski, J.: Tolerance Spaces with Applications to Linguistics. University of Adam Mickiewicz Press, Poznań (1981)
10. Poincaré, J.: Sur certaines surfaces algébriques; troisième complément 'a l'analysis situs. Bulletin de la Société de France 30, 49–70 (1902)
11. Poincaré, J.: Dernières pensées, trans. by J.W. Bolduc as Mathematics and Science: Last Essays. Flammarion & Kessinger Pub., Paris (1913 & 2009), http://docenti.lett.unisi.it/files/4/1/1/36/Dernierespenseespoinc.pdf
12. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27(2-3), 245–253 (1996)
13. Wasilewski, P.: On selected similarity relations and their applications into cognitive science. PhD thesis, Department of Logic, Jagiellonian University, Cracow (2004) (in Polish)
14. Zeeman, E.: The topology of the brain and visual perception. In: Fort Jr., M.K. (ed.) University of Georgia Institute Conference Proceedings, Topology of 3-Manifolds and Related Topics, vol. 1962, pp. 240–256. Prentice-Hall, Inc., Englewood Cliffs (1962)

Categories of Direlations and Rough Set Approximation Operators

Murat Diker

Hacettepe University, Department of Mathematics,
06800 Beytepe Ankara, Turkey
mdiker@hacettepe.edu.tr

Abstract. In this paper, we define a category **R-APR** whose objects are sets and morphisms are the pairs of rough set approximation operators. We show that **R-APR** is isomorphic to a full subcategory of the category **cdrTex** whose objects are complemented textures and morphisms are complemented direlations. Therefore, **cdrTex** may be regarded as an abstract model for the study of rough set theory. On the other hand, dagger symmetric monoidal categories play a central role in the abstract quantum mechanics. Here, we show that **R-APR** and **cdrTex** are also dagger symmetric monoidal categories.

Keywords: Approximation operator, dagger category, direlation, rough set, symmetric monoidal category, textural rough set, texture space.

1 Introduction

Category theoretical approaches to rough set theory are initiated by Banerjee and Chakraborty in [2]. They defined a category **ROUGH** using approximations with respect to given partitions. A category $\mathcal{C}_{\mathcal{G}}$ of granulations is proposed in [3] and in fact, **ROUGH** is a subcategory of $\mathcal{C}_{\mathcal{G}}$. Partially ordered monads provide an alternative perspective for rough set theory [9,10,11]. Further, I-rough sets are presented by Iwinski [13] and the category **RSC** of I-rough sets is introduced and topos properties are studied in [12]. Here, we present a categorical discussion on rough set theory using approximation operators as morphisms. In [8], a new direction in generalizing rough sets is presented and a formulation of rough set operators based on texture is given. Some important categories of textures have been studied in [6]. For instance, **dfTex** is a ground category whose objects are textures and morphisms are difunctions which plays an important role in the theory of texture spaces. Difunctions between textures are also direlations and it must be noted that the direlations can be regarded as a generalization of ordinary relations and rough set approximation operators between sets. Here, we report that the complemented textures and complemented direlations form a category which is denoted by **cdrTex**. In view of this fact, sets and rough set approximation operators also form a category denoted by **R-APR**. This category is isomorphic to the category **Rel** of sets and relations. Actually, **cdrTex** may be one of the suitable abstract models for rough set theory since **R-APR**

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 287–296, 2010.

is a full subcategory of **cdrTex**. On the other hand, dagger symmetric monoidal categories play a central role in the abstract quantum mechanics [1,15]. Here, we prove that **R-APR** and **cdrTex** are also dagger symmetric monoidal categories.

2 Basic Concepts

A texturing on a universe U is a point separating, complete, completely distributive lattice $\mathcal{U}$ of subsets of U with respect to inclusion which contains $U, \emptyset$ and, for which arbitrary meet coincides with intersection and finite joins coincide with union. Then the pair $(U,\mathcal{U})$ is called a *texture space*, or simply *a texture* [5]. A mapping $c_U : \mathcal{U} \to \mathcal{U}$ is called *a complementation* on $(U,\mathcal{U})$ if it satisfies the conditions $c_U(c_U(A)) = A$ for all $A \in \mathcal{U}$ and $A \subseteq B$ in $\mathcal{U}$ implies $c_U(B) \subseteq c_U(A)$. Then the triple $(U,\mathcal{U},c_U)$ is said to be *a complemented texture space*. For $u \in U$, the p-sets and q-sets are defined by $P_u = \bigcap\{A \in \mathcal{U} \mid u \in A\}$ and $Q_u = \bigvee\{A \in \mathcal{U} \mid u \notin A\}$.

Example 1. [6] (i) The pair $(U, \mathcal{P}(U))$ is a texture space where $\mathcal{P}(U)$ is the power set of U. It is called *a discrete texture*. For $u \in U$ we have $P_u = \{u\}$ and $Q_u = U \setminus \{u\}$ and $c_U : \mathcal{P}(U) \to \mathcal{P}(U)$ is the ordinary complementation on $(U, \mathcal{P}(U))$ defined by $c_U(A) = U \setminus A$ for all $A \in \mathcal{P}(U)$.

(ii) Let $U = \{a, b, c\}$. Then $\mathcal{U} = \{\emptyset, \{a\}, \{a, b\}, U\}$ is a texture on U. Clearly, $P_a = \{a\},\ P_b = \{a, b\},\ P_c = U$ and $Q_a = \emptyset,\ Q_b = \{a\},\ Q_c = \{a, b\}$. Further, the mapping $c_U : \mathcal{U} \to \mathcal{U}$ defined by $c_U(\emptyset) = U,\ c_U(U) = \emptyset,\ c_U(\{a\}) = \{a, b\},\ c_U(\{a, b\}) = \{a\}$ is a complementation on $(U,\mathcal{U})$.

(iii) The family $\mathcal{U} = \{(0, r] \mid r \in [0, 1]\}$ is a texture on $U = (0, 1]$ which is called the Hutton texture. Here, for $r \in [0, 1]$, we have $P_r = Q_r = (0, r]$ and the complementation $c_U : \mathcal{U} \to \mathcal{U}$ is defined by $\forall r \in (0, 1],\ c_U(0, r] = (0, 1 - r]$.

Now let us recall some basic concepts on direlations from [6]. Let $(U,\mathcal{U})$, $(V, \mathcal{V})$ be texture spaces and let us consider the product texture $\mathcal{P}(U) \otimes \mathcal{V}$ of the texture spaces $(U, \mathcal{P}(U))$ and $(V, \mathcal{V})$ and denote the p-sets and q-sets by $\overline{P}_{(u,v)}$ and $\overline{Q}_{(u,v)}$, respectively. Then

(i) $r \in \mathcal{P}(U) \otimes \mathcal{V}$ is called a *relation* from $(U,\mathcal{U})$ to $(V, \mathcal{V})$ if it satisfies

R1 $r \not\subseteq \overline{Q}_{(u,v)}, P_{u'} \not\subseteq Q_u \implies r \not\subseteq \overline{Q}_{(u',v)}$.

R2 $r \not\subseteq \overline{Q}_{(u,v)} \implies \exists u' \in U$ such that $P_u \not\subseteq Q_{u'}$ and $r \not\subseteq \overline{Q}_{(u',v)}$.

(ii) $R \in \mathcal{P}(U) \otimes \mathcal{V}$ is called a *corelation* from $(U,\mathcal{U})$ to $(V, \mathcal{V})$ if it satisfies

CR1 $\overline{P}_{(u,v)} \not\subseteq R, P_u \not\subseteq Q_{u'} \implies \overline{P}_{(u',v)} \not\subseteq R$.

CR2 $\overline{P}_{(u,v)} \not\subseteq R \implies \exists u' \in U$ such that $P_{u'} \not\subseteq Q_u$ and $\overline{P}_{(u',v)} \not\subseteq R$. A pair (r, R), where r is a relation and R a corelation from $(U,\mathcal{U})$ to $(V, \mathcal{V})$ is called a *direlation from* $(U,\mathcal{U})$ *to* $(V, \mathcal{V})$. The identity direlation (i, I) on $(U,\mathcal{U})$ is defined by $i = \bigvee\{\overline{P}_{(u,u)} \mid u \in U\}$ and $I = \bigcap\{\overline{Q}_{(u,u)} \mid u \in U^\flat\}$ where $U^\flat = \{u \mid U \not\subseteq Q_u\}$. Further, *the inverses* of r and R are defined by

$$r^{\leftarrow} = \bigcap\{\overline{Q}_{(v,u)} \mid r \not\subseteq \overline{Q}_{(u,v)}\} \text{ and } R^{\leftarrow} = \bigvee\{\overline{P}_{(v,u)} \mid \overline{P}_{(u,v)} \not\subseteq R\},$$

respectively where $r^{\leftarrow}$ is a corelation and $R^{\leftarrow}$ is a relation. Further, the direlation $(r,R)^{\leftarrow} = (R^{\leftarrow}, r^{\leftarrow})$ from $(U,\mathcal{U})$ to $(V,\mathcal{V})$ is called the *inverse* of the direlation (r,R). The B-presections with respect to relation and corelation are given as
$r^{\leftarrow}B = \bigvee\{P_u \mid \forall v, r \not\subseteq \overline{Q}_{(u,v)} \Rightarrow P_v \subseteq B\}$, and
$R^{\leftarrow}B = \bigcap\{Q_u \mid \forall v, \overline{P}_{(u,v)} \not\subseteq R \Rightarrow B \subseteq Q_v\}$
for all $B \in \mathcal{V}$, respectively. Now let $(U,\mathcal{U}), (V,\mathcal{V}), (W,\mathcal{W})$ be texture spaces. For any relation p from $(U,\mathcal{U})$ to $(V,\mathcal{V})$ and for any relation q from $(V,\mathcal{V})$ to $(W,\mathcal{W})$ their *composition* $q \circ p$ from $(U,\mathcal{U})$ to $(W,\mathcal{W})$ is defined by

$$q \circ p = \bigvee\{\overline{P}_{(u,w)} \mid \exists\, v \in V \text{ with } p \not\subseteq \overline{Q}_{(u,v)} \text{ and } q \not\subseteq \overline{Q}_{(v,w)}\}$$

and any co-relation P from $(U,\mathcal{U})$ to $(V,\mathcal{V})$ and for any co-relation Q from $(U,\mathcal{U})$ to $(V,\mathcal{V})$ their *composition* $Q \circ P$ from $(U,\mathcal{U})$ to $(V,\mathcal{V})$ defined by

$$Q \circ P = \bigcap\{\overline{Q}_{(u,w)} \mid \exists v \in V \text{ with } \overline{P}_{(u,v)} \not\subseteq P \text{ and } \overline{P}_{(v,w)} \not\subseteq Q\}.$$

Let c_U and c_V be the complementations on $(U,\mathcal{U})$ and $(V,\mathcal{V})$, respectively. The complement r' of the relation r is the corelation

$$r' = \bigcap\{\overline{Q}_{(u,v)} \mid \exists w,z \text{ with } r \not\subseteq \overline{Q}_{(w,z)},\ c_U(Q_u) \not\subseteq Q_w \text{ and } P_z \not\subseteq c_V(P_v)\}.$$

The complement R' of the corelation R is the relation

$$R' = \bigvee\{\overline{P}_{(u,v)} \mid \exists w,z \text{ with } \overline{P}_{(w,z)} \not\subseteq R, P_w \not\subseteq c_U(P_u) \text{ and } c_V(Q_v) \not\subseteq Q_z\}.$$

The complement $(r,R)'$ of the direlation (r,R) is the direlation $(r,R)' = (R', r')$. A direlation (r,R) is called *complemented* if $r = R'$ and $R = r'$. For the basic concepts and results which are not explained in this paper on textures and direlations, we refer to [6, 8].

3 The Category of Approximation Operators

Recall that if (r,R) is a complemented direlation on a complemented texture $(U,\mathcal{U},c_U)$, then the system $(\mathcal{U},\cap,\vee,c_U,r^{\leftarrow},R^{\leftarrow})$ is a textural rough set algebra [8]. For any $A \in \mathcal{U}$, the pair $(r^{\leftarrow}A, R^{\leftarrow}A)$ is called *a textural rough set*. Recall that if $\mathcal{U} = \mathcal{P}(U)$, then $R = (U \times U) \backslash r$ and so using the presections of direlations, we obtain

$$\underline{apr}_r A = r^{\leftarrow}A = U \setminus r^{-1}(U \setminus A) \text{ and } \overline{apr}_r A = R^{\leftarrow}X = r^{-1}(A)$$

for all $A \in \mathcal{P}(U)$ where $R = (U \times U) \setminus r$. The above argument can be also extended to the rough set models on two universes [7,16]. Indeed, if (r,R) is a complemented direlation from $(U,\mathcal{U},c_U)$ to $(V,\mathcal{V},c_V)$, then by Lemma 2.20 (2) in [6], for all $B \in \mathcal{V}$ we may write that

$$c_U R^{\leftarrow}B = r^{\leftarrow}c_V(B) \quad \text{and} \quad c_U r^{\leftarrow}B = R^{\leftarrow}c_V(B).$$

Hence, $R^{\leftarrow}$ and $r^{\leftarrow}$ are dual operators. The proof of the following result is similar to the proof of Theorem 5.7 in [8] and it is omitted.

Theorem 2. *Let L and H be dual operators from $(V,\mathcal{V},c_V)$ to $(U,\mathcal{U},c_U)$. If L satisfies the properties*

$$(\mathbf{L}_1)\ \ L(V)=U \ \ \textit{and}\ \ (\mathbf{L}_2)\ \ L(\bigcap_{j\in J}A_j)=\bigcap_{j\in J}L(A_j),$$

then there exists a complemented direlation (r,R) from $(U,\mathcal{U})$ to $(V,\mathcal{V})$ such that $L(A)=r^{\leftarrow}A$ and $H(A)=R^{\leftarrow}A$ for all $A\in\mathcal{V}$.

Now let $\mathcal{U}=\mathcal{P}(U)$ and $\mathcal{V}=\mathcal{P}(V)$. Then the system $(\mathcal{P}(U),\mathcal{P}(V),\cap,\cup,\underline{apr}_r,\overline{apr}_r)$ will be a rough set model on two universes in the sense of Yao. It is easy to check that

$$\underline{apr}_r Y=R^{\leftarrow}Y=U\setminus r^{-1}(V\setminus Y)\ \ \text{and}\ \ \overline{apr}_r Y=r^{\leftarrow}Y=r^{-1}(Y)$$

for any $Y\subseteq V$.

Lemma 3. *Let U,V,W and Z be universal sets and let $r\subseteq U\times V, q\subseteq V\times W$ and $p\subseteq W\times Z$. Then we have the following statements:*

(i) $\forall C\subseteq W$, $\underline{apr}_{q\circ r}(C)=\underline{apr}_r(\underline{apr}_q(C))$ *and* $\overline{apr}_{q\circ r}(C)=\overline{apr}_r(\overline{apr}_q(C))$

(ii) *For some universe U, let $\Delta_U=\{(u,u)\mid u\in U\}\subseteq U\times U$. Then*

$$\forall A\subseteq U,\ \ \overline{apr}_{r\circ\Delta_U}(A)=\overline{apr}_r(A)\ \textit{and}\ \forall B\subseteq V,\ \ \underline{apr}_{\Delta_V\circ r}(B)=\underline{apr}_r(B).$$

(iii) $\forall D\subseteq Z$, $\underline{apr}_{p\circ(q\circ r)}(D)=\underline{apr}_{(p\circ q)\circ r}(D)$ *and* $\overline{apr}_{p\circ(q\circ r)}(D)=\overline{apr}_{(p\circ q)\circ r}(D)$.

Proof.

(i) $$\begin{aligned}\underline{apr}_{q\circ r}(C)&=U\setminus((q\circ r)^{-1}(W\setminus C))=U\setminus(r^{-1}(q^{-1}(W\setminus C))\\&=U\setminus(\overline{apr}_r(\overline{apr}_q(W\setminus C)))=\underline{apr}_r(V\setminus(\overline{apr}_q(W\setminus C)))\\&=\underline{apr}_r(\underline{apr}_q(C)),\ \text{and}\end{aligned}$$

$\overline{apr}_{q\circ r}(C)=(q\circ r)^{-1}(C)=r^{-1}(q^{-1}(C))=r^{-1}(\overline{apr}_q(C))=\overline{apr}_r(\overline{apr}_q(C))$.

(ii) It is immediate since $r\circ\Delta_U=\Delta_V\circ r=r$.

(iii)
$$\begin{aligned}\underline{apr}_{p\circ(q\circ r)}(D)&=\underline{apr}_{q\circ r}(\underline{apr}_p(D))=\underline{apr}_r(\underline{apr}_q(\underline{apr}_p(D)))\\&=\underline{apr}_r(\underline{apr}_{p\circ q}(D))=\underline{apr}_{(p\circ q)\circ r}(D)\end{aligned}$$

and
$$\begin{aligned}\overline{apr}_{p\circ(q\circ r)}(D)&=\overline{apr}_{q\circ r}(\overline{apr}_p(D))=\overline{apr}_r(\overline{apr}_q(\overline{apr}_p(D))\\&=\overline{apr}_r(\overline{apr}_{p\circ q}(D))=\overline{apr}_{(p\circ q)\circ r}(D).\end{aligned}$$

Corollary 4. (i) *The composition of the pair of rough set approximation operators defined by*

$$(\underline{apr}_q,\overline{apr}_q)\circ(\underline{apr}_r,\overline{app}_r)=(\underline{apr}_{q\circ r},\overline{apr}_{q\circ r})$$

is associative.

(ii) $(\underline{apr}_r,\overline{app}_r)\circ(\underline{app}_{\Delta_U},\overline{app}_{\Delta_U})=(\underline{app}_{\Delta_V},\overline{app}_{\Delta_V})\circ(\underline{apr}_r,\overline{app}_r)=(\underline{apr}_r,\overline{app}_r)$.

Proof. (i) By Lemma 3 (iii), we have

$$(\underline{apr}_p, \overline{apr}_p) \circ ((\underline{apr}_q, \overline{apr}_q) \circ (\underline{apr}_r, \overline{apr}_r)) = (\underline{apr}_p, \overline{apr}_p) \circ (\underline{apr}_{q\circ r}, \overline{apr}_{q\circ r})$$
$$= (\underline{apr}_{p\circ(q\circ r)}, \overline{apr}_{p\circ(q\circ r)}) = (\underline{apr}_{(p\circ q)\circ r}, \overline{apr}_{(p\circ q)\circ r}) = (\underline{apr}_{p\circ q}, \overline{apr}_{p\circ q}) \circ (\underline{apr}_r, \overline{apr}_r)$$
$$= ((\underline{apr}_p, \overline{apr}_p) \circ (\underline{apr}_q, \overline{apr}_q)) \circ (\underline{apr}_r, \overline{apr}_r).$$

(ii) It is immediate by Lemma 3 (ii).

Corollary 5. *The pairs of rough set approximation operators between sets form a category which is denoted by* **R-APR**.

Theorem 6. *The contravariant functor* $\mathfrak{T}$*:* **Rel**$\to$ **R-APR** *defined by*

$$\mathfrak{T}(U) = U \quad \textit{and} \quad \mathfrak{T}(r) = (\underline{apr}_r, \overline{apr}_r)$$

for all sets U, V *and* $r \subseteq U \times V$ *is an isomorphism.*

Proof. For any object $(U,\mathcal{U})$, the pair $\mathrm{id}_U = (\underline{apr}_{\Delta_U}, \overline{apr}_{\Delta_U})$ is an identity morphism in the category of **R-APR** and $\mathfrak{T}(\Delta_U) = (\underline{apr}_{\Delta_U}, \overline{apr}_{\Delta_U})$. Further, $\mathfrak{T}(q \circ r) = \mathfrak{T}(r) \circ \mathfrak{T}(q)$ and so indeed $\mathfrak{T}$ is a functor. Let U and V be any two sets, and r, q be direlations from U to V where $r \neq q$. Suppose that $(u, v) \in r$ and $(u, v) \notin q$ for some $(u, v) \in U \times V$. Then we have $u \in r^{-1}(\{v\}) = \overline{apr}_r(\{v\})$ and $u \notin q^{-1}(\{v\}) = \overline{apr}_q(\{v\})$ and this gives $(\underline{apr}_r, \overline{apr}_r) \neq (\underline{apr}_q, \overline{apr}_q)$. Conversely, if $(\underline{apr}_r, \overline{apr}_r) \neq (\underline{apr}_q, \overline{apr}_q)$, then we have $\underline{apr}_r(B) \neq \underline{apr}_q(B)$ or $\overline{apr}_r(B) \neq \overline{apr}_q(B)$ for some $B \subseteq V$. With no loss of generality, if $\overline{apr}_r(B) \neq \overline{apr}_q(B)$, then $r^{-1}(B) \neq q^{-1}(B)$ and so clearly, $r \neq q$. Therefore, the functor $\mathfrak{F}$ is bijective on hom-sets. Clearly, it is also bijective on objects. □

4 The Category of Textures and Direlations

By Proposition 2.14 in [6], direlations are closed under compositions and the composition is associative. For any texture $(U,\mathcal{U})$, we have the identity direlation (i_U, I_U) on $(U,\mathcal{U})$. If (r, R) is a direlation from $(U,\mathcal{U})$ to $(V,\mathcal{V})$, then

$$(i_V, I_V) \circ (r, R) = (r, R) \text{ and } (r, R) \circ (i_U, I_U) = (r, R).$$

Now we may give:

Theorem 7. *Texture spaces and direlations form a category which is denoted by* **drTex**.

Let $(U,\mathcal{U}, c_U)$ and $(V,\mathcal{V}, c_V)$ be complemented textures, and (r, R) a complemented direlation from $(U,\mathcal{U})$ to $(V,\mathcal{V})$. If (q, Q) is a complemented direlation from $(V,\mathcal{V}, c_V))$ to $(Z,\mathcal{Z}, c_Z)$, then by Proposition 2.21 (3) in [6], the composition of (r, R) and (q, Q) is also complemented, that is

$$(q \circ r)' = q' \circ r' = Q \circ R \text{ and } (Q \circ R)' = Q' \circ R' = q \circ r.$$

Since the identity direlation (i_U, I_U) is also complemented, then we have the following result:

Theorem 8. *Complemented texture spaces and complemented direlations form a category which is denoted by* **cdrTex**.

Theorem 9
(i) *The functor* $\mathfrak{R} : \mathbf{R\text{-}APR} \to \mathbf{cdrTex}$ *defined by*

$$\mathfrak{R}(U) = (U, \mathcal{P}(U)), \quad \mathfrak{R}(\underline{apr}_r, \overline{apr}_r) = (R^{\leftarrow}, r^{\leftarrow})$$

for every morphism $(\underline{apr}_r, \overline{apr}_r) : U \to V$ *in* **R-APR** *where*

$$R^{\leftarrow} = ((U \times V) \setminus r)^{-1} \text{ and } r^{\leftarrow} = r^{-1}$$

is a full embedding.
(ii) *The functor* $\mathfrak{D} : \mathbf{Rel} \to \mathbf{cdrTex}$ *defined by*

$$\mathfrak{D}(U) = (U, \mathcal{P}(U)), \ \mathfrak{D}(r) = (r, R)$$

for every morphism $r : U \to V$ *in* **Rel** *where* $R = (U \times V) \setminus r$ *is a full embedding.*

Proof. The functors $\mathfrak{R}$ and $\mathfrak{D}$ are injective on objects and hom-sets. Further, if (r, R) is a complemented direlation from $(U, \mathcal{P}(U))$ to $(V, \mathcal{P}(V))$, then r is a relation from U to V. Further, $(\underline{apr}_r, \overline{apr}_r)$ is a pair of rough set approximation operators from V to U. □

5 Dagger Symmetric Monoidal Categories

Dagger symmetric monoidal categories play a central role in the abstract quantum mechanics [1,15]. The primary examples are the categories **Rel** of relations and sets, and **FdHilb** of finite dimensional Hilbert spaces and linear mappings. Since **Rel** and **R-APR** are isomorphic categories, then **R-APR** is also a dagger symmetric monoidal category. In this section, we show that the category **cdrTex** is also a dagger symmetric monoidal category.

Definition 10. (i) *A dagger category* [15] *is a category* **C** *together with an involutive, identity on objects, contravariant functor* $\dagger : \mathbf{C} \to \mathbf{C}$. *In other words, every morphism* $f : A \to B$ *in* **C** *corresponds to a morphism* $f^{\dagger} : B \to A$ *such that for all* $f : A \to B$ *and* $g : B \to C$ *the following conditions hold:*

$$id_A^{\dagger} = id_A : A \to A, \ (g \circ f)^{\dagger} = f^{\dagger} \circ g^{\dagger} : C \to A, \text{ and } f^{\dagger\dagger} = f : A \to A.$$

(ii) *A symmetric monoidal category* [14] *is a category* **C** *together with a bifunctor* $\otimes$, *a distinguish object* I, *and natural isomorphisms* $\alpha_{A,B,C} : (A \otimes B) \otimes C \to A \otimes (B \otimes C), \lambda_A : A \to I \otimes A, \ \sigma_{A,B} : A \otimes B \to B \otimes A$ *subject to standart coherence conditions.*
(iii) *A dagger symmetric monoidal category* [15] *is a symmetric monoidal category* **C** *with a dagger structure preserving the symmetric monoidal structure:*

For all $f : A \to B$ *and* $g : C \to D$, $(f \otimes g)^{\dagger} = f^{\dagger} \otimes g^{\dagger} : B \otimes D \to A \otimes C$,
$\alpha_{A,B,C}^{\dagger} = \alpha_{A,B,C}^{-1} : A \otimes (B \otimes C) \to (A \otimes B) \otimes C$, $\lambda^{\dagger} = \lambda^{-1} : I \otimes A \to A$,
$\sigma_{A,B}^{\dagger} = \sigma_{A,B}^{-1} : B \otimes A \to A \otimes B$.

Theorem 11. *The categories* **drTex** *and* **cdrTex** *are dagger categories.*

Proof. First let us determine the dagger structure on **drTex**. Note that $\dagger : \mathbf{drTex} \to \mathbf{drTex}$ is a functor defined by

$$\dagger(U,\mathcal{U}) = (U,\mathcal{U}) \text{ and } \dagger(r,R) = (r,R)^{\leftarrow}$$

for all $(U,\mathcal{U}) \in \mathrm{Ob}\mathbf{drTex}$ and $(r,R) \in \mathrm{Mor}\mathbf{drTex}$. Further, we have

(i) $\forall (U,\mathcal{U}),\ (i_U, I_U)^{\leftarrow} = (i_U, I_U)$,

(ii) $((q,Q)\circ(r,R))^{\leftarrow} = (r,R)^{\leftarrow}\circ(q,Q)^{\leftarrow}$, and

(iii) $(((r,R)^{\leftarrow})^{\leftarrow} = (r,R)$.

Therefore, **drTex** is a dagger category. On the other hand, if (r,R) is complemented, then $(r,R)^{\leftarrow} = (R^{\leftarrow}, r^{\leftarrow})$ is also complemented. Indeed, by Proposition 2.21 in [6],

$$(R^{\leftarrow})' = (R')^{\leftarrow} = r^{\leftarrow} \text{ and } (r^{\leftarrow})' = (r')^{\leftarrow} = R^{\leftarrow}.$$

As a result, the category **cdrTex** is also a dagger category. □

Corollary 12. *The diagram*

$$\begin{array}{ccc} \mathbf{Rel} & \xrightarrow{\ \mathfrak{T}\ } & \mathbf{R-APR} \\ \downarrow{\scriptstyle \mathfrak{D}} & & \downarrow{\scriptstyle \mathfrak{R}} \\ \mathbf{cdrTex} & \xrightarrow{\ \dagger\ } & \mathbf{cdrTex} \end{array}$$

commutes.

Proof. Let $r : U \to V$ be a morphism in **Rel**. If we take $R = (U \times V) \setminus r$, then

$$\begin{aligned} (\dagger\circ\mathfrak{D})(r) = \dagger(\mathfrak{D}(r)) &= \dagger(r,R) = (r,R)^{\leftarrow} = (R^{\leftarrow}, r^{\leftarrow}) \\ &= \mathfrak{R}(\underline{apr}_r,\ \overline{apr}_r) = \mathfrak{R}(\mathfrak{T}(r)) = (\mathfrak{R}\circ\mathfrak{T})(r). \end{aligned}$$ □

The proofs of Lemmas 13 and 15, and Proposition 14 can be easily given using textural arguments.

Lemma 13. (i) *Let ψ be a textural isomorphism from $(U,\mathcal{U})$ to $(V,\mathcal{V})$. Then the direlation (r_ψ, R_ψ) from $(U,\mathcal{U})$ to $(V,\mathcal{V})$ defined by*

$$r_\psi = \bigvee\{\overline{P}_{(u,v)} \mid P_{\psi(u)} \not\subseteq Q_v\} \text{ and } R_\psi = \bigcap\{\overline{Q}_{(u,v)} \mid P_v \not\subseteq Q_{\psi(u)}\}$$

is an isomorphism in **drTex**.

(ii) *Let c_U and c_V are complementations on the textures $(U,\mathcal{U})$ and $(V,\mathcal{V})$, respectively. If ψ is a complemented textural isomorphism, then (r_ψ, R_ψ) is also complemented.*

Proposition 14. (i) *Let* $(U,\mathcal{U}),(V,\mathcal{V})$ *and* $(W,\mathcal{W})$ *be texture spaces. Then*

$$((U\times V)\times W,\ (\mathcal{U}\otimes\mathcal{V})\otimes\mathcal{W})\cong(U\times(V\times W),\ \mathcal{U}\otimes(\mathcal{V}\otimes\mathcal{W})).$$

(ii) *Take the texture* $(E,\mathcal{E})$ *where* $E=\{e\}$ *and* $\mathcal{E}=\{\{e\},\emptyset\}$. *Then for any texture* $(U,\mathcal{U})$ *we have*

$$(U\times E,\ \mathcal{U}\otimes\mathcal{E})\cong(U,\mathcal{U})\ \textit{and}\ (E\times U,\ \mathcal{E}\otimes\mathcal{U})\cong(U,\mathcal{U}).$$

(iii) $(U\times V,\ \mathcal{U}\otimes\mathcal{V})\cong(V\times U,\ \mathcal{V}\otimes\mathcal{U})$.

Let (r,R) be a direlation from $(U,\mathcal{U},c_U)$ to $(V,\mathcal{V},c_V)$ and (q,Q) be a direlation from $(W,\mathcal{W},c_W)$ to $(Z,\mathcal{Z},c_Z)$. Then the direlation

$$(r\times q,\ R\times Q):(U\times W,\ \mathcal{U}\otimes\mathcal{W})\to(V\times Z,\ \mathcal{V}\otimes\mathcal{Z})$$

is defined by

$r\times q=\bigvee\{\overline{P}_{(u,w),(v,z))}\mid r\not\subseteq\overline{Q}_{(u,v)}$ and $q\not\subseteq\overline{Q}_{(w,z)}\}$, and

$R\times Q=\bigvee\{\overline{Q}_{(u,w),(v,z))}\mid\overline{P}_{(u,v)}\not\subseteq R$ and $\overline{Q}_{(w,z)}\not\subseteq Q\}$ [4].

Lemma 15. *We have the following equalities :*

(i) $(r\times q)'=r'\times q'$ *and* $(R\times Q)'=R'\times Q'$.

(ii) $(r\times q)^{\leftarrow}=r^{\leftarrow}\times q^{\leftarrow}$ *and* $(R\times Q)^{\leftarrow}=R^{\leftarrow}\times Q^{\leftarrow}$.

Corollary 16. *If* (r,R) *and* (q,Q) *are complemented direlations, then*

$$(r\times q,R\times Q)$$

is also a complemented direlation.

Corollary 17. *The mapping* $\otimes:\mathbf{cdrTex}\times\mathbf{cdrTex}\longrightarrow\mathbf{cdrTex}$ *defined by*

$$\otimes((U,\mathcal{U}),(V,\mathcal{V}))=(U\times V,\ \mathcal{U}\otimes\mathcal{V})\ \textit{and}\ \otimes((r,R),(q,Q))=(r\times q,R\times Q),$$

is a functor.

Corollary 18. (i) *For the functors*

$$\mathfrak{F},\mathfrak{B}:\mathbf{cdrTex}\times\mathbf{cdrTex}\times\mathbf{cdrTex}\to\mathbf{cdrTex}$$

defined by

$$\mathfrak{F}((U,\mathcal{U}),(V,\mathcal{V}),(W,\mathcal{W}))=(U\times(V\times W),\ \mathcal{U}\otimes(\mathcal{V}\otimes\mathcal{W})),$$

$$\mathfrak{F}((p,P),(q,Q),(r,R))=(p\times(q\times r),\ P\times(Q\times R))$$

and

$$\mathfrak{B}((U,\mathcal{U}),(V,\mathcal{V}),(W,\mathcal{W}))=((U\times V)\times W,\ (\mathcal{U}\otimes\mathcal{V})\otimes\mathcal{W}),$$

$$\mathfrak{B}((p,P),(q,Q),(r,R))=((p\times q)\times r,\ (P\times Q)\times R),$$

respecively, there exists a natural transformation $\alpha : \mathfrak{F} \to \mathfrak{B}$ *with the component*

$$\alpha_{(\mathcal{U},\mathcal{V},\mathcal{W})} : ((U \times V) \times W,\ (\mathcal{U} \otimes \mathcal{V}) \otimes \mathcal{W}) \cong (U \times (V \times W),\ \mathcal{U} \otimes (\mathcal{V} \otimes \mathcal{W}))$$

which is a natural isomorphism.

(ii) *Take the functors* $\mathfrak{R}, \mathfrak{D} : \mathbf{cdrTex} \to \mathbf{cdrTex}$ *defined by*

$\mathfrak{R}((U,\mathcal{U})) = (U \times E,\ \mathcal{U} \otimes \mathcal{E})$ $\qquad$ $\mathfrak{D}((U,\mathcal{U})) = (E \times U,\ \mathcal{E} \otimes \mathcal{U})$

$\mathfrak{R}((r,R)) = (r \times i_E,\ R \times I_E)$, *and* $\qquad$ $\mathfrak{D}((r,R)) = (i_E \times r,\ I_E \times R)$.

Then there exist the natural transformations $\lambda : \mathfrak{R} \to \mathfrak{I}_{\mathbf{cdrTex}}$ *and* $\rho : \mathfrak{D} \to \mathfrak{I}_{\mathbf{cdrTex}}$ *such that for all* $(U,\mathcal{U})$*, the components*

$$\lambda_{(U,\mathcal{U})} : (U \times E,\ \mathcal{U} \otimes \mathcal{E}) \cong (U,\mathcal{U}) \text{ and } \rho_{(U,\mathcal{U})} : (E \times U,\ \mathcal{E} \otimes \mathcal{U}) \cong (U,\mathcal{U}).$$

are natural isomorphisms.

(ii) *Consider the functors* $\mathfrak{S}, \mathfrak{U} : \mathbf{cdrTex} \times \mathbf{cdrTex} \to \mathbf{cdrTex}$ *defined by*

$\mathfrak{S}((U,\mathcal{U}),(V,\mathcal{V})) = (U \times V,\ \mathcal{U} \otimes \mathcal{V})$ $\qquad$ $\mathfrak{U}((U,\mathcal{U}),(V,\mathcal{V})) = (V \times U,\ \mathcal{V} \otimes \mathcal{U})$

$\mathfrak{S}((r,R),(q,Q)) = (r \times q,\ R \times Q)$, *and* $\qquad$ $\mathfrak{U}((r,R),(q,Q)) = (q \times r,\ Q \times R)$.

Then there exist a natural transformation $\sigma : \mathfrak{S} \to \mathfrak{U}$ *such that for all* $(U,\mathcal{U})$*, the component* $\sigma_{(\mathcal{U},\mathcal{V})} : (U \times V,\ \mathcal{U} \otimes \mathcal{V}) \cong (V \times U, \mathcal{V} \otimes \mathcal{U})$ *is a natural isomorphism.*

Proof. It is immediate by Proposition 14. □

Lemma 19. *Mac Lane's associativity and unit coherence conditions hold* [14]*:*

(i) *The following pentagonal diagram commutes:*

$$\begin{array}{ccccc}
((\mathcal{U} \otimes \mathcal{V}) \otimes \mathcal{W}) \otimes \mathcal{Z} & \xrightarrow{\alpha_{(\mathcal{U},\mathcal{V},\mathcal{W}) \otimes \mathcal{Z}}} & (\mathcal{U} \otimes (\mathcal{V} \otimes \mathcal{W})) \otimes \mathcal{Z} & \xrightarrow{\alpha_{(\mathcal{U},\mathcal{V} \otimes \mathcal{W},\mathcal{Z})}} & \mathcal{U} \otimes ((\mathcal{V} \otimes \mathcal{W}) \otimes \mathcal{Z}) \\
\downarrow{\scriptstyle \alpha_{(\mathcal{U} \otimes \mathcal{V},\mathcal{W},\mathcal{Z})}} & & & & \downarrow{\scriptstyle \mathcal{U} \otimes \alpha_{(\mathcal{V},\mathcal{W},\mathcal{Z})}} \\
(\mathcal{U} \otimes \mathcal{V}) \otimes (\mathcal{W} \otimes \mathcal{Z}) & & \xrightarrow{\alpha_{(\mathcal{U},\mathcal{V},\mathcal{W} \otimes \mathcal{Z})}} & & (\mathcal{U} \otimes (\mathcal{V} \otimes (\mathcal{W} \otimes \mathcal{Z}))
\end{array}$$

(ii) *The following diagram is commutative.*

$$\begin{array}{ccc}
(\mathcal{U} \otimes \mathcal{E}) \otimes \mathcal{V} & \xrightarrow{\alpha_{(\mathcal{U},\mathcal{E},\mathcal{V})}} & \mathcal{U} \otimes (\mathcal{E} \otimes \mathcal{V}) \\
& {\scriptstyle \rho_{\mathcal{U}} \otimes \mathcal{V}} \searrow \quad \swarrow {\scriptstyle \mathcal{U} \otimes \lambda_{\mathcal{V}}} & \\
& \mathcal{U} \otimes \mathcal{V} &
\end{array}$$

Proof. Immediade.

Corollary 20. *The category* **cdrTex** *is a dagger symmetric monoidal category.*

Acknowledgements. This work has been supported by the Turkish Scientific and Technological Research Organization under the project TBAG 109T683.

The author sincerely thanks the reviewers for their valuable comments that improve the presentation of the paper.

References

1. Abramsky, S., Coecke, B.: A categorical semantics of quantum protocols. In: Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, LICS, pp. 415–425. IEEE Computer Society Press, Los Alamitos (2004)
2. Banerjee, M., Chakraborty, M.K.: A category for rough sets. Foundations of Computing and Decision Sciences 18(3-4), 167–188 (1983)
3. Banerjee, M., Yao, Y.: A categorial basis for granular computing. In: An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.) RSFDGrC 2007. LNCS (LNAI), vol. 4482, pp. 427–434. Springer, Heidelberg (2007)
4. Brown, L.M., Irkad, A.: Binary di-operations and spaces of real difunctions on a texture. Hacettepe Journal of Mathematics and Statistics 37(1), 25–39 (2008)
5. Brown, L.M., Diker, M.: Ditopological texture spaces and intuitionistic sets. Fuzzy Sets and Systems (98), 217–224 (1998)
6. Brown, L.M., Ertürk, R., Dost, Ş.: Ditopological texture spaces and fuzzy topology, I. Basic Concepts. Fuzzy Sets and Systems 147, 171–199 (2004)
7. Daowu, P., Xu, Z.: Rough set models on two universes. International Journal of General Systems 33(5), 569–581 (2004)
8. Diker, M.: Textural approach to rough sets based on relations. Information Sciences 180(8), 1418–1433 (2010)
9. Eklund, P., Galán, M.Á.: 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)
10. Eklund, P., Galán, M.A.: The rough powerset monad. J. Mult.-Valued Logic Soft Comput. 13(4-6), 321–333 (2007)
11. Eklund, P., Galán, M.A., Gähler, W.: Partially Ordered Monads for Monadic Topologies, Rough Sets and Kleene Algebras. Electronic Notes in Theoretical Computer Science 225, 67–81 (2009)
12. Iwinski, T.B.: Algebraic approach to rough sets. Bull. Pol. Ac. Math. 35(6), 673–683 (1987)
13. Li, X., Yuan, X.: The category RSC of I-rough sets. In: 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD, vol. 1, pp. 448–452 (2008)
14. Mac Lane, S.: Categories for the working matematician. Springer Graduate Text in Mathematics 5 (1971)
15. Selinger, P.: Dagger compact closed categories and completely positive maps. Electronic Notes in Theoretical Computer Science 170, 139–163 (2007)
16. Yao, Y.: Constructive and algebraic methods of the theory of rough sets. Information Sciences 109, 21–47 (1998)

Approximations and Classifiers*

Andrzej Skowron[1] and Jarosław Stepaniuk[2]

[1] Institute of Mathematics, The University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl
[2] Department of Computer Science, Białystok University of Technology
Wiejska 45A, 15-351 Białystok, Poland
j.stepaniuk@pb.edu.pl

Abstract. We discuss some important issues for applications that are related to generalizations of the 1994 approximation space definition [11]. In particular, we present examples of rough set based strategies for extension of approximation spaces from samples of objects onto the whole universe of objects. This makes it possible to present methods for inducing approximations of concepts or classifications analogously to the approaches for inducing classifiers known in machine learning or data mining.

Keywords: approximation space, lower approximation, upper approximation, rough sets, extension of approximation space, classifiers.

1 Introduction

A rough set, first described by Z. Pawlak, is a pair of sets which give the lower and the upper approximation of the original set. In the standard version of rough set theory an approximation space is based on indiscernibility equivalence relation. Approximation spaces belong to the broad spectrum of basic subjects investigated in rough set theory (see, *e.g.*, [1,9,11,12,15,14,16,17]). Over the years different aspects of approximation spaces were investigated and many generalizations of the approach based on indiscernibility equivalence relation [7] were proposed. In this paper, we discuss some aspects of generalizations of approximation spaces investigated in [11,12,16] that are important from an application point of view, e.g., in searching for approximation of complex concepts (see, e.g., [1]).

2 Attributes, Signatures of Objects and Two Semantics

In [7] any attribute a is defined as a function from the universe of objects U into the set of attribute values V_a. However, in applications we expect that the value

* The research has been partially supported by the grants N N516 077837, N N516 069235, and N N516 368334 from Ministry of Science and Higher Education of the Republic of Poland.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 297–306, 2010.

of attribute should be also defined for objects from extensions of U, *i.e.*, for new objects which can be perceived in the future[1]. The universe U is only a sample of possible objects. This requires some modification of the basic definitions of attribute and signatures of objects.

We assume that for any attribute a under consideration there is given a relational structure R_a. Together with the simple structure $(V_a, =)$ [7], some other relational structures R_a with the carrier V_a for $a \in A$ and a signature τ are considered. We also assume that with any attribute a is identified a set of some generic formulas $\{\alpha_i\}_{i\in J}$ (where J is a set of indexes) interpreted over R_a as a subsets of V_a, i.e., $\|\alpha_i\|_{R_a} = \{v \in V_a : R_a, v \models \alpha_i\}$. Moreover, it is assumed that the set $\{\|\alpha_i\|_{R_a}\}_{i\in J}$ is a partition of V_a. Perception of an object u by a given attribute a is represented by selection of a formula α_i and a value $v \in V_a$ such that $v \in \|\alpha_i\|_{R_a}$. Using an intuitive interpretation one can say that such a pair (α_i, v) is selected from $\{\alpha_i\}_{i\in J}$ and V_a, respectively, as the result of sensory measurement. We assume that for a given set of attributes A and any object u the signature of u relative to A is given by $Inf_A(u) = \{(a, \alpha_u^a, v) : a \in A\}$, where (α_u^a, v) is the result of sensory measurement by a on u.

Let us observe that a triple (a, α_u^a, v) can be encoded by the atomic formula $a = v$ with interpretation

$$\|a = v\|_{U^*} = \{u \in U^* : (a, \alpha_u^a, v) \in Inf_a(u) \text{ for some } \alpha_u^a\}.$$

One can also consider a soft version of the attribute definition. In this case, we assume that the semantics of the family $\{\alpha_i\}_{i\in J}$ is given by fuzzy membership functions for α_i and the set of these functions define a fuzzy partition.

We construct granular formulas from atomic formulas corresponding to the considered attributes. In the consequence, the satisfiability of such formulas is defined if the satisfiability of atomic formulas is given as the result of sensor measurement. Hence, one can consider for any constructed formula α over atomic formulas its semantics $\|\alpha\|_U \subseteq U$ over U as well as the semantics $\|\alpha\|_U^* \subseteq U^*$ over U^*, where $U \subseteq U^*$. The difference between these two cases is the following. In the case of U, one can compute $\|\alpha\|_U \subseteq U$ but in the case $\|\alpha\|_{U^*} \subseteq U^*$, we only know that this set is well defined. However, we can compute the satisfiability of α for objects $u \in U^* \setminus U$ only after the relevant sensory measurements on u are performed resulting in selection of the satisfied atomic formulas for a given object. It is worthwhile mentioning that one can use some methods for estimation of relationships among semantics of formulas over U^* using the relationships among semantics of these formulas over U. For example, one can apply statistical methods. This step is crucial in considerations on extensions of approximation spaces relevant for inducing classifiers from data (see, e.g., [1,14]).

3 Uncertainty Function

In [11,12,16] the uncertainty function defines for every object u, a set of objects described similarly to x. The set $I(u)$ is called the neighborhood of u.

[1] Objects from U are treated as labels of real perceived objects.

In this paper, we propose uncertainty functions of the form $I : U^* \longrightarrow P_\omega(U^*)$, where $P_\omega(U^*) = \bigcup_{i \geq 1} P^i(U^*)$, $P^1(U^*) = P(U^*)$ and $P^{i+1}(U^*) = P(P^i(U^*)))$ for $i \geq 1$. The values of uncertainty functions are called granular neighborhoods. These granular neighborhoods are defined by the so called granular formulas. The values of such uncertainty functions are not necessarily from $P(U^*)$ but from $P_\omega(U^*)$. In the following sections, we will present more details on granular neighborhoods and granular formulas. Figure 1 presents an illustrative example of the uncertainty function with values in $P^2(U^*)$ rather than in $P(U^*)$.

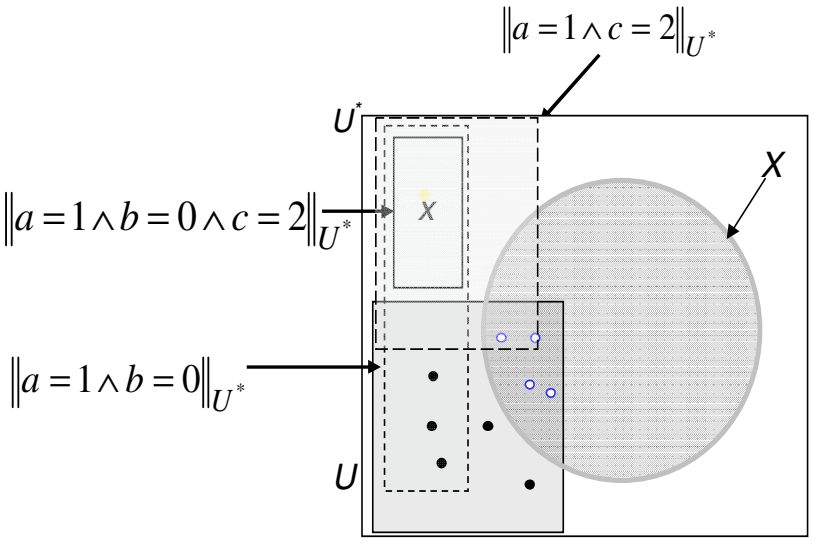


Fig. 1. Uncertainty function $I : U^* \rightarrow P^2(U^*)$. The neighborhood of $x \in U^* \setminus U$, where $Inf_A(x) = \{(a, 1), (b, 0), (c, 2)\}$, does not contain training cases from U. The generalizations of this neighborhood have non empty intersections with U.

If $X \in P_\omega(U^*)$ and $U \subseteq U^*$ then by $X \restriction U$ we denote the set defined as follows (i) if $X \in P(U^*)$ then $X \restriction U = X \cap U$ and (ii) for any $i \geq 1$ if $X \in P^{i+1}(U^*)$ then $X \restriction U = \{Y \restriction U : Y \in X\}$.

4 Rough Inclusion Function

The second component of any approximation space is the rough inclusion function [12], [16].

One can consider general constraints which the rough inclusion functions should satisfy. In this section, we present only some examples of rough inclusion functions.

The rough inclusion function $\nu : P(U) \times P(U) \rightarrow [0,1]$ defines the degree of inclusion of X in Y, where $X, Y \subseteq U$.[2]

In the simplest case the standard rough inclusion function can be defined by (see, e.g., [12], [7]):

$$\nu_{SRI}(X,Y) = \begin{cases} \frac{card(X \cap Y)}{card(X)} & \text{if } X \neq \emptyset \\ 1 & \text{if } X = \emptyset. \end{cases} \quad (1)$$

Some illustrative example is given in Table 1.

Table 1. Illustration of Standard Rough Inclusion Function

X	Y	$\nu_{SRI}(X,Y)$
$\{x_1, x_3, x_7, x_8\}$	$\{x_2, x_4, x_5, x_6, x_9\}$	0
$\{x_1, x_3, x_7, x_8\}$	$\{x_1, x_2, x_4, x_5, x_6, x_9\}$	0.25
$\{x_1, x_3, x_7, x_8\}$	$\{x_1, x_2, x_3, x_7, x_8\}$	1

It is important to note that an inclusion measure expressed in terms of the confidence measure, widely used in data mining, was considered by Łukasiewicz [4] long time ago in studies on assigning fractional truth values to logical formulas.

The rough inclusion function was generalized in rough mereology [10]. For definition of inclusion function for more general granules, e.g., partitions of objects one can use measure based on positive region [7], entropy or rough entropy [6,5]. Inclusion measures for more general granules were also investigated [13,2]. However, more work in this direction should be done, especially on inclusion of granules with complex structures, in particular for granular neighborhoods.

5 Approximation Spaces

In this section we present a generalization of definition of approximation space from [11,12,16]).

Definition 1. *An approximation space over a set of attributes A is a system*

$$AS = (U, L, I, \nu, LOW, UPP),$$

where

- *U is a sample of objects with known signatures relative to a given set of attributes A,*
- *L is a language of granular formulas defined over atomic formulas corresponding to generic formulas from signatures (see Section 3),*

[2] We assume that U is finite.

- $I : U^* \to P_\omega(U^*)$ *is an uncertainty function, where* $U^* \supseteq U$ *and the set* U^* *is such that for any object* $u \in U^*$ *the signature* $Inf_A(u)$ *of* u *relative to* A *can be obtained (as the result of sensory measurements on* u*); we assume that the granular neighborhood* $I(u)$ *is computable from* $Inf_A(u)$*, i.e.,* $I(u)$ *is defined by a granular formula* α *selected from* L[3]*.*
- $\nu : P_\omega(U^*) \times P_\omega(U^*) \to [0,1]$ *is a rough inclusion function,*
- LOW *and* UPP *are the lower approximation operation and the upper approximation operation, respectively, defined on elements from* $P_\omega(U^*)$ *with values in* $P_\omega(U^*)$ *such that*
  1. $\nu(LOW(AS,X), UPP(AS,X)) = 1$ *for any* $X \in P_\omega(U^*)$*,*
  2. $LOW(AS,X) \restriction U$ *is included in* $X \restriction U$ *to a degree at least deg, i.e.,* $\nu(LOW(AS,X) \restriction U, X \restriction U)) \geq deg$ *for any* $X \in P_\omega(U^*)$*,*
  3. $X \restriction U$ *is included in* $UPP(AS,X) \restriction U$ *to a degree at least deg, i.e.,* $\nu(X \restriction U, UPP(AS,X) \restriction U) \geq deg$ *for any* $X \in P_\omega(U^*)$*,*

  where deg is a given threshold from the interval $[0,1]$*.*

5.1 Approximations and Decision Rules

In this section we discuss generation of approximations on extensions of samples of objects.

In the example we illustrate how the approximations of sets can be estimated using only partial information about these sets. Moreover, the example introduces uncertainty functions with values in $P^2(U)$ and rough inclusion functions defined for sets from $P^2(U)$.

Let us assume $DT = (U, A \cup \{d\})$ be a decision table, where $U = \{x_1, \ldots, x_9\}$ is a set of objects and $A = \{a, b, c\}$ is a set of condition attributes (see Table 2).

Table 2. Decision table over the set of objects U

	a	b	c	d
x_1	1	1	0	1
x_2	0	2	0	1
x_3	1	0	1	0
x_4	0	2	0	1
x_5	0	1	0	1
x_6	0	0	0	0
x_7	1	0	2	0
x_8	1	2	1	0
x_9	0	0	1	0

There are two decision reducts: $\{a, b\}$ and $\{b, c\}$. We obtain the set $Rule\_set = \{r_1, \ldots, r_{12}\}$ of minimal (reduct based) [7] decision rules.

[3] For example, the granule $\alpha = \{\{\alpha_1, \alpha_2\}, \{\alpha_3, \alpha_4\}\}$, where $\alpha_i \in L$ for $i = 1, \ldots, 4$, defines the set $\{\{\|\alpha_1\|_{U^*}, \|\alpha_2\|_{U^*}\}, \{\|\alpha_3\|_{U^*}, \|\alpha_4\|_{U^*}\}\}$ and $\alpha \restriction U = \{\{\|\alpha_1\|_U, \|\alpha_2\|_U\}, \{\|\alpha_3\|_U, \|\alpha_4\|_U\}\}$.

From x_1 we obtain two rules:
r_1 : **if** $a = 1$ **and** $b = 1$ **then** $d = 1$, r_2 : **if** $b = 1$ **and** $c = 0$ **then** $d = 1$.
From x_2 and x_4 we obtain two rules:
r_3 : **if** $a = 0$ **and** $b = 2$ **then** $d = 1$, r_4 : **if** $b = 2$ **and** $c = 0$ **then** $d = 1$.
From x_5 we obtain one new rule:
r_5 : **if** $a = 0$ **and** $b = 1$ **then** $d = 1$.
From x_3 we obtain two rules:
r_6 : **if** $a = 1$ **and** $b = 0$ **then** $d = 0$, r_7 : **if** $b = 0$ **and** $c = 1$ **then** $d = 0$.
From x_6 we obtain two rules:
r_8 : **if** $a = 0$ **and** $b = 0$ **then** $d = 0$, r_9 : **if** $b = 0$ **and** $c = 0$ **then** $d = 0$.
From x_7 we obtain one new rule:
r_{10} : **if** $b = 0$ **and** $c = 2$ **then** $d = 0$.
From x_6 we obtain two rules:
r_{11} : **if** $a = 1$ **and** $b = 2$ **then** $d = 0$, r_{12} : **if** $b = 2$ **and** $c = 1$ **then** $d = 0$.
Let $U^* = U \cup \{x_{10}, x_{11}, x_{12}, x_{13}, x_{14}\}$ (see Table 3).

Table 3. Decision table over the set of objects $U^* - U$

	a	b	c	d	d_{class}
x_{10}	0	2	1	1	1 from r_3 or 0 from r_{12}
x_{11}	1	2	0	0	1 from r_4 or 0 from r_{11}
x_{12}	1	2	0	0	1 from r_4 or 0 from r_{11}
x_{13}	0	1	2	1	1 from r_5
x_{14}	1	1	2	1	1 from r_1

Let $h : [0, 1] \to \{0, 1/2, 1\}$ be a function defined by

$$h(t) = \begin{cases} 1 & \text{if } t > 1/2 \\ 1/2 & \text{if } t = 1/2 \\ 0 & \text{if } t < 1/2. \end{cases} \tag{2}$$

Below we present an example of the uncertainty and rough inclusion functions:

$$I(x) = \{\|lh(r)\|_{U^*} : x \in \|lh(r)\|_{U^*} \text{ and } r \in Rule\_set\}, \tag{3}$$

where $x \in U^*$ and $lh(r)$ denotes the formula on the left hand side of the rule r, and

$$\nu_U(X, Z) = \begin{cases} h\left(\frac{card(\{Y \in X : Y \cap U \subseteq Z\})}{card(\{Y \in X : Y \cap U \subseteq Z\}) + card(\{Y \in X : Y \cap U \subseteq U^* \setminus Z\})}\right) & \text{if } X \neq \emptyset \\ 0 & \text{if } X = \emptyset, \end{cases} \tag{4}$$

where $X \subseteq P(U^*)$ and $Z \subseteq U^*$.

The defined uncertainty and rough inclusion functions can now be used to define the lower approximation $LOW(AS^*, Z)$, the upper approximation $UPP(AS^*, Z)$, and the boundary region $BN(AS^*, Z)$ of $Z \subseteq P(U^*)$ by:

$$LOW(AS^*, Z) = \{x \in U^* : \nu_U(I(x), Z) = 1\}, \tag{5}$$

and

$$UPP(AS^*, Z) = \{x \in U^* : \nu_U(I(x), Z) > 0\}, \tag{6}$$

$$BN(AS^*, Z) = UPP(AS^*, Z) \setminus LOW(AS^*, Z). \tag{7}$$

In the example, we classify objects from U^* to the lower approximation of Z if majority of rules matching this object are voting for Z and to the upper approximation of Z if at least half of the rules matching x are voting for Z. Certainly, one can follow many other voting schemes developed in machine learning or by introducing less crisp conditions in the boundary region definition. The defined approximations can be treated as estimations of the exact approximations of subsets of U^* because they are induced on the basis of samples of such sets restricted to U only. One can use the standard quality measures developed in machine learning to calculate the quality of such approximations assuming that after estimation of approximations full information about membership for elements of the approximated subsets of U^* is uncovered analogously to the testing sets in machine learning.

Let $C_1^* = \{x \in U^* : d(x) = 1\} = \{x_1, x_2, x_4, x_5, x_{10}, x_{13}, x_{14}\}$. We obtain the set $U^* \setminus C_1^* = C_0^* = \{x_3, x_6, x_7, x_8, x_9, x_{11}, x_{12}\}$. The uncertainty function and rough inclusion are presented in Table 4.

Table 4. Uncertainty function and rough inclusion over the set of objects U^*

	$I(\cdot)$	$\nu_U(I(\cdot), C_1^*)$
x_1	$\{\{x_1, x_{14}\}, \{x_1, x_5\}\}$	$h(2/2) = 1$
x_2	$\{\{x_2, x_4, x_{10}\}, \{x_2, x_4, x_{11}, x_{12}\}\}$	$h(2/2) = 1$
x_3	$\{\{x_3, x_7\}, \{x_3, x_9\}\}$	$h(0/2) = 0$
x_4	$\{\{x_2, x_4, x_{10}\}, \{x_2, x_4, x_{11}, x_{12}\}\}$	$h(2/2) = 1$
x_5	$\{\{x_5, x_{13}\}, \{x_1, x_5\}\}$	$h(2/2) = 1$
x_6	$\{\{x_6, x_9\}, \{x_6\}\}$	$h(0/2) = 0$
x_7	$\{\{x_3, x_7\}, \{x_7\}\}$	$h(0/2) = 0$
x_8	$\{\{x_8, x_{11}, x_{12}\}, \{x_8, x_{10}\}\}$	$h(0/2) = 0$
x_9	$\{\{x_6, x_9\}, \{x_3, x_9\}\}$	$h(0/2) = 0$
x_{10}	$\{\{x_2, x_4, x_{10}\}, \{x_8, x_{10}\}\}$	$h(1/2) = 1/2$
x_{11}	$\{\{x_8, x_{11}, x_{12}\}, \{x_2, x_4, x_{11}, x_{12}\}\}$	$h(1/2) = 1/2$
x_{12}	$\{\{x_8, x_{11}, x_{12}\}, \{x_2, x_4, x_{11}, x_{12}\}\}$	$h(1/2) = 1/2$
x_{13}	$\{\{x_5, x_{13}\}\}$	$h(1/1) = 1$
x_{14}	$\{\{x_1, x_{14}\}\}$	$h(1/1) = 1$

Thus, in our example from Table 4 we obtain

$$LOW(AS^*, C_1^*) = \{x \in U^* : \nu_U(I(x), C_1^*) = 1\} = \{x_1, x_2, x_4, x_5, x_{13}, x_{14}\}, \tag{8}$$

$$UPP(AS^*, C_1^*) = \{x \in U^* : \nu_U(I(x), C_1^*) > 0\} =$$

$$\{x_1, x_2, x_4, x_5, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}\}, \tag{9}$$

$$BN(AS^*, C_1^*) = UPP(AS^*, C_1^*) \setminus LOW(AS^*, C_1^*) = \{x_{10}, x_{11}, x_{12}\}. \tag{10}$$

5.2 Approximations and Nearest Neighbors Classifiers

In this section, we present a method for construction of rough based classifiers based on the k-nearest neighbors idea. The k-nearest neighbors algorithm (k-NN, where k is a positive integer) is a method for classifying objects based on closest training examples in the attribute space. An object is classified by a majority vote of its neighbors, with the object being assigned to the decision class most common amongst its k nearest neighbors. If $k = 1$, then the object is simply assigned to the decision class of its nearest neighbor.

Let $DT = (U, A \cup \{d\})$ be a decision table and let $DT^* = (U^*, A^* \cup \{d^*\})$ be an extension of DT. We define $NN_k : U^* \to P(INF(A))$ by

$NN_k(x) =$ a set of k elements of $INF(A)$ with minimal distances to $Inf_A(x)$

The Hamming distance $\delta_A^H(u, v)$ between two strings $u, v \in \prod_{a \in A} V_a$ of length $card(A)$ is the number of positions at which the corresponding symbols are different. In our example we use a distance $\delta_A : \prod_{a \in A} V_a \times \prod_{a \in A} V_a \to [0, 1]$ defined by $\delta_A(u, v) = \frac{\delta_A^H(u,v)}{card(A)}$ (the Hamming distance value divided by the total number of condition attributes).

The description of x_1 is $Inf_A(x_1) = (1, 1, 0) \in INF(A)$[4] (see Table 2) and the description of x_{14} is $Inf_A(x_{14}) = (1, 1, 2) \in INF(A)$ (see Table 3). Because each object is described by 3 condition attributes, we say that the Hamming distance between $Inf_A(x_1) = (1, 1, 0)$ and $Inf_A(x_{14}) = (1, 1, 2)$ is 1 and the Hamming distance divided by the total number of attributes $\delta_A((1, 1, 0), (1, 1, 2)) = \frac{1}{3}$. We define

$$I_{NN_k}(x) = \{\| \bigwedge Inf_A(y) \|_{U^*} : y \in U^* \text{ and } Inf_A(y) \in NN_k(x)\}, \tag{11}$$

$$\nu_{NN_k}(X, Y) = \frac{card(\{\bigcup(Z \cap U) : Z \in X \& Z \cap U \subseteq Y \cap U\})}{card(U)}, \tag{12}$$

Let $J^\varepsilon : U^* \to P(\{d(x) : x \in U\})$ for $0 < \varepsilon << 1$ be defined by

$$J^\varepsilon(x) = \{i : \neg\exists j \neq i(\nu_{NN_k}(I_{NN_k}(x), C_j^*) > \nu_{NN_k}(I_{NN_k}(x), C_i^*) + \varepsilon)\}, \tag{13}$$

and

$$\nu_{NN_k}^\varepsilon(I_{NN_k}(x), C_i^*) = \begin{cases} 1 \text{ if } & J^\varepsilon(x) = \{i\} \\ \frac{1}{2} \text{ if } i \in J^\varepsilon(x) & \& \ card(J^\varepsilon(x)) > 1 \\ 0 \text{ if } & \{i\} \not\subseteq J^\varepsilon(x) \end{cases} \tag{14}$$

[4] We write $Inf_A(x_1) = (1, 1, 0) \in INF(A)$ instead of $Inf_A(x_1) = \{(a, 1), (b, 1), (c, 0)\}$.

[5] $\| \bigwedge Inf_A(y) \|_{U^*}$ denotes the set of all objects from U^* satisfying the conjuntion of all descriptors $a = a(y)$ for $a \in A$.

The defined uncertainty I_{NN_k} and rough inclusion $\nu^{\varepsilon}_{NN_k}$ functions can now be used to define the lower approximation $LOW(AS^*, C_i^*)$, the upper approximation $UPP(AS^*, C_i^*)$, and the boundary region $BN(AS^*, C_i^*)$ of $C_i^* \subseteq U^*$ by:

$$LOW(AS^*, C_i^*) = \{x \in U^* : \nu^{\varepsilon}_{NN_k}(I_{NN_k}(x), C_i^*) = 1\}, \tag{15}$$

$$UPP(AS^*, C_i^*) = \{x \in U^* : \nu^{\varepsilon}_{NN_k}(I_{NN_k}(x), C_i^*) > 0\}, \tag{16}$$

$$BN(AS^*, C_i^*) = UPP(AS^*, C_i^*) \setminus LOW(AS^*, C_i^*). \tag{17}$$

Let $k = 2$ and $\varepsilon = 0.1$, in our example we obtain the results presented in Table 5. The neighbors are taken from the set U of objects for which the correct classification is known. In the classification phase, a new object is classified by assigning the decision class which is most frequent among the 2 training objects nearest to that new object. In the case of more than two nearest objects we choose randomly two.

Table 5. Uncertainty function I_{NN_2} and rough inclusion $\nu^{0.1}_{NN_2}$ over the set of objects $U^* \setminus U = \{x_{10}, \ldots, x_{14}\}$

	$NN_2(\cdot)$	$I_{NN_2}(\cdot)$	$\nu_{NN_2}(I_{NN_2}(\cdot), C_1^*)$
x_{10}	$\{(0,2,0),(1,2,1)\}$	$\{\{x_2,x_4\},\{x_8\},\{x_{10}\}\}$	2/9
x_{11}	$\{(1,1,0),(0,2,0)\}$	$\{\{x_1\},\{x_2,x_4\},\{x_{11},x_{12}\}\}$	3/9
x_{12}	$\{(1,1,0),(0,2,0)\}$	$\{\{x_1\},\{x_2,x_4\},\{x_{11},x_{12}\}\}$	3/9
x_{13}	$\{(0,1,0),(1,1,0)\}$	$\{\{x_5\},\{x_1\}\}$	2/9
x_{14}	$\{(1,1,0),(1,0,2)\}$	$\{\{x_1\},\{x_7\}\}$	1/9
	$\nu_{NN_2}(I_{NN_2}(\cdot), C_0^*)$	$J^{0.1}(\cdot)$	$\nu^{0.1}_{NN_2}(I_{NN_2}(\cdot), C_1^*)$
x_{10}	1/9	$\{1\}$	1
x_{11}	0	$\{1\}$	1
x_{12}	0	$\{1\}$	1
x_{13}	0	$\{1\}$	1
x_{14}	1/9	$\{0,1\}$	1/2

Thus, in our example from Table 5 we obtain

$$LOW(AS^*, C_1^*) = \{x \in U^* : \nu^{0.1}_{NN_2}(I_{NN_2}(x), C_1^*) = 1\} = \{x_1, x_2, x_4, x_5, x_{10}, x_{11}, x_{12}, x_{13}\}, \tag{18}$$

$$UPP(AS^*, C_1^*) = \{x \in U^* : \nu^{0.1}_{NN_2}(I_{NN_2}(x), C_1^*) > 0\} = \{x_1, x_2, x_4, x_5, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}\}, \tag{19}$$

$$BN(AS^*, C_1^*) = UPP(AS^*, C_1^*) \setminus LOW(AS^*, C_1^*) = \{x_{14}\}. \tag{20}$$

6 Conclusions

We discussed a generalization of approximation spaces based on granular formulas and neighborhoods. Efficient searching strategies for relevant approximation

spaces are crucial for application (e.g., in searching for approximation of complex concepts).

References

1. Bazan, J.G., Skowron, A., Świniarski, R.W.: Rough sets and vague concept approximation: From sample approximation to adaptive learning. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, pp. 39–62. Springer, Heidelberg (2006)
2. Bianucci, D., Cattaneo, G.: Information Entropy and Granulation Co–Entropy of Partitions and Coverings: A Summary. In: Peters, J.F., Skowron, A., Wolski, M., Chakraborty, M.K., Wu, W.-Z. (eds.) Transactions on Rough Sets X. LNCS, vol. 5656, pp. 15–66. Springer, Heidelberg (2009)
3. Jankowski, A., Skowron, A.: Logic for artificial intelligence: The Rasiowa-Pawlak school perspective. In: Ehrenfeucht, A., Marek, V., Srebrny, M. (eds.) Andrzej Mostowski and Foundational Studies, pp. 106–143. IOS Press, Amsterdam (2008)
4. Łukasiewicz, J.: Die logischen Grundlagen der Wahrscheinlichkeitsrechnung, 1913. In: Borkowski, L. (ed.) Jan Łukasiewicz - Selected Works, pp. 16–63. North Holland Publishing Company/Polish Scientific Publishers, Amsterdam/Warsaw (1970)
5. Malyszko, D., Stepaniuk, J.: Adaptive Multilevel Rough Entropy Evolutionary Thresholding. Information Sciences 180(7), 1138–1158 (2010)
6. Pal, S.K., Shankar, B.U., Mitra, P.: Granular computing, rough entropy and object extraction. Pattern Recognition Letters 26(16), 2509–2517 (2005)
7. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007)
8. Pedrycz, W., Skowron, A., Kreinovich, V. (eds.): Handbook of Granular Computing. John Wiley & Sons, New York (2008)
9. Peters, J., Skowron, A., Stepaniuk, J.: Nearness of objects: extension of the approximation space model. Fundamenta Informaticae 79(3-4), 497–512 (2007)
10. Polkowski, L., Skowron, A.: Rough mereology: A new paradigm for approximate reasoning. Int. J. Approximate Reasoning 51, 333–365 (1996)
11. Skowron, A., Stepaniuk, J.: Generalized approximation spaces. In: Lin, T.Y., Wildberger, A.M. (eds.) The Third Int. Workshop on Rough Sets and Soft Computing Proceedings (RSSC 1994), November 10-12, pp. 156–163. San Jose State University, San Jose (1994)
12. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
13. Skowron, A.: Toward intelligent systems: Calculi of information granules. In: Terano, T., Nishida, T., Namatame, A., Tsumoto, S., Ohsawa, Y., Washio, T. (eds.) JSAI-WS 2001. LNCS (LNAI), vol. 2253, pp. 251–260. Springer, Heidelberg (2001)
14. Skowron, A., Stepaniuk, J., Peters, J., Swiniarski, R.: Calculi of approximation spaces. Fundamenta Informaticae 72(1-3), 363–378 (2006)
15. Słowiński, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Trans. Knowledge and Data Engineering 12, 331–336 (2000)
16. Stepaniuk, J.: Rough–Granular Computing in Knowledge Discovery and Data Mining. Springer, Heidelberg (2008)
17. Zhu, W.: Relationship between generalized rough sets based on binary relation and covering. Information Sciences 179(3), 210–225 (2009)
18. Ziarko, W.: Variable precision rough set model. J. Computer and System Sciences 46, 39–59 (1993)

A Note on a Formal Approach to Rough Operators

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. The paper is devoted to the formalization of two elementary but important problems within rough set theory. We mean searching for the minimal requirements of the well-known rough operators – the lower and the upper approximations in an abstract approximation space to retain their natural properties. We also discuss pros and cons of the development of the computer-checked repository for rough set theory based on the comparison of certain rough approximation operators proposed by Anna Gomolińska.

1 Introduction

Rough set theory stemmed as an emerging trend reflecting the need for getting knowledge when the information is imprecise, incomplete or just too complex to obtain in the real time valuable answer for agent's query.

Although the situation of the missing data is clear and present in mathematical practice, it is not so easy to understand what the rough reasoning is when it comes to the process of theorem proving. Still, the theory itself has a strong mathematical flavour and hence is a field where available math-assistants can be successfully used to obtain some new results and to verify a bunch of solutions of older ones. However, if we tend to model a larger collection of mathematical papers, the challenge gets even bigger. Multiplicity of notations, different levels of obviousness – all that becomes a real problem, especially when we look at the theory from a broader perspective, merging views of different authors.

The formalization is a term with a broad meaning which denoted rewriting the text in a specific manner, usually in a rigorous (i.e. strictly controlled by certain rules), although sometimes cryptic language. Obviously the notion itself is rather old, originated definitely from pre-computer era, and in the early years formalization was to ensure the correctness of the approach. Together with the evolution of the tools, the new paradigm was established: computers can potentially serve as a kind of the oracle to check if the text is correct. But as of now we do not know about any (other than ours) translation into leading proof-assistants (see [15] for comprehensive list). Hence adjective "formal" occurs rather in the

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 307–316, 2010.

context of rough sets mostly in the sense of formal concept analysis, but of course some papers of this kind are available, e.g. [1]. Computer-driven encoding draws the attention of researchers of both mathematics and computer science, and if the complexity of the tools will be too high, only software engineers will be attracted and all the usefulness for an ordinary mathematician will be lost.

Our paper is organized as follows: in the next section we touch the issue of minimal assumptions under which elementary properties can be proved, and explain some basic elements of type theory within Mizar language. Third section contains the solution of the problem of approximation spaces based on relations, treated formally, while the fourth draws the perspective of the automatic improvement of proofs. Section 5 shows how various approaches can be successfully merged, and in the remaining sections we describe how the repository itself can benefit just from our development and draw some final remarks.

2 The Quest for Minimal Requirements

The Mizar system (designed by Andrzej Trybulec in 1973) consists of a language for writing formalized mathematical definitions and proofs, the software for checking correctness of proofs written in this language, and the database. The Mizar Mathematical Library (MML for short) is considered one of the largest repositories of computer verified mathematical knowledge in the world. The basic item in the MML, called Mizar article, reflects roughly a structure of an ordinary paper, being considered at two main layers – the declarative one, where definitions and theorems are stated, and the other one – proofs. Naturally, although the latter is the larger part, the earlier needs some additional care – the submission will be accepted for inclusion into the MML if the approach is correct and the topic isn't already present there.

Some of the problems formalized can be solved by certain algorithm, others can be just calculated – for computer algebra systems (CAS) it is not that hard. Although basically being not CAS, the Mizar system has already some automatic mechanisms implemented – e.g. arithmetic on complex numbers etc. As for now, more automation for an ordinary user is not easily accessible – via `requirements` library directive one can obtain nice results, but is has to be implemented in the sources of the systems (although direct results can be browsed in the file containing the formalization).

The beginnings of the project of encoding rough sets into MML were described in [5] and the results can be browsed from the homepage of the Mizar system [10] under the identifier `ROUGHS_1`. The classical rough set theory was based on equivalence relations, but the extensive research was done on arbitrary binary relation-based rough sets. This work is twofold: on the one hand starting from relations' properties such as reflexivity, symmetry or transitivity which led to studies on properties of approximation operators; on the other hand classical properties of these operators are identified (normality, extension, monotonicity or idempotency), to obtain specific attributes of binary relations.

3 Rough Tolerance and Approximation Operators

First theorem provers were based on a pure predicate calculus [14], nowadays type information is also important and it is also the case of typed language of the Mizar system. There are three main constructors of types in Mizar:

- modes – their existence has to be proven, here a good example is the most general type `set`. Other examples of such types are `Rough_Set`; among structural modes – e.g. `Approximation_Space`;
- functors – should be proven both the existence and its uniqueness; used to obtain e.g. the upper and the lower approximation operators;
- attributes – although no correctness conditions are needed, to use this, one should register the so-called existential registration of the cluster (which is just a collection of attributes). Here we can give `rough` or `exact` (crisp) as examples.

Some statistics of the usage of various constructors and symbols is contained in Table 1. As we can conclude from it, functors and attributes are defined most often in the Mizar library (also in our development we can find some); structures play a role of the framework for the theory, and in fact to formalize RST as described in this paper we need only one.

Table 1. Occurrences of symbols and constructors in the MML

Type	Symbols	%	Constructors	%
attributes	1388	20.3	2428	19.0
functors	3725	54.5	7717	60.6
modes	759	11.1	1103	8.6
predicates	639	9.3	1090	8.5
structures	122	1.8	123	1.0
Total	6831	100.0	12740	100.0

3.1 Type Refinement

Hierarchy of types can be realized threefold:

1. by defining an object with its type, based on the "widening" relation with `set` being the widest:
   `Function of X,Y` ⟶ `PartFunc of X,Y` ⟶ `Relation of X,Y` ⟶ `Subset of [:X,Y:]` ⟶ `Element of bool [:X,Y:]` ⟶ `set`
2. by adjectives:
   `reflexive transitive antisymmetric RelStr` (poset) ⟶ ⟶ `reflexive transitive RelStr` (preorder) ⟶ `reflexive RelStr` ⟶ `RelStr` (relational structure)
   Adjectives are processed to enable automatic deriving of type information (so called "registrations of clusters").

3. by polymorphic structure type expansion – this will be described in Sect. 5.

Pioneering works in rough set theory (RST for short) were stated in unified framework of equivalence relations and partitions. Later on, some of the assumptions were dropped, and as of now many theorems of the theory are formulated in the form of "if the operator $A1$ satisfies the property $P1$, then the corresponding approximation space satisfies $P2$."

In Mizar pseudo-code this can be expressed as follows:

```
definition let A1 be with_property_P1 <variable_type>;
  cluster Space (A1) -> with_property_P2;
  coherence;
end;
```

This gives the advantage of automatic adding to the object `Space (A1)` underlying properties if the type of an argument is sufficiently narrow. We formulated basic theorems of the theory under the assumption of binary relations to be tolerances, but it soon appeared that it was too restrictive. However the adjective below still plays a crucial role in our work.

```
definition
  let P be RelStr;
  attr P is with_tolerance means
:: ROUGHS_1:def 3
  the InternalRel of P is Tolerance of the carrier of P;
end;
```

3.2 Zhu's vs. Mizar Notation

When encoding Zhu's paper [17] into Mizar, we noticed that too complex notation decreased the readability of the text. It is useful to have explicitly stated arguments without browsing virtually any occurrence of the variable, but we can make them locally fixed, similarly as we reserve variables in programming languages. Especially in examples Zhu [17] writes $L(R)(X)$ to underline that the lower approximation is taken with respect to an arbitrary binary relation R. Our straightforward approach fails when we come to considering two different indiscernibility relations on the same space – then we have to apply merging operator on these spaces. Luckily we know that two different binary relations on the universe generate different approximation operators. To show how close is the Mizar language to the mathematical jargon, we give here well-known definition of the lower approximation $X_\star$ of a rough set in the approximation space.

```
definition let A be non empty RelStr;
           let X be Subset of A;
  func LAp X -> Subset of A equals
:: ROUGHS_1:def 4
  { x where x is Element of A : Class (the InternalRel of A, x) c= X };
end;
```

Observe that `Class` is just another name for the image of a relation (as a result of a revision, originally the class of abstraction of an equivalence relation) and we don't need any additional assumptions bounding relational structure A. It allows us to write in the arbitrary but fixed space A the set `LAp X` without mentioning R as an indiscernibility relation.

3.3 The Proofs

Zhu deals with the basic types of relations, such as serial, mediate (dense) and alliance ones. Our aim was to formulate and prove all the facts from [17].

As the example of the machine translation let us quote here the property (9LH) – called by Zhu *appropriateness*. The proof of Proposition 2 (9LH) can be written in Mizar formalism as follows:

```
theorem Th9LH:
  for R being non empty serial RelStr, X being Subset of R holds
    LAp X c= UAp X
  proof
    let R be non empty serial RelStr, X be Subset of R;
    let y be set;
    assume y in LAp X; then
    y in { x where x is Element of R :
      Class (the InternalRel of R, x) c= X } by ROUGHS_1:def 4; then
    consider z being Element of R such that
A1: z = y & Class (the InternalRel of R, z) c= X;
    Class (the InternalRel of R, z) meets X by XBOOLE_1:69,A1; then
    z in {x where x is Element of R :
      Class (the InternalRel of R, x) meets X};
    hence thesis by A1,ROUGHS_1:def 5;
  end;
```

The keywords as `let`, `assume` and `consider` change the thesis and form the skeleton of the proof; the latter introduces a new local constant, the remaining are self-explanatory. Serial relations were not available in the MML, we defined them for arbitrary relations, and then for relational structures.

```
definition let R be non empty RelStr;
  redefine attr R is serial means
    for x being Element of R holds ex y being Element of R st x <= y;
  compatibility;
end;
```

As it soon appeared, it is useful to have not only concrete application of the approximation (i.e. a language function), but also approximations as mathematical functions.

```
definition let R be non empty RelStr;
  func LAp R -> Function of bool the carrier of R, bool the carrier of R
    means
```

```
  for X being Subset of R holds it.X = LAp X;
end;
```

The dot symbol "." is an application of a function; `bool A` denotes powerset of the set A. Taking into account a definition of co-normality as $L_\star(U) = U$, we can write (and prove, but we omit the proof here for obvious reasons):

```
definition let R be serial non empty RelStr;
  cluster LAp R -> co-normal;
  coherence;
end;
```

The advantage of the latter registration is that after this appropriate adjective is added to the functor `LAp R` and this worked rather smoothly; more painful was the construction of an abstract space based on the properties of a function.

```
theorem
  for A being non empty set,
      L being Function of bool A, bool A st
  L.A = A & L.{} = {} &
  for X, Y being Subset of A holds L.(X /\ Y) = L.X /\ L.Y holds
  ex R being non empty serial RelStr st the carrier of R = A & L = LAp R;
```

4 The Formalization Issues

The Mizar system is freely available for download in precompiled form, together with the bunch of programs. We enumerate here some of them discovering and suggesting possible improvements on the source:

- `relprem` – detects unnecessary premises (both after `by` and linking previous sentence after `then`);
- `chklab` – checks which labels are not used and may be deleted;
- `inacc` – marks inaccessible parts of the text;
- `trivdemo` – nesting of the proof can be removed – the proof can be straightforward.

In such a way, unused assumptions are marked and are easy to remove, as well as unnecessary (unused and not exportable to the database) blocks of the text. This helps to keep automatically the level of generality.

Also the communication back and forth between various formal systems is noteworthy [14] (although this work is in an early stage), including translation into the natural language. The latter has some didactic value, enabling people who are not acquainted with the syntax understanding of the formalized facts. Because the construction of the environment in which the researcher can prove something non-trivial is not that elementary, and based on the opinions of the people writing to the Mizar Forum mailing list and asking the Mizar User Service, it is the most time-consuming part of the work, we used especially developed environments for students' training.

The so-called MML Query is up and running to enhance formalization work by better searching, not only classically, via textual `grep`, but giving the set of needed constructors and asking which theorems use them all (e.g. querying for theorems which use `LAp` and set-theoretic inclusion).

Usual activities when using computer proof-assistants are defining notions (`definition` block) and formulating and proving theorems (with `theorem` keyword). We can also build certain models with properties verified by computer which is pretty close to the contemporary use of the rough set theory – but although the construction of reasonably big models is possible in Mizar, such investments can be one-shot – the reusability of such object within the Mizar Mathematical Library can be small. Note however that in the MML we already have e.g. electric circuit modelling (CIRCUIT or GATE series) or Random Access Turing Machines model (SCM, AMI series), so having a collection of computer-verified examples or software enabling easy construction of such objects would be very useful.

5 Rough Approximation Mappings Revealed

In this section we characterize formally various operators of rough approximation as compared in [3]. The generalized approximation space is taken into account as a triple $\mathcal{A} = \langle U, I, \kappa \rangle$, where U is a non-empty universe, $I : U \mapsto \wp(U)$ is an uncertainty mapping, and $\kappa : \wp(U) \times \wp(U) \mapsto [0, 1]$ is a rough inclusion function. Gomolińska lists the "rationality" postulates (a1)–(a6) which approximation mappings should possess. Then she enumerates ten mappings $(f_i,\ i = 1, \ldots, 9)$ and proves their properties.

In fact, under relatively weak condition (1) from [3] $(\forall_{u \in U}\ u \in I(u))$, the function I generates a covering of the universe U, and it soon turns out that using topological notions we can go further. The role of the uncertainty mapping I may be played by a binary relation on U, but any mapping I satisfying (1) generates a reflexive relation $\rho \subseteq U \times U$ such that for every $u, w \in U$

$$(w, u) \in \rho \text{ iff } w \in I(u).$$

Then we obtain ρ as indiscernibility relation. Conversely, any reflexive relation $\rho \subseteq U \times U$ generates an uncertainty mapping $I : U \mapsto \wp(U)$, satisfying (1), where

$$I(u) = \rho^{\leftarrow}(\{u\}) \qquad \text{and} \qquad \tau(u) = \rho^{\rightarrow}(\{u\}).$$

Remembering in the MML we defined the tolerance approximation space as the pair $\langle U, R \rangle$, we had in mind two solutions available not to duplicate all the formal apparatus from scratch:

1. to extend already existing tolerance space structure via uncertainty mappings I and τ and to obtain explicit operators as mathematical functions;
2. to use existing type machinery and expressive power of the Mizar language and to define both maps as language functors, retaining original structure of a space.

The first part of the alternative is realized as the ordinary concatenation of structures as tuples, we obtain $\langle U, R, I, \tau \rangle$, formally:

```
definition
  struct (RelStr) InfoStruct (#
         carrier -> set,
     InternalRel -> Relation of the carrier,
  UncertaintyMap -> Function of the carrier, bool the carrier,
          TauMap -> Function of [:bool the carrier, bool the carrier:],
                    [. 0,1 .] #);
end;
```

As usual, to obtain semantics of this object, we can restrict the universe of discourse by introducing attributes for natural definitions of τ and I, where the latter is as follows (we changed an actual definition a little, not to use additional symbols):

```
definition let R be non empty InfoStruct;
  attr R is with_uncertainty means
    for u, w being Element of R holds
    [u,w] in the InternalRel of R iff u in (the UncertaintyMap of R).w;
end;
```

Hence we can use the type `with_uncertainty with_tolerance non empty InfoStruct` to have a tolerance approximation space with I uncertainty mapping as built-in object and this is what we eventually decided for. Otherwise, I mapping would be defined as

```
definition let R be non empty reflexive RelStr;
  func UncMap -> Function of R, bool the carrier of R means
  for u being Element of R holds
    it.u = CoIm (the InternalRel of R, u);
  correctness;
end;
```

where `CoIm` stands for counterimage of relation applied to a singleton of an element u. Then proving properties of maps f_i's defined by Gomolińska either follow her suggestions or filling the gaps in the proof by ourselves was just a typical exercise (with difficulty level varied), done partly by the students.

6 Side-Effects for the MML Repository

The important limitation which we should have in mind when developing basics of RST is that the Mizar system has fixed both logic and set theory. While classical first-order logic with some constructions of the second order (like the scheme of induction as a most used example) is a part of the implementation, the axioms of the Tarski-Grothendieck set theory, can be potentially modified by ordinary user. Essentially though a few other logics are formalized, and hence can be chosen as a language describing rough sets.

6.1 Actual Gains

The current policy of the Mizar Mathematical Library is that all duplications are to be removed. For example, a few articles sent to the repository for review obtained a negative grade, hence they were rejected due to the fact that the topic was already formalized.[1]

The level of generalization. Although the submission to the MML (a Mizar article) is copyrighted and frozen – and in such form it is translated into natural language, the repository as a whole is subject to modifications, called revisions. In such a manner, instead of a collection of papers gradually extending the topic (in case of RST – defining approximation spaces based on equivalence relations, tolerances, etc.), we can have one library item under various assumptions.

Copyright issues. In case of unchanged items it is not the problem, but if someone else does the generalization, the issues of authorship of the material are questionable. As a rule, first authorship rule is claimed. In the reality of Wikipedia and the world evolving fast, we can track changes to obtain parameters of the revision.

The searching. The smaller database is, the faster we can find the appropriate object – basically we have no duplications of notions for approximations, tolerances etc.

Coherence. Formal writing forces the author to use existing notions without any variants (although redefinitions after proving equivalence with the original are permitted); otherwise software will complain.

Didactic issues. Via XML exchange format (with possible plain HTML clickable output) any occurrence of the notion is linked directly to its definition – it proved its usefulness during exercises taken on our students.

Bright example of the mentioned case is the solution of Robbins' problem about the alternative axiomatization of Boolean algebras, rather cryptic in its original form discovered by EQP/Otter software, but after machine suggestions proposed by Dahn [2] and his δ notation and making lemmas of a more general interest it was more understandable and less painful to the eye.

7 Final Remarks

Our research can be understood as a step towards a kind of computer-supported reverse mathematics (asking which axioms are required to prove theorems), following [7], [13], [8], although we stop earlier than the founder of the program, Harvey Friedman proposed – not somewhere close to the second-order arithmetic, but on elementary properties of relations in pure set theory. So it turns out in our case that we are looking for minimal requirements needed to obtain the desired properties of the object. Using machine proof-assistants extends possibilities of the reasoning, however much more progress should be made to attract

[1] This policy was widely criticized, also by the reviewers of this paper.

potential collaborators. As a rule, the Mizar checker is independent of the set theory, however some constructions are significant restrictions. E.g. one cannot prove freely within the Mizar Mathematical Library that the Axiom of Choice is equivalent to the Tichonov Theorem. As the topology induced by a reflexive and transitive binary relation is an Alexandrov topology, and the topology and continuous lattice theory are areas very well developed in the MML, we hope that in the nearest future the exhaustive study of the conditions of indiscernibility relations in terms of coverings [12] to retain usual properties of the approximation operators will be completed.

References

1. Bryniarski, E.: Formal conception of rough sets. Fundamenta Informaticae 27(2-3), 109–136 (1996)
2. Dahn, B.I.: Robbins algebras are Boolean: A revision of McCune's computer-generated solution of the Robbins problem. J. of Algebra 208(2), 526–532 (1998)
3. Gomolińska, A.: A comparative study of some generalized rough approximations. Fundamenta Informaticae 51(1-2), 103–119 (2002)
4. Grabowski, A.: Basic properties of rough sets and rough membership function. Formalized Mathematics 12(1), 21–28 (2004)
5. Grabowski, A., Jastrzębska, M.: Rough set theory from a math-assistant perspective. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 152–161. Springer, Heidelberg (2007)
6. Grabowski, A., Schwarzweller, C.: 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)
7. Järvinen, J.: Approximations and rough sets based on tolerances. In: Ziarko, W., Yao, Y.Y. (eds.) RSCTC 2000. LNCS (LNAI), vol. 2005, pp. 182–189. Springer, Heidelberg (2001)
8. Jiang, B., Qin, K., Pei, Z.: On transitive uncertainty mappings. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 42–49. Springer, Heidelberg (2007)
9. Liang, X., Li, D.: On rough subgroup of a group. To appear in Formalized Mathematics (2009); MML Id: GROUP_11
10. Mizar Home Page, http://mizar.org/
11. Padlewska, B.: Families of sets. Formalized Mathematics 1(1), 147–152 (1990)
12. Samanta, P., Chakraborty, M.: Covering based approaches to rough sets and implication lattices. In: Sakai, H., Chakraborty, M.K., Hassanien, A.E., Ślęzak, D., Zhu, W. (eds.) RSFDGrC 2009. LNCS, vol. 5908, pp. 127–134. Springer, Heidelberg (2009)
13. Qin, K., Yang, J., Pei, Z.: Generalized rough sets based on reflexive and transitive relations. Information Sciences 178, 4138–4141 (2008)
14. Urban, J.: Translating Mizar for first order theorem provers. In: Asperti, A., Buchberger, B., Davenport, J.H. (eds.) MKM 2003. LNCS, vol. 2594, pp. 203–215. Springer, Heidelberg (2003)
15. Wiedijk, F. (ed.): The Seventeen Provers of the World. LNCS (LNAI), vol. 3600. Springer, Heidelberg (2006)
16. Zhang, H., Ouyang, Y., Wang, Z.: Note on "Generalized rough sets based on reflexive and transitive relations". Information Sciences 179, 471–473 (2009)
17. Zhu, W.: Generalized rough sets based on relations. Information Sciences 177, 4997–5011 (2007)

Communicative Approximations as Rough Sets

Mohua Banerjee[1,⋆], Abhinav Pathak[2],
Gopal Krishna[2], and Amitabha Mukerjee[2]

[1] Dept of Mathematics and Statistics,
Indian Institute of Technology, Kanpur, India
mohua@iitk.ac.in
[2] Dept of Computer Science & Engineering,
Indian Institute of Technology, Kanpur, India
amit@cse.iitk.ac.in

Abstract. Communicative approximations, as used in language, are equivalence relations that partition a continuum, as opposed to observational approximations on the continuum. While the latter can be addressed using tolerance interval approximations on interval algebra, new constructs are necessary for considering the former, including the notion of a "rough interval", which is the indiscernibility region for an event described in language, and "rough points" for quantities and moments. We develop the set of qualitative relations for points and intervals in this "communicative approximation space", and relate them to existing relations in exact and tolerance-interval formalisms. We also discuss the nature of the resulting algebra.

1 Tolerances for Points and Intervals

When telling someone the time, saying "quarter past nine" has an implicit tolerance of about fifteen minutes, whereas the answer "9:24" would indicate a resolution of about a minute. Communication about quantities are defined on a shared conventional space, which constitutes a tesselation on the real number line. In this paper, we attempt to develop the first steps toward a theory that formulates these questions in terms of an indistinguishability relation [11], defining a tolerance approximation space common to participants in the discourse.

We take the *communicative approximation space* to be a set of cultural conventions that define a hierarchy of tesselations on a continuum. The two statements above reflect differing tolerances, defined on different discrete tesselations (granularities). The granularity adopted in a speech act reflects the measurement error or task requirement, and typically adopts the closest tesselation available in the shared communicative approximation space.

The communicative approximation space $\mathbb{C}$ then gives a discourse grid that is available at a number of scales, defined by equivalence classes (e.g. of one

⋆ The research was supported by grant NN516 368334 from the Ministry of Science and Higher Education of the Republic of Poland.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 317–326, 2010.

minute, five minutes, quarter hour, etc.). Similar questions of scale also inform communicative models for other continua such as space or measures.

The uncertainty resulting from measurement error $\pm\eta$ is defined on a tolerance approximation space $\mathbb{T}$ defined on a continuum, whereas the uncertainty reflected in communication, say ϵ, is defined on a discrete set of scales defined in the communicative approximation space $\mathbb{C}$. The ϵ-tesselation is a partition of the space via a series of ticks or *grid points*. These partitions or intervals then represent equivalence classes underlying the utterance. In honest communication, given a hierarchy of tesselations, ϵ will be chosen so as to be less precise than $\pm\eta$, in order to avoid a false impression of greater precision. Thus, we may assume that $\epsilon \geq |\eta|$. The greatest flexibility (worst case) arises when $\pm\eta = \epsilon$, and this is what we shall be assuming in the rest of this paper.

1.1 Measurement Tolerance vs. Communication Succinctness

The mapping from quantitative measures to common conceptual measures involves a step-discretization which has been the subject of considerable work in *measure theory* [5,15], *mereotopology* [2,16] and *interval analysis* [17,1]. At the same time, there is a rich tradition on information granulation in *rough set theory* [9,12]. Pawlak's premise [11] was that knowledge is based on the ability to classify objects, and by object one could mean 'anything we can think of' – real things, states, abstract concepts, processes, moments of time, etc. The original mathematical formulation of this assumption was manifested in the notion of an "approximation space": the domain of discourse, together with an equivalence relation on it.

$\mathbb{R}^+$, the set of non-negative real numbers, partitioned by half-open intervals [i,i+1), i=0,1,2,..., is an approximation space that is relevant to our work. One may remark that, in [9], Pawlak defines "internal" and "external measures" of any open interval (0,r) based on this partition, giving rise to a "measurement system". Later, in [8], this "inexactness" of measurement is further discussed, and contrasted with the theory of measurement of [14].

Our approach is related to *interval algebra* and *qualitative reasoning* [1]; however these operate with exact intervals and ignore tolerances. In this work, we develop the idea of interval tolerances [7] and map these onto communicative space discretization. We restrict ourselves to intervals, defined with two endpoints, with a single uncertainty ϵ. The next sections introduce the notion of a "rough interval" defined in terms of lower and upper approximations on the ϵ-tesselation. The end points of these rough intervals are indiscernibility regions which we call "rough points" by analogy to the continuum situation, though these are not rough sets except in a degenerate sense, since the lower approximation is empty. This extends the rough set [10] characterization for moments. Qualitative relations for rough points and intervals are defined, and compared with existing relations in the tolerance interval framework. A preliminary study is made of the relational algebraic structures that result from these constructs.

2 Rough Point

We consider a discretization of the continuum $\mathbb{R}$ by a (granularity) measure ϵ ($\in \mathbb{R}^+$). A real quantity ζ is taken as a 'reference point'. The communicative approximation space $\mathbb{C}$ is then a partition on $\mathbb{R}$ with the half-open intervals $[\zeta + k\epsilon, \zeta + (k+1)\epsilon)$, k being any integer. The points $\zeta + k\epsilon$ are called **grid points** (or 'ticks'), and the collection of grid points is called the **grid space**. Each grid interval is equivalent to an ϵ measure in $\mathbb{R}$. We notice that $\mathbb{C}$ is an approximation space that is a generalization of the one considered by Pawlak in [9] (cf. Section 1.1).

Note 1. For simplicity, we denote the k-th grid point, viz. $\zeta + k\epsilon$, as k, and the communicative approximation space $\mathbb{C}$ is taken to be the continuum $\mathbb{R}$ with this simplified representation of the discretization.

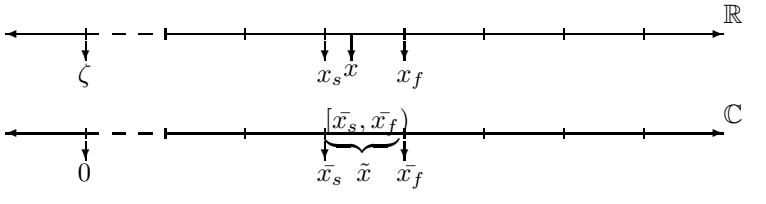
Observation 1. *The grid space is isomorphic to the set of integers, $\mathcal{Z}$.*

Example 1. A discretization of $\mathbb{R}$ with $\epsilon = 0.5$, $\zeta = 1.2$, would map a real line with grid points at 0.2, 0.7, 1.2, 1.7, 2.2, ... The communicative space then has the interval [0.2,0.7) as "-2", [0.7,1.2) as "-1", [1.2, 1.7) as "0" etc.

Now, given an exact x, one can locate the interval $[x_s, x_f)$ of $\mathbb{R}$ in which x lies (x_s, x_f are the 'start', 'end' of the interval). Thus $x_s = \zeta + \epsilon\bar{x_s}$, $x_f = \zeta + \epsilon\bar{x_f}$, where $\bar{x_s} \equiv \lfloor \frac{x-\zeta}{\epsilon} \rfloor$, and $\bar{x_f} \equiv \lfloor \frac{x-\zeta}{\epsilon} + 1 \rfloor$. $[\bar{x_s}, \bar{x_f})$ is the corresponding interval in $\mathbb{C}$. Note that $\bar{x_f} = \bar{x_s} + 1$.
In the above example, the real number $x = 0.9$ would lie in the interval $[-1, 0)$ or, equivalently, in the interval $[0.7, 1.2)$ of $\mathbb{R}$.

Definition 1. *A **rough point** is any interval $[k, k+1)$ in $\mathbb{C}$.*

We observe that $[k, k+1)$ is a representation in $\mathbb{C}$ of all such real points x in $[\bar{x_s}, \bar{x_f})$, and *we denote it as* $\tilde{x}$. In other words, $\tilde{x}$ is the denotation for the unique equivalence class in $\mathbb{C}$ of $x \in [k, k+1)$. The quotient set $\mathbb{R}/\epsilon$ is thus the collection of all rough points.



For any rough point $\tilde{x}$ in $\mathbb{C}$, $\tilde{x} + 1_\epsilon$ and $\tilde{x} - 1_\epsilon$ are defined respectively as:

$$\tilde{x} + 1_\epsilon \equiv [\bar{x_s} + 1, \bar{x_f} + 1),$$

and

$$\tilde{x} - 1_\epsilon \equiv [\bar{x_s} - 1, \bar{x_f} - 1).$$

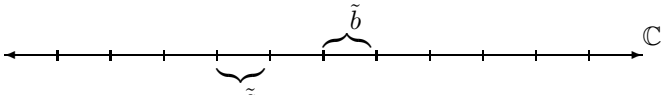
The rough points $\tilde{x} + 1_\epsilon$ and $\tilde{x} - 1_\epsilon$ are said to be **contiguous** to $\tilde{x}$. Quite similarly, $\tilde{x} + 2_\epsilon$, $\tilde{x} + 3_\epsilon$, etc. are defined.

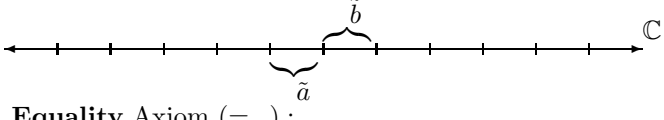
2.1 Rough Point-Rough Point Relations

Three binary relations $\succeq_\eta, \prec_\eta, \succ_\eta$ may be defined on the tolerance approximation space $\mathbb{T}$ with a tolerance measure η ($\in \mathbb{R}^+$) [7]. Let $x, y \in \mathbb{R}$.

($P1$) *Identity Axiom* ($\succeq_\eta$) :
$x \succeq_\eta y \Leftrightarrow (|x - y| < \eta)$
($P2$) *Lesser Inequality Axiom* ($\prec_\eta$) :
$x \prec_\eta y \Leftrightarrow (x \leq y - \eta)$
($P3$) *Greater Inequality Axiom* ($\succ_\eta$) :
$x \succ_\eta y \Leftrightarrow (x \geq y + \eta)$

In contrast, we observe the following five relations on the set of rough points defined in the communicative approximation space $\mathbb{C}$ with a granularity measure ϵ. Let $\tilde{a} \equiv [\ \bar{a_s}, \bar{a_f})$ and $\tilde{b} \equiv [\ \bar{b_s}, \bar{b_f})$ be two rough points.

1. **Before** Axiom ($<_\epsilon$) :
$$\tilde{a} <_\epsilon \tilde{b} \Leftrightarrow (\bar{a_f} < \bar{b_s}) \qquad (\tilde{a}\ before\ \tilde{b})$$


2. **Before Equality** Axiom ($=_{<\epsilon}$) :
$$\tilde{a} =_{<\epsilon} \tilde{b} \Leftrightarrow (\bar{a_f} = \bar{b_s}) \qquad (\tilde{a}\ equalsBefore\ \tilde{b})$$


3. **Exact Equality** Axiom ($=_{e\epsilon}$) :
$$\tilde{a} =_{e\epsilon} \tilde{b} \Leftrightarrow (\bar{a_s} = \bar{b_s}) \qquad (\tilde{a}\ equalsExact\ \tilde{b})$$

The relations **After Equality**, and **After**, are defined in a dual manner.

Observation 2. *Let $\tilde{a} = [\bar{a_s}, \bar{a_f})$ and $\tilde{b} = [\bar{b_s}, \bar{b_f})$ be two rough points.*

1. *If $\tilde{a} <_\epsilon \tilde{b}$, there is an integer $k > 1$ such that $\tilde{b} = \tilde{a} + k_\epsilon$. If $\tilde{a} =_{<\epsilon} \tilde{b}$, $\tilde{b} = \tilde{a} + 1_\epsilon$, i.e. $\tilde{a}$, $\tilde{b}$ are contiguous.*
2. *$\tilde{a} <_\epsilon \tilde{b}$ if and only if there is a rough point $\tilde{c}$ such that $\tilde{a} <_\epsilon \tilde{c}$ and $\tilde{c} <_\epsilon \tilde{b}$, or $\tilde{a} =_{<\epsilon} \tilde{c}$ and $\tilde{c} =_{<\epsilon} \tilde{b}$, or $\tilde{a} <_\epsilon \tilde{c}$ and $\tilde{c} =_{<\epsilon} \tilde{b}$, or $\tilde{a} =_{<\epsilon} \tilde{c}$ and $\tilde{c} <_\epsilon \tilde{b}$.*
3. *The equalsExact relation is an equivalence relation on the set of rough points. It is, in fact, a congruence relation with respect to the equalsBefore and equalsAfter relations: $\tilde{a} =_{e\epsilon} \tilde{b}$ and $\tilde{b} R \tilde{c}$ imply $\tilde{a} R \tilde{c}$, where R is $=_{<\epsilon}$ or $=_{>\epsilon}$.*
4. *The before and after relations are transitive.*
5. *The relations in the pairs (before, after), and (equalsBefore, equalsAfter) are converses of each other: $\tilde{a} <_\epsilon \tilde{b} \Leftrightarrow \tilde{b} >_\epsilon \tilde{a}; \tilde{a} =_{<\epsilon} \tilde{b} \Leftrightarrow \tilde{b} =_{>\epsilon} \tilde{a}$.*

Remark 1. The correspondence between the set of all pairs of rough points and the set of all Rough Point-Rough Point relations defines a function.

Note 2. We shall drop the ϵ subscript in all our notations to make them more readable, but they would be assumed to be valid in some communicative approximation space $\mathbb{C}$ with a granularity ϵ.

Mapping Point-Point Relations in $\mathbb{C}$ and $\mathbb{T}$. As stated earlier, we consider a tolerance approximation space $\mathbb{T}$ with tolerance measure $\eta = \epsilon$. The transition from point-point relations in $\mathbb{T}$ to those in $\mathbb{C}$, and vice-versa is given by the following propositions. We drop the subscript ϵ in the notation of the $\mathbb{T}$-relations.

Proposition 1. $\mathbb{T}$ to $\mathbb{C}$: *For $x, y \in \mathbb{R}$, the $\mathbb{T}$-relations defined between them through $(P1)-(P3)$ are $\prec$, $\asymp$, and $\succ$. Then the possible $\mathbb{C}$-relations between the corresponding rough points $\tilde{x}, \tilde{y}$ are given in the table on the left.*

	$\mathbb{T}$-*Relation*	$\mathbb{C}$-*Relation*
(a)	$\prec$	$<$, $=_<$
(b)	$\asymp$	$=_<$, $=_e$, $=_>$
(c)	$\succ$	$=_>$, $>$

	$\mathbb{C}$-*Relation*	$\mathbb{T}$-*Relation*
(a)	$<$	$\prec$
(b)	$=_<$	$\asymp$, $\prec$
(c)	$=_e$	$\asymp$
(d)	$=_>$	$\asymp$, $\succ$
(e)	$>$	$\succ$

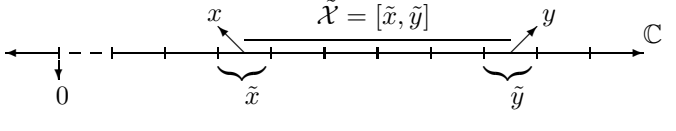
Proposition 2. $\mathbb{C}$ to $\mathbb{T}$: *Let $\tilde{x}$, $\tilde{y}$ be rough points. The possible $\mathbb{T}$-relations between any two real points $x \in \tilde{x}$, $y \in \tilde{y}$ are given in the table on the right.*

3 Rough Interval

Definition 2. *A* **rough interval** *is the union of any finite number of contiguous rough points $\tilde{x_1}, \tilde{x_2}, ..., \tilde{x_k}$.*

A rough point, in particular, is also a rough interval. Further, considering the rough interval $\tilde{x_1} \cup \tilde{x_2} \cup ... \cup \tilde{x_k}$, one observes that for any $x \in \tilde{x_1},\ y \in \tilde{x_k},\ \tilde{x_1} = \tilde{x},\ \tilde{x_2} = \tilde{x} + 1, ...,\ \tilde{x_k} = \tilde{y}$. In the terminology of rough set theory, for all such x, y, the intervals $X \equiv [x, y]$ are therefore *roughly equal*. They share the same upper approximation, which is the rough interval in question, and the same lower approximation (empty for $k = 1, 2$, and $\tilde{x_2} \cup \tilde{x_3} \cup ... \cup \tilde{x_{k-1}}$ for $(k \geq 3)$).

We denote the rough interval $\tilde{x_1} \cup \tilde{x_2} \cup ... \cup \tilde{x_k}$ as $\tilde{\mathcal{X}}$ – corresponding to any real interval $X \equiv [x, y]$ with x, y as above. Another denotation used would be $[\tilde{x}, \tilde{y}]$, to indicate the 'starting rough point' and 'end rough point' of the rough interval.



Any real interval X is thus a *rough set* in the communicative approximation space $\mathbb{C}$, and $\tilde{\mathcal{X}}$ is its *upper approximation*. A rough interval, on the other hand, is a *definable/exact* set in $\mathbb{C}$. The *lower approximation* $\underline{X}$ of X in $\mathbb{C}$ is the rough interval $[\tilde{x}+1, \tilde{y}-1]$. It would also be termed the **interior** of the rough interval $\tilde{\mathcal{X}}$. If $\tilde{x}, \tilde{y}$ are contiguous or equal, $\underline{X}$ is empty.

In the degenerate case $x = y$, i.e. when $X = \{x\}$, $\tilde{\mathcal{X}}$ is just the rough point $\tilde{x}$. Generally, if $x < y$, we can have any of the three possibilities (i) $\tilde{x} < \tilde{y}$, (ii) $\tilde{x} =_< \tilde{y}$, or (iii) $\tilde{x} =_e \tilde{y}$. As noted in Observation 2, in case (i), $\tilde{y} = \tilde{x} + k$, for some integer k and so $\tilde{\mathcal{X}} = \tilde{x} \cup \tilde{x}+1 \cup \tilde{x}+2 \cup ... \cup \tilde{x}+k$. In case (ii), $\tilde{\mathcal{X}} = \tilde{x} \cup \tilde{x}+1$. (iii) gives $\tilde{\mathcal{X}} = \tilde{x}$ again.

Remark 2. Having said this, we observe that most discussions on interval algebras in tolerance spaces assume that $|I| \gg \eta$ for any real interval I, and tolerance measure η. For realistic discourse on a rough interval $\tilde{x_1} \cup \tilde{x_2} \cup ... \cup \tilde{x_k}$, we would expect that $k \gg 1$. Minimally, for a non-empty interior, we consider intervals with at least four contiguous rough points, i.e. we assume $\tilde{x_1} + 1 < \tilde{x_k}$.

3.1 Rough Point-Rough Interval Relations

Given a point x in $\mathbb{R}$, and a real interval I, we have the following binary relations between x and I in the tolerance approximation space $\mathbb{T}$ with a tolerance η:

$(PI1)$ $x - I \Leftrightarrow x < i_1 - \eta$
$(PI2)$ $x \ b \ I \Leftrightarrow x \asymp i_1 \Leftrightarrow |i_1 - x| < \eta$
$(PI3)$ $x \ i \ I \Leftrightarrow x \in (i_1 + \eta, i_2 - \eta)$
$(PI4)$ $x \ f \ I \Leftrightarrow x \asymp i_2 \Leftrightarrow |i_2 - x| < \eta$
$(PI5)$ $x + I \Leftrightarrow x > i_2 + \eta$

In $\mathbb{C}$, a rough point $\tilde{x}$ and a rough interval $\tilde{\mathcal{I}} \equiv [\tilde{i}_1, \tilde{i}_2]$ have nine possible relations:

1. $\tilde{x}$ **before** $\tilde{\mathcal{I}}$ $(<)$: $(\tilde{x} < \tilde{\mathcal{I}}) \Leftrightarrow (\tilde{x} < \tilde{i}_1)$
2. $\tilde{x}$ **startsBefore** $\tilde{\mathcal{I}}$ $(s_<)$: $(\tilde{x} \ s_< \ \tilde{\mathcal{I}}) \Leftrightarrow (\tilde{x} =_< \tilde{i}_1)$
3. $\tilde{x}$ **startsExact** $\tilde{\mathcal{I}}$ (s_e): $(\tilde{x} \ s_e \ \tilde{\mathcal{I}}) \Leftrightarrow (\tilde{x} =_e \tilde{i}_1)$
4. $\tilde{x}$ **startsAfter** $\tilde{\mathcal{I}}$ $(s_>)$: $(\tilde{x} \ s_> \ \tilde{\mathcal{I}}) \Leftrightarrow (\tilde{x} =_> \tilde{i}_1)$
5. $\tilde{x}$ **interior** $\tilde{\mathcal{I}}$ (in): $(\tilde{x} \ in \ \tilde{\mathcal{I}}) \Leftrightarrow (\tilde{i}_1 < \tilde{x} < \tilde{i}_2)$
   It may be remarked that the interior relation exists if and only if the interior of the rough interval has at least three contiguous rough points.
6. $\tilde{x}$ **finishesBefore** $\tilde{\mathcal{I}}$ $(f_<)$: $(\tilde{x} \ f_< \ \tilde{\mathcal{I}}) \Leftrightarrow (\tilde{x} =_< \tilde{i}_2)$
7. $\tilde{x}$ **finishesExact** $\tilde{\mathcal{I}}$ (f_e): $(\tilde{x} \ f_e \ \tilde{\mathcal{I}}) \Leftrightarrow (\tilde{x} =_e \tilde{i}_2)$
8. $\tilde{x}$ **finishesAfter** $\tilde{\mathcal{I}}$ $(f_>)$: $(\tilde{x} \ f_> \ \tilde{\mathcal{I}}) \Leftrightarrow (\tilde{x} =_> \tilde{i}_2)$
9. $\tilde{x}$ **after** $\tilde{\mathcal{I}}$ $(>)$: $(\tilde{x} > \tilde{\mathcal{I}}) \Leftrightarrow (\tilde{x} > \tilde{i}_2)$

Mapping Point-Interval Relations in $\mathbb{C}$ and $\mathbb{T}$. As in Section 2.1, we assume that the tolerance approximation space $\mathbb{T}$ has the tolerance measure $\eta = \epsilon$.

Proposition 3. $\mathbb{T}$ **to** $\mathbb{C}$**:** *Consider a real point x, a real interval $I \equiv [i_1,i_2]$, and the corresponding rough point $\tilde{x}$ and rough interval $\tilde{\mathcal{I}}$. The possible $\mathbb{C}$-relations between $\tilde{x}$ and $\tilde{\mathcal{I}}$ are given as follows.*

	$\mathbb{T}$-Relation	$\mathbb{C}$-Relation
(a)	$-$ $(x < i_1 - \epsilon)$	$<$, $s_<$
(b)	b $(\|i_1 - x\| < \epsilon)$	$s_<$, s_e, $s_>$
(c)	i $(x \in (i_1 + \epsilon, i_2 - \epsilon))$	$s_>$, in, $f_<$
(d)	f $(\|i_2 - x\| < \epsilon)$	$f_<$, f_e, $f_>$
(e)	$+$ $(x > i_2 + \epsilon)$	$f_>$, $>$

Proposition 4. $\mathbb{C}$ **to** $\mathbb{T}$**:** *Let $\tilde{x}$ be a rough point and $\tilde{\mathcal{I}} \equiv [\tilde{i_1}, \tilde{i_2}]$ a rough interval corresponding to any real interval I. The $\mathbb{C}$ to $\mathbb{T}$ mappings are unique, except for $\mathbb{C}$-Relations $s_<$ and $f_>$, for which the $\mathbb{T}$-Relations are $\{-, b\}$ and $\{f, +\}$ respectively.*

4 Relations between Rough Intervals

Relations between two rough intervals $\tilde{\mathcal{A}}$ $(\equiv [\tilde{a}_1, \tilde{a}_2])$ and $\tilde{\mathcal{B}}$ $(\equiv [\tilde{b}_1, \tilde{b}_2])$ are defined by the relations that the starting rough point $\tilde{a}_1$ and the end rough point $\tilde{a}_2$ of $\tilde{\mathcal{A}}$ have with the rough interval $\tilde{\mathcal{B}}$. Any such relation shall be represented by a pair (R_1, R_2), provided $\tilde{a}_1 R_1 \tilde{\mathcal{B}}$, and $\tilde{a}_2 R_2 \tilde{\mathcal{B}}$, where R_1, R_2 denote any of the relations defined in Section 3.1.
For example: $\tilde{\mathcal{A}}$ $(<, f_<)$ $\tilde{\mathcal{B}} \Leftrightarrow (\tilde{a}_1 < \tilde{\mathcal{B}})$ and $(\tilde{a}_2$ $f_<$ $\tilde{\mathcal{B}})$.

Remark 3. Due to the non-empty interior constraint, $(\tilde{a}_1 + 1_\epsilon < \tilde{a}_2) \wedge (\tilde{b}_1 + 1_\epsilon < \tilde{b}_2)$, some Rough Interval-Rough Interval Relations are not acceptable, e.g. $(\tilde{\mathcal{A}}(s_>, <)\tilde{\mathcal{B}})$ is not acceptable as $\tilde{a}_2 < \tilde{\mathcal{B}}$ and $\tilde{a}_1 < \tilde{a}_2 \Rightarrow \tilde{a}_1 < \tilde{\mathcal{B}}$ which contradicts $\tilde{\mathcal{A}}$ $s_>$ $\tilde{\mathcal{B}}$.

So, we have the following possible relations between two rough intervals $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ – given in the Table 1.

Observation 3. *Inclusion:* Considering ordinary set inclusion, we have
1. $\tilde{\mathcal{A}} \subset \tilde{\mathcal{B}} \Leftrightarrow \tilde{\mathcal{A}}$ $(=_{e<}, =_{><}, =_{>e}, s_e, s_>, cb, f_<, f_e)$ $\tilde{\mathcal{B}}$, *and*
2. $\tilde{\mathcal{A}} \subseteq \tilde{\mathcal{B}} \Leftrightarrow \tilde{\mathcal{A}}$ $(=_{e<}, =_{ee}, =_{><}, =_{>e}, s_e, s_>, cb, f_<, f_e)$ $\tilde{\mathcal{B}}$.

One may define *containment* $(\supset/\supseteq)$ dually.

4.1 Mapping Interval-Interval Relations from Toleranced Real Model

Proposition 5. *Interval-interval relations are written concatenated from the point-interval relations: thus the notation A bf B indicates that interval A begins (b) and finishes (f) at the same points as B, i.e., the intervals A and B are equal. There are 13 relations between two real intervals in the Toleranced Real Model [7]. The Rough Set Model relations for corresponding rough intervals are given below for seven relations (the other six are inverses of cases a-f).*

Table 1. Relations between two rough intervals $\tilde{\mathcal{A}}[\tilde{a}_1, \tilde{a}_2]$ and $\tilde{\mathcal{B}}[\tilde{b}_1, \tilde{b}_2]$; 19 (including inverses) of the 33 relations are shown; the others are finishedBy{After,Exact,Before}, contains, starts{Before,Exact,After} which have inverses in finishes{After,Exact,Before}, containedBy, startedBy{Before,Exact,After} respectively. (Yes, do let us know if you can suggest some more readable names!)

$\tilde{A}$ Relation $\tilde{B}$	Definition	$\tilde{B}$ Relation $\tilde{A}$
$\tilde{A}$ before $\tilde{B}$ $(<)$	$(\tilde{A} < \tilde{B}) \Leftrightarrow (\tilde{a}_2 < \tilde{B})$	$\tilde{B}$ after $\tilde{A}$
$\tilde{A}$ meetsBefore $\tilde{B}(m_<)$	$(\tilde{A}\ m_< \tilde{B}) \Leftrightarrow (\tilde{a}_2\ s_< \tilde{B})$	$\tilde{B}$ metByBefore $\tilde{A}$
$\tilde{A}$ meetsExact $\tilde{B}(m_e)$	$(\tilde{A}\ m_e \tilde{B}) \Leftrightarrow (\tilde{a}_2\ s_e \tilde{B})$	$\tilde{B}$ metByExact $\tilde{A}$
$\tilde{A}$ meetsAfter $\tilde{B}$ $(m_>)$	$(\tilde{A}\ m_> \tilde{B}) \Leftrightarrow (\tilde{a}_2\ s_> \tilde{B})$	$\tilde{B}$ metByAfter $\tilde{A}$
$\tilde{A}$ overlaps $\tilde{B}$ (o)	$(\tilde{A}\ o\ \tilde{B}) \Leftrightarrow ((\tilde{a}_1 < \tilde{B}) \wedge (\tilde{a}_2\ i\ \tilde{B}))$	$\tilde{B}$ overlappedBy $\tilde{A}$
$\tilde{A}$ equalsBeforeBefore $\tilde{B}$ $(=_{<<})$	$(\tilde{A} =_{<<} \tilde{B}) \Leftrightarrow ((\tilde{a}_1\ s_< \tilde{B}) \wedge (\tilde{a}_2\ f_< \tilde{B}))$	$\tilde{B}$ equalsAfterAfter $\tilde{A}$
$\tilde{A}$ equalsBeforeExact $\tilde{B}(=_{<e})$	$(\tilde{A} =_{<e} \tilde{B}) \Leftrightarrow ((\tilde{a}_1\ s_< \tilde{B}) \wedge (\tilde{a}_2\ f_e \tilde{B}))$	$\tilde{B}$ equalsAfterExact $\tilde{A}$
$\tilde{A}$ equalsBeforeAfter $\tilde{B}$ $(=_{<>})$	$(\tilde{A} =_{<>} \tilde{B}) \Leftrightarrow ((\tilde{a}_1\ s_< \tilde{B}) \wedge (\tilde{a}_2\ f_> \tilde{B}))$	$\tilde{B}$ equalsAfterBefore $\tilde{A}$
$\tilde{A}$ equalsExactBefore $\tilde{B}$ $(=_{e<})$	$(\tilde{A} =_{e<} \tilde{B}) \Leftrightarrow ((\tilde{a}_1\ s_e \tilde{B}) \wedge (\tilde{a}_2\ f_< \tilde{B}))$	$\tilde{B}$ equalsExactAfter $\tilde{A}$
$\tilde{A}$ equalsExactExact $\tilde{B}(=_{ee})$:	$(\tilde{A} =_{ee} \tilde{B}) \Leftrightarrow ((\tilde{a}_1\ s_e \tilde{B}) \wedge (\tilde{a}_2\ f_e \tilde{B}))$	$\tilde{B}$ equalsExactExact $\tilde{A}$

	Tolerance Model Relation	*Rough Set Model Relations*
(a)	$--$ (before)	$<$, $m_<$
(b)	$-b$ (meets)	$m_<$, m_e, $m_>$
(c)	$-i$ (overlaps)	$m_>$, o, $fb_>$, $s_<$, $=_{<<}$
(d)	$-f$ (finishedBy)	$fb_>$, fb_e, $fb_<$, $=_{<<}$, $=_{<e}$, $=_{<>}$
(e)	$-f\ -+$ (contains)	$fb_<$, c, $=_{<>}$, $sb_>$
(f)	$-f\ bi$ (starts)	$s_<$, s_e, $s_>$, $=_{<<}$, $=_{e<}$, $=_{><}$
(g)	$-f\ bf$ (equals)	any one of all 9 equalities ($=_{<<}$, $=_{<e}$, etc.)

From Rough Set Model to Toleranced Real Model. Two rough intervals can have 33 Rough Set Model relations among them. The corresponding relations in Toleranced Real Model for the corresponding real points and real intervals can be determined as in the earlier cases, but are omitted here.

5 Algebraic Aspects: A Preliminary Study

Rough sets have been extensively studied from the algebraic viewpoint (cf. [3]). In particular, a study in the context of *relation algebras* [6] may be found, for instance, in the work of Düntsch [4] where, following Tarski, a generalized notion of a 'rough relation algebra' is defined. Our interest here is slightly different. It is well-known that points and intervals on the rationals or reals constitute basic examples of relation algebras. We investigate the relational algebraic structure obtained from the rough points defined here.

Let us consider the communicative space $\mathbb{C}$ and the field $\mathcal{R}(\mathbb{C})$ of binary relations over $\mathbb{R}/\epsilon$ (the collection of all rough points in $\mathbb{C}$), i.e.

$$\mathcal{R}(\mathbb{C}) \equiv (\mathcal{P}(\mathbb{R}/\epsilon \times \mathbb{R}/\epsilon), \cup, ^c, \emptyset, \mathbb{R}/\epsilon \times \mathbb{R}/\epsilon, ^\smile, ;, 1'),$$

where $\mathcal{P}$ represents the power set, $^\smile$ is the converse operation, ; the composition operation and $1'$ the identity relation.

Now let us look at the set of five relations between rough points on $\mathbb{C}$ (cf. Section 2.1), $X_0 \equiv \{<, =_<, =_e, =_>, >\}$, and the subalgebra $\mathcal{S}(X_0)$ of $\mathcal{R}(\mathbb{C})$ generated by X_0. As noted in Observation 2(5), $<$ is $>^\smile$, and $=_<$ is $=_>^\smile$. In the following proposition, we mention some results of composition of the relations in X_0. For any relation R, let $R\ ;^n R$ denote $R; R; \ldots; R$ (n times).

Proposition 6

1. $<;^{n+1} < \ \subset\ <;^n < \ \subset\ \ldots\ \subset\ <;< \ \subset\ <$.
2. $<;^n < \ =\ <; (=_<;^{2n-1} =_<)$.
3. $=_<;^n =_< \ \subset\ <$.
   However, $=_<;^n =_<$ *is not comparable with* $=_<;^m =_<$, *where* $m \neq n$.

Using Observation 1, we see that the set of relations obtained by the composition and converse operations on elements of X_0 is isomorphic to the set $\mathcal{Z}$ of integers. Moreover, from Proposition 6 we conclude that the subalgebra $\mathcal{S}(X_0)$ is infinite. Further, closure with respect to $\cup$, $\cap$, and complementation (to make a Boolean algebra) gives that $\mathcal{S}(X_0)$ is isomorphic to the Boolean algebra of all finite and co-finite subsets of $\mathcal{Z}$. Thus, $\mathcal{S}(X_0)$ is isomorphic to a subalgebra of the complex algebra [6] $(\mathcal{P}(\mathcal{Z}), \cup, ^c, ^\smile, ;, 1')$ of the group $(\mathcal{Z}, +, 0)$.[1]

6 Conclusion

In this paper we present an approach for mapping quantities in a communicative approximation space where indistinguishability relations are modeled through rough intervals. In this work, we assume that intervals exhibit similar tolerances at both end points; where this does not hold, one needs to construct a formalism for asymmetric relations. The rough interval formalism introduced here is aimed merely at capturing the communication tolerances where explicit quantities are mentioned, how such tolerances are to be identified remains a complex question in pragmatics, and is beyond the scope of the present work.

A preliminary study of the relational algebraic aspects of the constructs defined here, has been reported in this article. Much more needs to be investigated, for instance, structures that are formed by rough intervals. In any case, it is clear that we shall obtain an infinite relation algebra for rough intervals as well.

Finally, we have assumed that the communicative tolerance ϵ is about the same as the observation tolerance $\pm\tau$. However, sometimes these two may be quite disparate – e.g. we may read that the time is "9:23.43", but we may not use such an accuracy in reporting it if we know that the listener has no use for such precision. Thus, situations with asymmetric observational and communicational tolerance also deserve further analysis, which has not been attempted here.

[1] Discussions with Robin Hirsch and Ian Hodkinson helped in relating our algebra to the group relation algebra over $\mathcal{Z}$.

Another important aspect is that of making transitive inferences between communicatively specified events. There is a large literature on complexity classes associated with transitive inference; for interval algebras defined on the real line, subalgebras involving contiguous relations are usually found to be tractable, whereas the full algebras are generally NP-hard [13]. We suspect this may also be the case for transitivity here, but this requires formal verification.

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. CACM 26(11), 832–843 (1983)
2. Asher, N., Vieu, L.: Toward a geometry of common sense – a semantics and a complete axiomatization of mereotopology. In: IJCAI 1995, pp. 846–852 (1995)
3. Banerjee, M., Chakraborty, M.K.: Algebras from rough sets. In: Pal, S.K., Polkowski, L., Skowron, A. (eds.) Rough-neuro Computing: Techniques for Computing with Words, pp. 157–184. Springer, Berlin (2004)
4. Düntsch, I.: Rough sets and algebras of relations. In: Orłowska, E. (ed.) Incomplete Information: Rough Set Analysis, pp. 95–108. Physica-Verlag, Heidelberg (1998)
5. Luce, R.D., Narens, L.: Measurement of scales on the continuum. Science 236, 1527–1532 (1987)
6. Maddux, R.D.: Relation Algebras. Elsevier, Amsterdam (2006)
7. Mukerjee, A., Schnorrenberg, F.: Hybrid systems: reasoning across scales in space and time. In: AAAI Symposium on Principles of Hybrid Reasoning, Asilomar, CA, November 15-17 (1991)
8. Orłowska, E., Pawlak, Z.: Measurement and indiscernibility. Bull. Polish Acad. Sci. (Th. Comp. Sc.) 32(9-10), 617–624 (1984)
9. Pawlak, Z.: Rough sets. Int. J. Computer and Information Science 11(5), 341–356 (1982)
10. Pawlak, Z.: Rough classification. Int. J. Man-Machine Studies 20, 469–483 (1984)
11. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic, Dordecht (1991)
12. Polkowski, L., Skowron, A.: Rough mereological calculi of granules: a rough set approach to computation. Computational Intelligence 17(3), 472–492 (2001)
13. Renz, J., Nebel, B.: On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus. Artificial Intelligence 108(1), 69–123 (1999)
14. Scott, D., Suppes, P.: Foundational aspects of theories of measurement. J. Symb. Logic 28, 113–128 (1958)
15. Taylor, B.N., Kuyatt, C.E.: Guidelines for evaluating and expressing the uncertainty of NIST measurement results. Technical Report 1297, NIST (1994)
16. Varzi, A.C.: Vagueness. In: Nadel, L., et al. (eds.) Encyclopedia of Cognitive Science, pp. 459–464. Macmillan and Nature Publishing Group, London (2003)
17. Warmus, M.: Calculus of approximations. Bull. Polish Acad. Sci. 4(5), 253–257 (1956)

On the Correctness of Rough-Set Based Approximate Reasoning*

Patrick Doherty[1] and Andrzej Szałas[1,2]

[1] Dept. of Computer and Information Science
Linköping University, SE-581 83 Linköping, Sweden
patdo@ida.liu.se
[2] Institute of Informatics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
andsz@mimuw.edu.pl

Abstract. There is a natural generalization of an indiscernibility relation used in rough set theory, where rather than partitioning the universe of discourse into indiscernibility classes, one can consider a covering of the universe by similarity-based neighborhoods with lower and upper approximations of relations defined via the neighborhoods. When taking this step, there is a need to tune approximate reasoning to the desired accuracy. We provide a framework for analyzing self-adaptive knowledge structures. We focus on studying the interaction between inputs and output concepts in approximate reasoning. The problems we address are:

- given similarity relations modeling approximate concepts, what are similarity relations for the output concepts that guarantee correctness of reasoning?
- assuming that output similarity relations lead to concepts which are not accurate enough, how can one tune input similarities?

1 Introduction

There is a natural generalization of relational databases where one uses intuitions from rough set theory [13] and rather than storing and querying crisp relations, one stores and queries rough relations consisting of an upper and lower approximation of the implicit crisp relation whose definition one tries to approximate [3,8]. There is also a natural generalization of an indiscernibility relation used in rough set theory, where rather than partitioning the universe of discourse U into indiscernibility classes, one can consider a covering of U by similarity-based neighborhoods (see, e.g., [5,11,14,15,16]) with lower and upper approximations of relations defined via the neighborhoods. To mark the difference, we will use the terms approximate relations and approximate databases instead of rough relations and rough databases. Approximate databases have been shown to be quite versatile in many application areas requiring the use of approximate knowledge structures [4,5].

* Partially supported by grant N N206 399134 from Polish MNiSW and grants from the Swedish Foundation for Strategic Research (SSF) Strategic Research Center MOVIII and the Swedish Research Council (VR) Linnaeus Center CADICS.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 327–336, 2010.

In [4] a framework for the specification, construction and management of approximate knowledge structures for intelligent artifacts has been proposed. The structures used there are called approximation transducers and approximation trees and the underlying framework is based on a generalization of deductive database technology. It is assumed that certain *primitive* concepts have been acquired, e.g., through a learning process where approximations of concepts are induced from the data. It is important to emphasize that the induced concepts are fluid in the sense that additional learning may modify the concept. Assuming these primitive concepts as given one then uses them as the *ur*-elements in knowledge representation structures. One can view this idea as *webs of imprecise knowledge*, gradually incremented with additional approximate and sometimes crisp facts and knowledge. Approximate definitions of concepts appear to be the rule rather than the exception.

Specifically, webs of approximate knowledge, as proposed in [4], are constructed from primitive concepts together with *approximation transducers* providing an approximate definition of one or more output concepts in terms of a set of input concepts and consist of three components:

- an input consisting of one or more approximate concepts, some of which might be primitive
- an output consisting of one or more new and possibly more abstract concepts defined partly in terms of the input concepts
- a local logical theory specifying constraints or dependencies between the input concepts and the output concepts. The theory may also refer to other concepts not expressed in the input.

The local logical theory specifies dependencies or constraints an expert for the application domain would be able to specify. Generally the form of the constraints would be in terms of some necessary and some sufficient conditions for the output concept. The local theory is viewed as a set of *crisp* logical constraints specified in the language of first-order logic. During the generation of the approximate concept output by the transducer, the crisp relations mentioned in the local theory are substituted with the actual approximate definitions of the input. Either lower or upper approximations of the input concepts may be used in the substitution. The resulting output specifies the output concept in terms of newly generated lower and upper approximations. It may then be used as input to other transducers creating *approximation trees*. The resulting tree represents a web of approximate knowledge capturing intricate and complex dependencies among an agent's conceptual vocabulary.

When taking this step, there is a need to ensure a form of stability or correctness of approximate reasoning. In the current paper we focus on studying the interaction between approximate concepts constituting reasoning inputs and outputs. The problems we address are:

- given similarity relations modeling approximate concepts, what are similarity relations for the output concepts that guarantee correctness of reasoning?
- assuming that output similarity relations lead to concepts which are not accurate enough, how can one tune input similarities?

The paper is structured as follows. Section 2 recalls necessary preliminaries. In Section 3 we discuss and motivate our definition of correctness of approximate reasoning. Section 4 illustrates the use of the general technique on first-order formulas and natural definitions of their approximations. In Section 5 we discuss how the proposed methodology can be used in tuning approximate knowledge structures. Finally, Section 6 concludes the paper.

2 Preliminaries

Let U be a set and $\sigma \subseteq U \times U$ a binary relation, further called a *similarity relation*. For any set $A \subseteq U$, the *lower approximation of A w.r.t. σ*, denoted by A_{σ^+} and the *upper approximation of A w.r.t. σ*, denoted by $A_{\sigma^\oplus}$ are defined as follows:

$$A_{\sigma^+} \stackrel{\text{def}}{=} \{x \mid \forall y[\sigma(x,y) \rightarrow A(y)]\} \tag{1}$$

$$A_{\sigma^\oplus} \stackrel{\text{def}}{=} \{x \mid \exists y[\sigma(x,y) \wedge A(y)]\}. \tag{2}$$

We will use second-order quantifier elimination, in particular the technique of [7] as well as techniques for analyzing correspondences between similarities and approximations developed in [9].

As a basis for doing quantifier elimination, we will use the following lemma of Ackermann [1] (see also, e.g., [7,10]), where $\Psi\left[P(\overline{\alpha}) \leftarrow [\Phi]^{\overline{x}}_{\overline{\alpha}}\right]$ means that every occurrence of P in Ψ is to be replaced by Φ where the actual arguments $\overline{\alpha}$ of P, replaces the variables of $\overline{x}$ in Φ (and the bound variables are renamed if necessary).

Lemma 1. *Let P be a predicate variable and let Φ and $\Psi(P)$ be first–order formulas such that Φ contains no occurrences of P. Then:*

- *if $\Psi(P)$ is positive w.r.t. P then*

$$\exists P\left\{\forall \overline{x}\,[P(\bar{x}) \rightarrow \Phi(\bar{x},\bar{y})] \wedge \Psi(P)\right\} \;\equiv\; \Psi\left[P(\overline{\alpha}) \leftarrow [\Phi]^{\overline{x}}_{\overline{\alpha}}\right]$$

- *if $\Psi(P)$ is negative w.r.t. P then*

$$\exists P\left\{\forall \overline{x}\,[\Phi(\bar{x},\bar{y}) \rightarrow P(\bar{x})] \wedge \Psi(P)\right\} \;\equiv\; \Psi\left[P(\overline{\alpha}) \leftarrow [\Phi]^{\overline{x}}_{\overline{\alpha}}\right]. \qquad \triangleleft$$

3 Correctness of Approximate Reasoning

In the reminder of the paper we mainly focus on sets. Of course, relations are sets of tuples, so are covered, too. Observe also that there is a natural correspondence between sets and formulas. Namely, a formula defines a set of tuples satisfying the formula. We shall sometimes use both sets and formulas, where it does not lead to a misunderstanding.

Let sets $A, B, \ldots, C$ and similarity relations $\sigma_A, \sigma_B, \ldots, \sigma_C$ be given. Assume we define a new set $R = \Gamma_R(A, B, \ldots, C)$ (see Figure 1). Since the sets $A, B, \ldots, C$

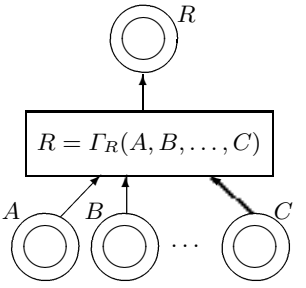


Fig. 1. A schema of defining approximate set R on the basis of approximations of input sets $A, B, \ldots, A_k$

are only given by their approximations $\tilde{A} = \left\langle A_{\sigma_A^+}, A_{\sigma_A^\oplus} \right\rangle$, $\tilde{B} = \left\langle B_{\sigma_B^+}, B_{\sigma_B^\oplus} \right\rangle, \ldots,$ $\tilde{C} = \left\langle C_{\sigma_C^+}, C_{\sigma_C^\oplus} \right\rangle$, R is also only approximated, so we consider:

$$\tilde{\Gamma}_R(\tilde{A}, \tilde{B}, \ldots, \tilde{C}) \stackrel{\text{def}}{=} \left\langle \tilde{\Gamma}_R^+(\tilde{A}, \tilde{B}, \ldots, \tilde{C}), \tilde{\Gamma}_R^\oplus(\tilde{A}, \tilde{B}, \ldots, \tilde{C}) \right\rangle, \tag{3}$$

where the first coordinate, $\tilde{\Gamma}_R^+(\tilde{A}, \tilde{B}, \ldots, \tilde{C})$, serves as the lower approximation of R and the second coordinate, $\tilde{\Gamma}_R^\oplus(\tilde{A}, \tilde{B}, \ldots, \tilde{C})$, serves as the upper approximation of R.

Example 1. Assume we define R to be the disjunction $A \vee B$. Then $\Gamma_R(A, B) \stackrel{\text{def}}{=} A \vee B$ and for some σ_A, σ_B, we have that $\tilde{A} = \left\langle A_{\sigma_A^+}, A_{\sigma_A^\oplus} \right\rangle$ and $\tilde{B} = \left\langle B_{\sigma_B^+}, B_{\sigma_B^\oplus} \right\rangle$. An exemplary $\tilde{\Gamma}_R$ can be given by $\tilde{\Gamma}_R(\tilde{A}, \tilde{B}) \stackrel{\text{def}}{=} \left\langle A_{\sigma_A^+} \vee B_{\sigma_B^+},\ A_{\sigma_A^\oplus} \vee B_{\sigma_B^\oplus} \right\rangle$. ◁

Remark 1. Note that, for clarity of presentation, we deal with sets of objects of the same type. For example, if A is the set of red objects and B is the set of large objects, then σ_A is not the similarity on colors but on objects w.r.t. colors. Similarly, σ_B is a similarity of objects w.r.t. their size. Therefore, for example σ_R, where $R \stackrel{\text{def}}{=} A \vee B$, is a similarity on objects w.r.t. both color and size. ◁

The question is how well is the crisp set R approximated and whether the method expressed by (3) is correct. We then have the following definition.

Definition 1. *Let $\sigma_A, \sigma_B, \ldots, \sigma_C, \sigma$ be given similarity relations. We say that the method expressed by (3) is* correct w.r.t. *$\sigma_A, \sigma_B, \ldots, \sigma_C, \sigma$ provided that for all $A, B, \ldots, C$,*

$$\tilde{\Gamma}_R^+(\tilde{A}, \tilde{B}, \ldots, \tilde{C}) \subseteq R_{\sigma^+} \tag{4}$$

$$R_{\sigma^\oplus} \subseteq \tilde{\Gamma}_R^\oplus(\tilde{A}, \tilde{B}, \ldots, \tilde{C}). \tag{5}$$

i.e., the computed lower approximation of R is included in the actual lower approximation of R and the computed upper approximation of R includes the actual upper approximation of R. ◁

Since $R = \Gamma_R(A, B, \ldots, C)$, inclusions (4) and (5) are equivalent to:

$$\tilde{\Gamma}_R^+(\tilde{A}, \tilde{B}, \ldots, \tilde{C}) \subseteq \Gamma_R(A, B, \ldots, C)_{\sigma^+} \tag{6}$$

$$\Gamma_R(A, B, \ldots, C)_{\sigma^\oplus} \subseteq \tilde{\Gamma}_R^\oplus(\tilde{A}, \tilde{B}, \ldots, \tilde{C}). \tag{7}$$

4 Correctness of Computing First-Order Formulas

4.1 The Case of Negation

We define R as the negation of A, so $\Gamma_\neg(A) \stackrel{\text{def}}{=} \neg A$. Let us consider the method of computing negation given by:

$$\tilde{\Gamma}_\neg(\tilde{A}) \stackrel{\text{def}}{=} \left\langle -A_{\sigma_A^\oplus}, -A_{\sigma_A^+} \right\rangle. \tag{8}$$

According to Definition 1, the method expressed by (8) is correct provided that the following conditions hold:

$$\forall A\big[-A_{\sigma_A^\oplus} \subseteq (\neg A)_{\sigma^+}\big] \tag{9}$$

$$\forall A\big[(\neg A)_{\sigma^\oplus} \subseteq -A_{\sigma_A^+}\big]. \tag{10}$$

We have the following theorem.

Theorem 1. *Each of the formulas (9) and (10) is equivalent to*

$$\forall x \forall y[\sigma(x, y) \rightarrow \sigma_A(x, y)]. \tag{11}$$

Proof. Consider first formula (9). By (1) and (2), it is equivalent to

$$\forall A \forall x\big[\neg\exists y[\sigma_A(x, y) \wedge A(y)] \rightarrow \forall z[\sigma(x, z) \rightarrow \neg A(z)]\big],$$

i.e., to $\neg\exists x \exists A\big[\forall y[A(y) \rightarrow \neg\sigma_A(x, y)] \wedge \exists z[\sigma(x, z) \wedge A(z)]\big]$. Lemma 1 is now applicable and results in its equivalent $\neg\exists x \exists z\big[\sigma(x, z) \wedge \neg\sigma_A(x, z)\big]$, clearly equivalent to (11).

The proof for (10) is analogous. ◁

Note that any σ satisfying (11) guarantees that the method of computing negation expressed by (8) is correct w.r.t. σ_A, σ.

The maximal σ satisfying (11) is the one modeling the worst accuracy, given σ_A. Such σ is given by $\sigma(x, y) \stackrel{\text{def}}{=} \sigma_A(x, y)$. As a consequence, one also obtains that the method expressed by (8) is not correct for any σ not included in σ_A.

4.2 The Case of Disjunction

We define R as the disjunction $A \vee B$, so $\Gamma_\vee(A, B) \stackrel{\text{def}}{=} A \vee B$. Let us consider the method of computing disjunction, given by:

$$\tilde{\Gamma}_\vee(\tilde{A}, \tilde{B}) \stackrel{\text{def}}{=} \left\langle A_{\sigma_A^+} \cup B_{\sigma_B^+}, A_{\sigma_A^\oplus} \cup B_{\sigma_B^\oplus} \right\rangle. \tag{12}$$

According to Definition 1, the method expressed by (12) is correct provided that the following conditions hold:

$$\forall A \forall B \big[(A_{\sigma_A^+} \cup B_{\sigma_B^+}) \subseteq (A \vee B)_{\sigma^+} \big] \tag{13}$$

$$\forall A \forall B \big[(A \vee B)_{\sigma^\oplus} \subseteq (A_{\sigma_A^\oplus} \cup B_{\sigma_B^\oplus}) \big]. \tag{14}$$

We have the following theorem.

Theorem 2. *Each of the formulas (13) and (14) is equivalent to*

$$\forall x \forall y [\sigma(x,y) \rightarrow \big(\sigma_A(x,y) \wedge \sigma_B(x,y)\big)]. \tag{15}$$

Proof. Consider formula (13). By (1), it is equivalent to

$$\forall A \forall B \forall x \big[\forall y [\sigma_A(x,y) \rightarrow A(y)] \vee \forall y [\sigma_B(x,y) \rightarrow B(y)] \rightarrow \forall z [\sigma(x,z) \rightarrow (A(z) \vee B(z))] \big],$$

i.e., to

$$\neg \exists x \exists A \exists B \big[\forall y [\sigma_A(x,y) \rightarrow A(y)] \vee \forall y [\sigma_B(x,y) \rightarrow B(y)] \wedge \exists z [\sigma(x,z) \wedge \neg A(z) \wedge \neg B(z)] \big]$$

and further to

$$\neg \exists z \exists x \exists A \exists B \big[\forall y [\sigma_A(x,y) \rightarrow A(y)] \vee \forall y [\sigma_B(x,y) \rightarrow B(y)] \wedge \sigma(x,z) \wedge \forall u [A(u) \rightarrow u \neq z] \wedge \forall u [B(u) \rightarrow u \neq z] \big].$$

Two successive applications of Lemma 1 result now in

$$\neg \exists z \exists x \big[\forall y [\sigma_A(x,y) \rightarrow y \neq z] \vee \forall y [\sigma_B(x,y) \rightarrow y \neq z] \wedge \sigma(x,z) \big],$$

which is equivalent to $\neg \exists z \exists x \big[(\neg \sigma_A(x,z) \vee \neg \sigma_B(x,z)) \wedge \sigma(x,z) \big]$, i.e., to (15).

The proof for (14) is analogous. ◁

The maximal σ satisfying (15), modeling the worst accuracy, is given by

$$\sigma(x,y) \stackrel{\text{def}}{\equiv} \sigma_A(x,y) \wedge \sigma_B(x,y). \tag{16}$$

As a consequence, one obtains that the method expressed by (15) is not correct for any σ not included in $\sigma_A \wedge \sigma_B$.

4.3 The Case of Conjunction

We define R as the conjunction $A \wedge B$, so $\Gamma_\wedge(A,B) \stackrel{\text{def}}{=} A \wedge B$. Let us consider the method of computing disjunction, given by:

$$\tilde{\Gamma}_\wedge(\tilde{A}_1, \tilde{A}_2) \stackrel{\text{def}}{=} \left\langle A_{\sigma_A^+} \cap B_{\sigma_B^+}, A_{\sigma_A^\oplus} \cap B_{\sigma_B^\oplus} \right\rangle. \tag{17}$$

According to Definition 1, the method expressed by (17) is correct provided that the following conditions hold:

$$\forall A \forall B \big[(A_{\sigma_A^+} \cap B_{\sigma_B^+}) \subseteq (A \wedge B)_{\sigma^+} \big] \tag{18}$$

$$\forall A \forall B \big[(A \wedge B)_{\sigma^\oplus} \subseteq (A_{\sigma_A^\oplus} \cap B_{\sigma_B^\oplus}) \big]. \tag{19}$$

The following theorem can be proved similarly to Theorem 2.

Theorem 3. *Each of the formulas (18) and (19) is equivalent to (15).* ◁

The maximal σ that guarantees correctness of (17) is then also given by (16).

4.4 The Case of Existential Quantification

We define R as the existential quantification $\exists x[A(x,\bar{y})]$, where $\bar{y}$ is the tuple of all free variables in $A(x,\bar{y})$ except for x. Therefore, given a universe U, $\Gamma_{\exists x}(A(x,\bar{y}))$ is defined as the set of tuples of arity the same as $\bar{y}$, for which there is $w \in U$ such that $A(w,\bar{u})$ is satisfied in a given interpretation.

Let us consider the method of computing existential quantification given by:

$$\tilde{\Gamma}_{\exists}(\tilde{A}) \stackrel{\text{def}}{=} \left\langle \exists x\left[A(x,\bar{y})_{\sigma_A^+}\right], \exists x\left[A(x,\bar{y})_{\sigma_A^{\oplus}}\right]\right\rangle. \tag{20}$$

According to Definition 1, the method expressed by (20) is correct provided that the following conditions hold:

$$\forall A \forall \bar{y}\left[\exists x\left[A(x,\bar{y})_{\sigma_A^+}\right] \subseteq (\exists x[A(x,\bar{y})])_{\sigma^+}\right] \tag{21}$$

$$\forall A \forall \bar{y}\left[(\exists x[A(x,\bar{y})])_{\sigma^{\oplus}} \subseteq \exists x\left[A(x,\bar{y})_{\sigma_A^{\oplus}}\right]\right]. \tag{22}$$

Let $\bar{z} = \langle z_1, \ldots, z_k\rangle$ be a tuple of variables and x be a variable. Then $x\bar{z}$ stands for the concatenation of x and $\bar{z}$, i.e., $x\bar{z} \stackrel{\text{def}}{=} \langle x, z_1, \ldots, z_k\rangle$.

We have the following theorem.

Theorem 4. *Formula (21) is equivalent to (23) and formula (22) is equivalent to (24):*

$$\forall \bar{y}\forall \bar{z}\left[\sigma(\bar{y},\bar{z}) \rightarrow \forall x \exists u[\sigma_A(x\bar{y},u\bar{z})]\right] \tag{23}$$

$$\forall \bar{y}\forall \bar{z}\left[\sigma(\bar{y},\bar{z}) \rightarrow \forall x \exists u[\sigma_A(u\bar{y},x\bar{z})]\right]. \tag{24}$$

Proof. Let us prove the equivalence of (21) and (23).

Formula (21) is equivalent to $\forall A \forall \bar{y}\left[\exists x\left[A(x,\bar{y})_{\sigma_A^+}\right] \rightarrow (\exists x[A(x,\bar{y})])_{\sigma^+}\right]$, i.e., to

$$\neg\exists A \exists x \exists \bar{y}\left[A(x,\bar{y})_{\sigma_A^+} \wedge \neg(\exists x[A(x,\bar{y})])_{\sigma^+}\right].$$

By (1), this formula is equivalent to

$$\neg\exists A \exists x \exists \bar{y}\left[\forall x'\bar{y}'[\sigma_A(x\bar{y},x'\bar{y}') \rightarrow A(x',\bar{y}')] \wedge \neg\forall \bar{z}[\sigma(\bar{y},\bar{z}) \rightarrow \exists u[A(u,\bar{z})]]\right].$$

Applying Ackermann's lemma (Lemma 1), we obtain the equivalent formula

$$\neg\exists x \exists \bar{y}\left[\neg\forall \bar{z}[\sigma(\bar{y},\bar{z}) \rightarrow \exists u[\sigma_A(x\bar{y},u\bar{z})]\right].$$

i.e., $\forall \bar{y}\forall \bar{z}\left[\sigma(\bar{y},\bar{z}) \rightarrow \forall x \exists u[\sigma_A(x\bar{y},u\bar{z})]\right]$, which is exactly the required formula (23).

The proof of equivalence of (22) and (24) is analogous. ◁

The maximal σ satisfying (23) and (24) is then given by

$$\sigma(\bar{y},\bar{z}) \stackrel{\text{def}}{=} \forall x \exists u[\sigma_A(x\bar{y},u\bar{z})] \wedge \forall x \exists u[\sigma_A(u\bar{y},x\bar{z})]. \tag{25}$$

4.5 The Case of Universal Quantification

We define R as the universal quantification $\forall x[A(x,\bar{y})]$, where $\bar{y}$ is the tuple of all free variables in $A(x,\bar{y})$ except for x. Therefore, given a universe U, $\Gamma_{\exists}(\forall x[A(x,\bar{y})])$ is

defined as the set of tuples of arity the same as $\bar{y}$, for which $A(w, \bar{u})$ is satisfied in a given interpretation for all $w \in U$.

Let us consider the method of computing existential quantification given by:

$$\tilde{\Gamma}_{\forall}(\tilde{A}_1) \stackrel{\text{def}}{=} \left\langle \forall x\big[A(x,\bar{y})_{\sigma_A^+}\big], \forall x\big[A(x,\bar{y})_{\sigma_A^{\oplus}}\big] \right\rangle. \tag{26}$$

According to Definition 1, the method expressed by (26) is correct provided that the following conditions hold:

$$\forall A \forall \bar{y}\big[\forall x\big[A(x,\bar{y})_{\sigma_A^+}\big] \subseteq (\forall x[A(x,\bar{y})])_{\sigma^+}\big] \tag{27}$$

$$\forall A \forall \bar{y}\big[(\forall x[A(x,\bar{y})])_{\sigma^{\oplus}} \subseteq \forall x\big[A(x,\bar{y})_{\sigma_A^{\oplus}}\big]\big]. \tag{28}$$

The following theorem can be proved by applying the technique used in the proof of Theorem 4.

Theorem 5. *Formula (27) is equivalent to (24) and formula (28) is equivalent to (23).* ◁

The maximal σ is then given by (25).

Remark 2. In general, the result of quantifier elimination may not be as simple as in Theorems 1–5. In such cases, the maximal similarity relation can be computed using a suitable form of circumscription. In fact, second-order quantifier elimination is often successful in such cases (see [7] and for more advanced techniques, e.g., [10,6]). ◁

5 Tuning Input Relations

Observe that the results of the previous section give us a tool for tuning the accuracy of input relations to the required accuracy of the result. Namely, if the worst-case similarity relation is not satisfactory, one has to improve similarities on inputs. Sometimes this requires to tune sensors or to install better ones.

The following example illustrates this point.

Example 2. Let us define dangerous situations based on temperature readings and the robot's distance to the heat source:

$$D(s) \stackrel{\text{def}}{=} S(s) \wedge \exists t[H(s,t)], \tag{29}$$

where $D(s)$ states that situation s is dangerous, $S(s)$ states that in situation s the robot's distance to the heat source is small and $H(s,t)$ states that temperature t is considered high in situation s. The intended meaning of (29) is that a situation is dangerous if the robot's distance to the heat source is small and there is a reading of temperature which is considered high.

Here the resulting σ is the similarity on situations, σ_S which is a similarity on situations w.r.t. the robot's distance to the heat source and σ_H is the similarity on pairs consisting of situations and temperature measurements. According to Theorems 3 and 4,

$$\forall y \forall z[\sigma(y,z) \rightarrow \big(\sigma_S(y,z) \wedge \forall x \exists u[\sigma_H(xy,uz)] \wedge \forall x \exists u[\sigma_H(uy,xz)]\big)].$$

By formula (25), the worst-case σ is given by the conjunction

$$\sigma_S(y,z) \wedge \forall x \exists u[\sigma_H(xy,uz)] \wedge \forall x \exists u[\sigma_H(uy,xz)]. \tag{30}$$

Assume now that, for example, $\sigma(s,s')$ holds for situations s and s' such that s is considered dangerous and s' is considered not dangerous. In such a case one would not make them similarly dangerous, so one has to exclude the pair $\langle s,s'\rangle$ from the conjunction (30), which can only be done by shrinking σ_S or σ_H. Assuming that these similarities are based on the accuracy of measurements, this can only be done by tuning or replacing one of respective sensors (or or both of them). ◁

6 Conclusions

In the paper we have proposed a notion of correctness of approximate reasoning based on logical formalisms. To formally verify correctness of investigated techniques we used second-order quantifier elimination. In fact, all calculations used in the paper can automatically be carried out using the DLS algorithm of [7]. The use of second-order logic makes the proposed method rather general and widely applicable.

Many calculations can also be automated by using the algorithm SQEMA [2] by noticing that approximations may be expressed as modalities.

References

1. Ackermann, W.: Untersuchungen über das eliminationsproblem der mathematischen logik. Mathematische Annalen 110, 390–413 (1935)
2. Conradie, W., Goranko, V., Vakarelov, D.: Algorithmic correspondence and completeness in modal logic: I. The core algorithm SQEMA. Logical Methods in Computer Science 2(1-5), 1–26 (2006)
3. Doherty, P., Kachniarz, J., Szałas, A.: Using contextually closed queries for local closed-world reasoning in rough knowledge databases. In: Pal, et al [12]
4. Doherty, P., Łukaszewicz, W., Skowron, A., Szałas, A.: Approximation transducers and trees: A technique for combining rough and crisp knowledge. In: Pal, et al [12]
5. Doherty, P., Łukaszewicz, W., Skowron, A., Szałas, A.: Knowledge Representation Techniques. A Rough Set Approach. Studies in Fuziness and Soft Computing, vol. 202. Springer, Heidelberg (2006)
6. Doherty, P., Łukaszewicz, W., Szałas, A.: A reduction result for circumscribed semi-Horn formulas. Fundamenta Informaticae 28(3-4), 261–271 (1996)
7. Doherty, P., Łukaszewicz, W., Szałas, A.: Computing circumscription revisited. Journal of Automated Reasoning 18(3), 297–336 (1997)
8. Doherty, P., Magnusson, M., Szałas, A.: Approximate databases: A support tool for approximate reasoning. Journal of Applied Non-Classical Logics 16(1-2), 87–118 (2006); Special issue on Implementation of logics
9. Doherty, P., Szałas, A.: On the correspondence between approximations and similarity. In: Tsumoto, S., Slowinski, R., Komorowski, J., Grzymala-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 143–152. Springer, Heidelberg (2004)
10. Gabbay, D.M., Schmidt, R., Szałas, A.: Second-Order Quantifier Elimination. In: Foundations, Computational Aspects and Applications. Studies in Logic, vol. 12. College Publications (2008)

11. Liau, C.-J.: An overview of rough set semantics for modal and quantifier logics. Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 8(1), 93–118 (2000)
12. Pal, S.K., Polkowski, L., Skowron, A. (eds.): Rough-Neuro Computing: Techniques for Computing with Words. Springer, Heidelberg (2003)
13. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
14. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
15. Słowiński, R., Vanderpooten, D.: Similarity relation as a basis for rough approximations. In: Wang, P. (ed.) Advances in Machine Intelligence & Soft Computing, Raleigh, NC, pp. 17–33. Bookwrights (1997)
16. 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)

Unit Operations in Approximation Spaces

Zbigniew Bonikowski

Institute of Mathematics and Informatics
Opole University, Poland
zbonik@math.uni.opole.pl

Abstract. Unit operations are some special functions on sets. The concept of the unit operation originates from researches of U.Wybraniec-Skardowska. The paper is concerned with the general properties of such functions. The isomorphism between binary relations and unit operations is proved. Algebraic structures of families of unit operations corresponding to certain classes of binary relations are considered. Unit operations are useful in Pawlak's Rough Set Theory. It is shown that unit operations are upper approximations in approximation space. We prove, that in the approximation space (U, R) generated by a reflexive relation R the corresponding unit operation is the least definable approximation if and only if the relation R is transitive.

Keywords: unit operations, approximation space, upper approximations, binary relation.

1 Introduction

The basic notion of rough set theory proposed by Pawlak [12,13] is an approximation space. Originally an approximation space is a pair consisting of a set U of objects (called *universe*) and an equivalence relation on U. This equivalence relation generates a partition of the universe. Equivalence classes of considered relation may be treated as elementary granules of information. By *granule of information* we mean a clump of objects which are drawn towards an object (Lin [10]). To characterize subsets of the universe two operations are considered in the rough set theory: a lower approximation and an upper approximation.

The theory of rough sets based on an equivalence relation is not useful in some applications (e.g.. in an analysis of incomplete information tables (Stefanowski, Tsioukàs [15], Słowiński [14], Grzymała-Busse [5,6])). Therefore many generalizations of the notion of approximation space were created. We mention two directions of generalization of approximation space. Firstly, approximation space may be considered as a pair consisting of a universe U and a covering of U, i.e. a family of non-empty subsets of U whose union is U (Bonikowski, Bryniarski, Wybraniec-Skardowska [1], Bonikowski [2], Liu, Sai [11]). On the other hand, one can consider any binary relation on U (Yao [18], Słowiński [14], Grzymała-Busse, Rząsa [7], Zhu [20,21]).

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 337–346, 2010.

In this paper by approximation space we mean a pair consisting of a universe U and a binary relation R on U. We show, that in this case we can use special mappings in the power set of U called *unit operation*. The concept of the unit operation originates from researches of U.Wybraniec-Skardowska. Unit operations are upper approximations in approximation space. There are many approximation mappings (Gomolińska [3,4], Grzymała-Busse, Rząsa [7], Wybraniec-Skardowska [16]). These approximations are defined using elementary granules generated by the relation R. Approximations, which values are definable, are very important, especially in data mining. A set is *definable* if it is a union of elementary granules of information. Unit operations may be treated as "standard" approximations for all relations. Moreover, under special conditions, there are the best upper approximations.

The paper is organized as follows. In Section 2 a notion of unit operation and connection between unit operations and binary relations are presented. In the next section we consider algebraic properties of unit operations. In Section 4 we recall notions of an approximation space and approximation mappings. Next we investigate some properties of unit operations treated as upper approximations due to binary relations, which correspond to the above operations. Section 5 contains a brief summary.

2 Unit Operations

Let U be a non-empty set. The family of all subsets of U will be denoted by $P(U)$.

Definition 1. *A function $f\colon P(U) \to P(U)$ is called a* ***unit operation*** *if*

$$f(X) = \bigcup_{x \in X} f(\{x\}) \quad \text{for any } X \subseteq U. \tag{1}$$

Let us recall some definitions of properties of functions (see e.g. Jónsson, Tarski [8]).

Definition 2. *A function $f\colon P(U) \to P(U)$ is called:*

(a) ***normal*** *if $f(\emptyset) = \emptyset$,*
(b) ***monotonic*** *if $X \subseteq Y$ implies $f(X) \subseteq f(Y)$ for any $X, Y \subseteq U$,*
(c) ***additive*** *if $f(X \cup Y) = f(X) \cup f(Y)$ for any $X, Y \subseteq U$,*
(d) ***completely additive*** *if $f\left(\bigcup_{t \in T} X_t\right) = \bigcup_{t \in T} f(X_t)$.*

It is easy to prove ([17]), that:

Proposition 1. *Every unit operation is normal, monotonic, additive and completely additive function.*

Proposition 2. *Let $f\colon P(U) \to P(U)$ be a function. The function f is a unit operation if and only if the function f is normal and completely additive.*

Definition 3. *Let $f\colon P(U) \to P(U)$ be a function. By R_f we denote the relation:*

$$R_f = \{(x, y) \in U \times U : y \in f(\{x\})\}. \tag{2}$$

Definition 4. *Let $R \subseteq U \times U$ be a binary relation. An image relation determined by R (R-image relation) is a function $\overrightarrow{R}\colon P(U) \to P(U)$ such that:*

$$\overrightarrow{R}(X) = \{y \in U : \exists x \in X.(x, y) \in R\}. \tag{3}$$

In particular, for every $x \in U$:

$$\overrightarrow{R}(\{x\}) = \{y \in U : (x, y) \in R\}. \tag{4}$$

R-image relations of singletons of U are the same as well known from literature *neighborhoods* (Lin [10], Yao [19]), *similarity classes* (Słowiński [14]) or *R-successors* (Grzymała, Rząsa [7]).

Lemma 1. *Let $R \subseteq U \times U$ be a binary relation. The R-image relation is a unit operation.*

Lemma 2. *Let $R, S \subseteq U \times U$ be binary relations. If $\overrightarrow{R} = \overrightarrow{S}$ then $R = S$.*

Lemma 3. *Let $f\colon P(U) \to P(U)$ be a unit operation. The R_f-image relation is equal to f.*

Lemma 4. *Let $f\colon P(U) \to P(U)$ be a function. If there exists a relation $R \subseteq U \times U$ such that $\overrightarrow{R} = f$, then the function f is a unit operation.*

Let $\mathcal{R}$ denote the family of all binary relations and $\mathcal{F}$ denote the family of all unit operations.

Theorem 1. *The families $\mathcal{R}$ and $\mathcal{F}$ are bijective.*

Proof. Let $g\colon \mathcal{R} \to \mathcal{F}$ be defined for any $R \in \mathcal{R}$ as follows:

$$g(R) = \overrightarrow{R}. \tag{5}$$

By Lemma 1, g is well defined. Let us recall that a function is a bijection if it is one-to-one (injection) and onto (surjection). From Lemma 2 it follows that g is an injective mapping. Lemma 3 shows that g is a surjection. Therefore the function g is a bijection. □

Corollary 1. *If $|U| = n$, then $|\mathcal{F}| = 2^{n^2}$.*

Proof. Let U be an n-element set. Hence there are 2^{n^2} subsets of $U \times U$. By Theorem 1, there are 2^{n^2} different unit operations, too. □

According to types of binary relation we may consider different types of unit operations. In particular, unit operation corresponding to reflexive (symmetric, antisymmetric, transitive, tolerance, equivalence) relation will be called *reflexive* (*symmetric, antisymmetric, transitive, tolerance, equivalence*) *unit operation.*

Proposition 3. *Let f be a unit operation.*

(a) f is reflexive	*iff* $x \in f(\{x\})$	*for any* $x \in U$,
(b) f is symmetric	*iff* $y \in f(\{x\}) \iff x \in f(\{y\})$	*for any* $x, y \in U$,
(c) f is antisymmetric	*iff* $y \in f(\{x\} \wedge x \in f(\{y\}) \Rightarrow x = y$	*for any* $x, y \in U$,
(d) f is transitive	*iff* $f(f(\{x\})) \subseteq f(\{x\})$	*for any* $x \in U$.

Proof. The proof is easy, so we will prove illustrative only (d).

($\Rightarrow$) Assume f be transitive. Hence the relation R_f is transitive. Let $x \in X$ and $z \in f(f(\{x\}))$. By Lemma 3, $z \in \overrightarrow{R_f}(f(\{x\}))$. By (3), there is $w \in f(\{x\})$ such that $(w, z) \in R_f$. Because $f(\{x\}) = \overrightarrow{R_f}(\{x\})$, then $(x, w) \in R_f$. R_f is transitive, which gives $(x, z) \in R$. Hence $z \in \overrightarrow{R_f}(\{x\}) = f(\{x\})$.

($\Leftarrow$) Let $f(f(\{x\})) \subseteq f(\{x\})$ for any $x \in U$, $x, y, z \in U$, $(x, y) \in R_f$ and $(y, z) \in R_f$. Hence $y \in \overrightarrow{R_f}(\{x\})$ and $z \in \overrightarrow{R_f}(\{y\})$. Because $y \in \overrightarrow{R_f}(\{x\}) = f(\{x\})$, then $f(\{y\}) \subseteq f(f(\{x\}))$ by monoticity of unit operations. Since $z \in \overrightarrow{R_f}(\{y\}) = f(\{y\})$, we conclude by assumption, that $z \in f(\{x\})$. Hence $(x, z) \in R_f$. The relation R_f is then transitive, so f is transitive. □

3 Algebraic Structures of Unit Operations

Let us define some operations in the family of unit operations.

Definition 5. *Let f and g be unit operations, X be a subset of U.*

(a) $(f \oplus g)(X) = f(X) \cup g(X)$
(b) $(f \otimes g)(X) = \bigcup_{x \in X}(f(\{x\}) \cap g(\{x\}))$
(c) $(\ominus f)(X) = \bigcup_{x \in X}(-f(\{x\}))$

Let us observe, that:

$$(f \oplus g)(\{x\}) = f(\{x\}) \cup g(\{x\}) \tag{6}$$
$$(f \otimes g)(\{x\}) = f(\{x\}) \cap g(\{x\}) \tag{7}$$
$$(\ominus f)(\{x\}) = -f(\{x\}) \tag{8}$$

Proposition 4. *Let f and g be unit operations.*

(a) $f \oplus g$ is a unit operation.
(b) $f \otimes g$ is a unit operation.
(c) $\ominus f$ is a unit operation.

Proof. Let f and g be unit operations and $X \subseteq U$.

$$(f \oplus g)(X) \overset{Def.\ 5}{=} f(X) \cup g(X) \overset{(1)}{=} \bigcup_{x \in X} f(\{x\}) \cup \bigcup_{x \in X} g(\{x\})$$
$$= \bigcup_{x \in X}(f(\{x\}) \cup g(\{x\}) \overset{(6)}{=} \bigcup_{x \in X}(f \oplus g)(\{x\}).$$
$$(f \otimes g)(X) \overset{Def.\ 5}{=} \bigcup_{x \in X}(f(\{x\}) \cap g(\{x\})) \overset{(7)}{=} \bigcup_{x \in X}(f \otimes g)(\{x\}).$$
$$(\ominus f)(X) \overset{Def.\ 5}{=} \bigcup_{x \in X}(-f(\{x\})) \overset{(8)}{=} \bigcup_{x \in X}((\ominus f)(\{x\})).$$

□

Let us denote by $f_{\mathbf{0}}$ and $f_{\mathbf{1}}$ the following special unit operations:

$$f_{\mathbf{0}}(X) = \emptyset, \quad \text{for any} X \subseteq U. \tag{9}$$

$$f_{\mathbf{1}}(X) = \begin{cases} \emptyset, & \text{for } X = \emptyset, \\ U, & \text{for any non-empty } X \subseteq U. \end{cases} \tag{10}$$

Theorem 2. *The families* $\langle \mathcal{R}, \cup, \cap, -, \emptyset, U \times U \rangle$ *and* $\langle \mathcal{F}, \oplus, \otimes, \ominus, f_{\mathbf{0}}, f_{\mathbf{1}} \rangle$ *are isomorphic.*

Proof. Let $g: \mathcal{R} \to \mathcal{F}$ be defined by (5). From Theorem 1 it follows that G is a bijection. It remains to prove that g preserves operations.

Assume R and S are binary relations and $X \subseteq U$. First we will show, that $g(R \cup S) = g(R) \oplus g(S)$.

$$\begin{aligned} g(R \cup S)(X) &\overset{(5)}{=} \overrightarrow{R \cup S}(X) \overset{(3)}{=} \{y \in U : \exists x \in X.(x, y) \in R \cup S)\} \\ &= \{y \in U : \exists x \in X.((x, y) \in R \vee (x, y) \in S)\} \\ &= \{y \in U : \exists x \in X.((x, y) \in R \text{ or } \exists x \in X.(x, y) \in S)\} \\ &= \{y \in U : \exists x \in X.(x, y) \in R\} \cup \{y \in U : \exists x \in X.(x, y) \in R\} \\ &\overset{(3)}{=} \overrightarrow{R}(X) \cup \overrightarrow{S}(X) = (g(R) \oplus g(S))(X) \end{aligned}$$

Now assume $z \in X$ and $z \in g(R \cap S)(X) = \overrightarrow{R \cap S}(X)$. By (3), there exists $x \in X$ such that $(x, z) \in R \cap S$. Hence $(x, z) \in R$ and $(x, z) \in S$. By (3), $z \in \overrightarrow{R}(\{x\}) \cap \overrightarrow{S}(\{x\})$. Thus:

$$z \in \bigcup_{x \in X} \left(\overrightarrow{R}(\{x\}) \cap \overrightarrow{S}(\{x\}) \right) = \bigcup_{x \in X} (g(R)(\{x\}) \cap g(S)(\{x\})) = (g(R) \otimes g(S))(X).$$

Similar considerations apply to inverse inclusion. Therefore $g(R \cap S) = g(R) \otimes g(S)$.

The proof of $g(-R) = \ominus g(R)$ is similar.

Let us observe furthermore that

$$g(\emptyset)(X) = \{y \in U : \exists x \in X.(x, y) \in \emptyset\} = \emptyset \quad \text{for any } X \subseteq U,$$

$$g(U \times U)(X) = \{y \in U : \exists x \in X.(x, y) \in U \times U\} = \begin{cases} \emptyset, & \text{for } X = \emptyset, \\ U, & \text{for } \emptyset \neq X \subseteq U. \end{cases}$$

It is easily seen that $g(\emptyset) = f_{\mathbf{0}}$ and $g(U \times U) = f_{\mathbf{1}}$. □

Corollary 2. *The algebra* $\langle \mathcal{F}, \oplus, \otimes, \ominus, f_{\mathbf{0}}, f_{\mathbf{1}} \rangle$ *of unit operations is a Boolean algebra.*

Proof. The family $\langle \mathcal{R}, \cup, \cap, -, \emptyset, U \times U \rangle$ of all binary relations with standard set-theoretical operations is a Boolean algebra. By Theorem 2, the family of unit operations is a Boolean algebra, too. □

Proposition 5. *The algebra* $\langle \mathcal{F}_r, \oplus, \otimes, f_{\mathbf{0}}, f_{\mathbf{1}} \rangle$ *of reflexive unit operations is a distributive lattice with zero and unit.*

Proposition 6. *The algebra* $\langle \mathcal{F}_s, \oplus, \otimes, f_{\mathbf{0}}, f_{\mathbf{1}} \rangle$ *of symmetric unit operations is a distributive lattice with zero and unit.*

Proposition 7. *The algebra* $\langle \mathcal{F}_t, \otimes, f_{\mathbf{1}} \rangle$ *of transitive unit operations is a commutative monoid.*

In the proofs of the above propositions it is sufficient to show that the algebra of reflexive relations and the algebra of symmetric relations are distributive lattices with zero and unit, and algebra of transitive relations is a commutative monoid. The proofs are standard.

4 Approximation Space

Definition 6. *Let* U *be a finite, non-empty set called a universe and* R *be a binary relation on* U*. The ordered pair* $\mathcal{A} = (U, R)$ *is called an* ***approximation space****.*

Some special types of relations were extensively analyzed. These are: tolerance relations (Kryszkiewicz [9]) and similarity relations (Słowiński [14]). Moreover several papers have been published, in which there are not any assumptions about the relation R (Grzymała, Rząsa [7], Zhu [20,21]).

In an approximation space we can represent any subset $X \subseteq U$ by a pair of sets, called the lower and upper approximation. Approximations are usually defined using granules of information. In the case of an approximation space generated by a binary relation, granules are classes of this relation (in this paper R-image relations of singletons of U).

Very important property of subsets of U is a property of *definability*.

Definition 7. *Let* $X \subseteq U$*.* X *is called* ***definable*** *if* X *is a union of granules of information, i.e. there is a set* $Z \subseteq U$ *such that* $(X = \bigcup_{z \in Z} \overrightarrow{R}(\{z\}))$.

We accept (with slight modification) postulates for approximation mappings formulated by [3]. Let mapping $f_l \colon P(U) \to P(U)$ denotes a lower approximation and mapping $f_u \colon P(U) \to P(U)$ denotes an upper approximation. Then:

(L1) $\forall X \subseteq U.\ f_l(X) \subseteq X$.
(L2) $\forall X \subseteq U\, \forall x \in f_l(x)\, \exists z \in X.\ x \in R(z) \subseteq X$.
(L3) For each $X \subseteq U$, $f_l(X)$ is definable.
(L4) For each definable $X \subseteq U$, $f_l(X) = X$.
(U1) $\forall X \subseteq U.\ X \subseteq f_u(X)$.
(U2) $\forall X \subseteq U\, \forall x \in f_u(X)\, \exists z \in U.\ x \in R(z) \wedge R(z) \cap X \neq \emptyset$.
(U3) For each $X \subseteq U$, $f_u(X)$ is definable.

The most questionable postulates it seem postulates of definability of approximations (L3 and U3). There are many definitions of approximation mappings that are not definable (see e.g. [3,7]). However, not definable approximations are not interesting from practical point of view (for example in data mining [7]).

Let us observe, that a unit operation is one of the upper approximations considered in the above mentioned papers of Wybraniec-Skardowska, Gomolińska, Słowiński.

Lemma 5. *Let f be a unit operation.*
$X \subseteq f(X)$ for any $X \subseteq U$ if and only the relation R_f is reflexive.

Proof. The proof is straightforward. □

Corollary 3. *$f(X)$ is definable for any unit operation f and $X \subseteq U$.*

By *a definable lower (upper) approximation* we will understand a lower (upper) approximation satisfying postulates (L1)–(L4) (resp. (U1)–(U3)).

A lower approximation is a unit operation only if a relation R is the diagonal relation ($R = \{(x, x): x \in U\}$). We may define a lower approximation for example as a dual to an image relation. In the case R is an equivalence relation, thus defined lower approximation is definable, but in general it may not be a definable approximation.

Corollary 4. *If R is a reflexive relation, then the R-image relation $\overrightarrow{R}$ is a definable upper approximation.*

In the family of all mappings $F = \{f: P(U) \to P(U)\}$ we define the following relation: $f \leq g$ iff $\forall X \subseteq U.\ f(X) \subseteq g(X)$. The relation $\leq$ is a partial order relation, so we may ask for the existence of the least (the greatest, minimal, maximal) mapping.

Theorem 3. *Let R be a reflexive relation on U. Then the R-image relation $\overrightarrow{R}$ is the least definable upper approximation if and only if the relation R is transitive.*

Proof. Let R be a reflexive relation.
($\Rightarrow$) Assume R-image relation $\overrightarrow{R}$ is the least definable upper approximation. Let $(x, y) \in R$ and $(y, w) \in R$. Hence $y \in \overrightarrow{R}(\{x\})$ and $w \in \overrightarrow{R}(\{y\})$. Let $W = \{x, y\}$. By (1), $\overrightarrow{R}(W) = \overrightarrow{R}(\{x\}) \cup \overrightarrow{R}(\{y\})$. By reflexivity of R, we have $x \in \overrightarrow{R}(\{x\})$. Hence $W = \{x, y\} \subseteq \overrightarrow{R}(\{x\})$. Since $\overrightarrow{R}(W)$ is the least definable set containing W, then $\overrightarrow{R}(W) \subseteq \overrightarrow{R}(\{x\})$. Thus $w \in \overrightarrow{R}(\{x\})$. Hence $(x, w) \in R$.
($\Leftarrow$) Let $X \subseteq U$. Of course, $X \subseteq \overrightarrow{R}(X) = \bigcup_{x \in X} \overrightarrow{R}(\{x\})$ by Lemma 5 and the assumption of reflexivity. We show that $\overrightarrow{R}(X)$ is the least definable set containing X. Assume $Z \subseteq U$ is a definable set such that $X \subseteq Z$. It is sufficient to show that $\overrightarrow{R}(X) \subseteq Z$. Let $z \in \overrightarrow{R}(X)$. Hence there exists $x \in X$ such that $z \in \overrightarrow{R}(\{x\})$ (i.e $(x, z) \in R$). Since $x \in X$, then $x \in Z$. By Definition 7, there exists $t \in Y$such that $x \in \overrightarrow{R}(\{t\})$ (i.e. $(t, x) \in R$). By the assumption of transitivity, it follows from this that $(t, z) \in R$ (i.e. $z \in \overrightarrow{R}(\{t\}) \subseteq Z$). □

Example 1. Let us consider an incomplete information table $(U, \mathcal{A})$, where U is a non-empty set of objects and $\mathcal{A}$ is a non-empty set of attributes. Every attribute $a \in \mathcal{A}$ is a function $c: U \to V_a \cup \{*\}$, where V_a is a domain of a set of possible values and "*" is a special symbol. It means an unknown value. We assume, that unknown values of attributes do not allow any comparison (Stefanowski, Tsoukiàs [15]). Stefanowski and Tsoukiàs introduce a similarity relation S as follows:

$$S(x, y) \iff \forall c \in \mathcal{A}.\ (c(x) \neq *) \Rightarrow (c(x) = c(y)).$$

The relation S is reflexive, transitive and not symmetric. The upper approximation defined in [15] is a S-image relation and, by Theorem 3, is the best definable upper approximation. □

Let us notice, that if R is not transitive, then the least definable upper approximation (if exists) is not a unit operation.

Example 2. Let $U = \{1, 2, 3\}$ and $R = \{(1,1), (1,2), (1,3), (2,1), (2,2), (3,3)\}$. The relation R is reflexive.

Let $f: P(U) \to P(U)$ be a mapping such that $f(\{1,2\}) = \{1,2\}$ and $f(X) = \overrightarrow{R}(X)$ for $X \neq \{1,2\}$.

The mapping f is the least definable upper approximation, but it is not a unit operation. □

Example 3. Let $U = \{1, 2, 3, 4\}$ and $R = \{(1,1), (1,2), (1,3), (2,1), (2,2), (2,4), (3,1), (3,3), (4,2), (4,4)\}$. The relation R is reflexive and symmetric.

Let $X = \{1,2\}$. The only definable sets containing X are: $\overrightarrow{R}(\{1,2\}) = \{1,2,3,4\}$, $\overrightarrow{R}(\{1\}) = \{1,2,3\}$ and $\overrightarrow{R}(\{2\}) = \{1,2,4\}$.

In approximation space (U, R) does not exist the least definable upper approximation. □

The above example shows that sometimes it may be impossible to find the least definable upper approximation. This is not a good fact. However, in practise we do not need to approximate all subsets. In decision systems we have several concepts (sets of objects pre-classified by an expert) and we are interested in approximation only these sets. In this case may be helpful to consider unit operation and corresponding binary relation. It is possible to find conditions under which there exist the least definable sets containing these concepts. These conditions are based on the binary relation and the least definable sets are defined using corresponding unit operation.

We can also consider a family of binary relations instead of one relation.

Proposition 8. *Let $\{R_i\}_{i \in I}$ be a family of reflexive and transitive relations on U. The unit operation $\otimes_{i \in I} \overrightarrow{R_i}$ is the least definable upper approximation in the approximation space $(U, \bigcap_{i \in I} R_i)$.*

Proof. It follows easily from Theorem 3 and Proposition 7. □

5 Summary

In this paper we considered unit operations. Some algebraic properties of them are given. We have shown that every unit operation correspond with some binary relation. Next we proved, that if a relation in approximation space is transitive, then there exists the least definable upper approximation and it equals the image relation. It remains an open question, how to define upper approximations in the case of non-transitive relation to be the least definable upper approximations.

References

1. Bonikowski, Z., Bryniarski, E., Wybraniec-Skardowska, U.: Extensions and intentions in the rough set theory. Journal of Information Sciences 107, 149–167 (1998)
2. Bonikowski, Z.: Algebraic Structures of Rough Sets in Representative Approximation Spaces. Electronic Notes in Theoretical Computer Science 82, 1–12 (2003)
3. Gomolińska, A.: A Comparative Study of Some Generalized Rough Approximations. Fundamenta Informaticae 51, 103–119 (2002)
4. Gomolińska, A.: Approximation Spaces Based on Relations of Similarity and Dissimilarity of Objects. Fundamenta Informaticae 79, 319–333 (2007)
5. Grzymała-Busse, J.: Characteristic Relations for Incomplete Data: A Generalization of the Indiscernibility Relation. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets IV. LNCS, vol. 3700, pp. 58–68. Springer, Heidelberg (2005)
6. Grzymała-Busse, J.W.: Incomplete Data and Generalization of Indiscernibility Relation, Definability, and Approximations. In: Ślęzak, D., Wang, G., Szczuka, M.S., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 244–253. Springer, Heidelberg (2005)
7. Grzymała-Busse, J., Rząsa, W.: Definability of Approximations for a Generalization of the Indiscernibility Relation. In: Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence, pp. 65–72 (2007)
8. Jónsson, B., Tarski, A.: Boolean Algebras with Operators. Part I. Amer. J. Math. 73, 891–939 (1951)
9. Kryszkiewicz, M.: Rules in incomplete information systems. Information Sciences 113, 271–292 (1999)
10. Lin, T.Y.: Granular Computing on Binary Relations I. Data Mining and Neighborhood Systems. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery, pp. 107–121. Physica Verlag, Heidelberg (1998)
11. Liu, G., Sai, Y.: A comparison of two types of rough sets induced by coverings. International Journal of Approximate Reasoning 50, 521–528 (2009)
12. Pawlak, Z.: Rough Sets. Intern. J. Comp. Inform. Sci. 11, 341–356 (1982)
13. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
14. Słowiński, R., Vanderpooten, D.: A Generalized Definition of Rough Approximations Based on Similarity. IEEE Transactions on Knowledge and Data Engineering 12(2), 331–336 (2000)
15. Stefanowski, J., Tsoukiàs, A.: Incomplete Information Tables and Rough Classification. Computational Intelligence 17(3), 545–566 (2001)
16. Wybraniec-Skardowska, U.: On a Generalization of Approximation Space. Bull. Polish Acad. Sci. Math. 37(1-6), 51–62 (1989)

17. Wybraniec-Skardowska, U.: Unit Operations. Zeszyty Naukowe WSP w Opolu, Matematyka XXVIII, pp. 113–129 (1992)
18. Yao, Y.Y.: Generalized Rough Set Models. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery, pp. 286–318. Physica Verlag, Heidelberg (1998)
19. Yao, Y.Y.: Relational interpretations of neighborhood operators and rough set approximation operators. Information Sciences 111, 239–259 (1998)
20. Zhu, W.: Generalized rough sets based on relations. Information Sciences 177, 4997–5011 (2007)
21. Zhu, W.: Relationship between generalized rough sets based on binary relation and covering. Information Sciences 179, 210–225 (2009)

Weighted Nearest Neighbor Classification via Maximizing Classification Consistency

Pengfei Zhu, Qinghua Hu, and Yongbin Yang

Harbin Institute of Technology, Harbin 150001, China
huqinghua@hit.edu.cn

Abstract. The nearest neighbor classification is a simple and effective technique for pattern recognition. The performance of this technique is known to be sensitive to the distance function used in classifying a test instance. In this paper, we propose a technique to learn sample weights via maximizing classification consistency. Experimental analysis shows that the distance trained in this way enlarges the classification consistency on several datasets and has a strong ability to tolerate noise. Moreover, the proposed approach has better performance than nearest neighbor classification and several state-of-the-art methods.

1 Introduction

The Nearest-Neighbor rule is among the most popular and successful pattern classification techniques. The NN classifier can be represented by the following simple rule: the label of an unknown pattern is identified by choosing the class of the nearest stored training instance [2].

The performance of 1-NN classification is influenced by several factors, including the distance metric used to find the NN of a query pattern, the curse of dimension and so on. There have been consistent efforts devoted to improving the performance of 1-NN classification. In the last few years, many methods have been developed to locally adapt the distance metric [7,8,9,10] or prototype editing. In [4], a locally adaptive distance measure was used based on assigning a weight to each training instance. In [6], a prototype weighting algorithms was derived by approximately maximizing the Leaving-One-Out classification error of the given training set. In [5], a method for learning a Mahalanobis distance metric from training samples by semidefinite programming was introduced. These methods have been proved to be efficient on some real-world datasets.

Similar to [4], the weighted metric we use in this paper is based on assigning a weight to each instance in the training set. The technique we propose can learn a sample weight vector via maximizing classification consistency, which is defined as the average memberships to the fuzzy lower approximations in fuzzy rough sets. Lower approximations are introduced in rough sets and considered as the subset of samples which can be grouped into one of the decision classes without doubt [1,11,12,15] . Then, fuzzy lower approximations are proposed to deal with numerical or fuzzy features. In this paper, classification consistency is defined

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 347–355, 2010.

as the average memberships to the fuzzy lower approximations and reflects the percentage of fuzzy consistent objects over the whole universe. In classification learning, one naturally expects to get a highly consistent classification. However, there are usually some inconsistent samples due to the existence of noisy or insufficient information. As to the inconsistent samples, we can expect to get a highly consistent classification via maximizing fuzzy lower approximation.

In this work, a sample weighted matrix is learned to maximize classification consistency. The proposed method, Sample Weight Learning via Maximizing classification consistency (SWL-MCC), assigns different weights to different samples according to their positions in the feature spaces. SWL-MCC can enlarge the classification consistency by assigning greater weights to the boundary samples. Besides, we also show that the proposed technique has a strong ability to tolerate noisy samples, whose memberships to their lower approximation are relatively small. Experimental analysis shows that the metric trained with SWL-MCC enlarges the classification consistency on several benchmark data sets, and the proposed approach has better performance than the nearest neighbor rule and A-NN [4].

The rest of the paper is organized as follows. Section 2 introduces the basic concept of classification consistency. In Section 3, a sample weighted distance learning algorithm is proposed via maximizing classification consistency. Section 4 presents experimental results on some artificial and real-world data sets. Finally, conclusions are given in Section 5.

2 Classification Consistency

Given a nonempty and finite set U of objects, R is a fuzzy equivalence relation on U. For $\forall x \in U$, we associate a fuzzy equivalence class $[x]_R$ with x. The membership function of y to is defined as $[x]_R(y) = R(x, y), \forall y \in U$. The family of fuzzy equivalence classes forms a set of fuzzy elemental granules for approximating arbitrary subset of the universe. Given a fuzzy subset $X \in F(U)$, the lower approximation and upper approximation of X with respect to R were defined as

$$\underline{R_{\max}}X(x) = \inf_{y \in U} \max(1 - R(x, y), X(y)),\ \overline{R_{\min}}X(x) = \sup_{y \in U} \min(R(x, y), X(y))$$

Given a $T-$equivalence relation and a residual implication θ induced with T, the fuzzy lower and fuzzy upper approximations of fuzzy subset X were defined as

$$\underline{R_\theta}X(x) = \inf_{y \in U} \theta(R(x, y), X(y)),\ \overline{R_T}X(x) = \sup_{y \in U} T(R(x, y), X(y)).$$

In this context, the membership of a sample to its fuzzy lower approximation is the minimal distance to other classes. In fact, fuzzy approximation is a fuzzy set and the memberships reflect the consistence degree of classification. Hence, we can define classification consistency as the average memberships to the fuzzy lower approximations.

Definition 1. *Let $X \subseteq U$is a fuzzy subset, the cardinality of X is defined as $|X| = \sum_{x \in U} X(x)$, where $X(x)$is the membership of x to X.*

Definition 2. *Given a classification learning problem, k is $T-$equivalence relation on U computed with kernel function $k(x, y)$ in the feature space $B \subseteq A$. U is divided into $\{d_1, d_2, \cdots, d_N\}$ with the decision attribute. The fuzzy positive regions of D in term of B are defined as*

$$POS_B(D) = \bigcup_{i=1}^{N} \underline{k}d_i, \tag{1}$$

where $N$is the number of classes.

Definition 3. *Given a classification learning problem, k is $T-$equivalence relation on U computed with Gaussian function $k(x, y)$ in feature space $B \subseteq A$. U is divided into $d_1, d_2, \cdots, d_N$ with the decision attribute. The classification consistency of D on B is defined as*

$$C_{\mathbf{B}}(D) = \frac{|\bigcup_{i=1}^{N} \underline{k}d_i|}{|U|}. \tag{2}$$

As $\underline{k_S}d_i(x) = \inf_{y \notin d_i} (1 - k(x, y))$, we get that $|\bigcup_{i=1}^{N} \underline{k}d_i| = \sum_{i=1}^{n} \sum_{j}^{N} \underline{k}d_j(x_i)$. Furthermore, we also know that $\underline{k}d_j(x_i) = 0$ if $x_i \notin d_j$. Therefore $|\bigcup_{i=1}^{N} \underline{k}d_i| = \sum_{i=1}^{n} \underline{k}d(x_i) = \sum_{i=1}^{n} \inf_{x_i \in d, y \notin d} (1 - k(x_i, y))$, where is the class label d of x_i.

In essence, classification consistency is the average distance of each sample to other classes or the inter-class distance in kernel space. One expects the distance is large enough to discriminate different classes. Maximizing classification consistency means maximizing the inter-class distance.

3 Sample Weight Learning

Faced with the inconsistent samples, we can expect to get a highly consistent classification via maximizing fuzzy lower approximation.

Assumed Gaussian function $k = \exp(-\frac{||x-y||^2}{\sigma})$is used to compute the fuzzy similarity relation between samples, and then we approximate the decision subsets with the fuzzy granules induced by k. The classification consistency can be defined as:

$$C_{\mathbf{B}}(D) = \frac{1}{n} \sum_{i=1}^{n} 1 - \exp\left(-\frac{\|x_i - NM(x_i)\|^2}{\sigma}\right) \tag{3}$$

where $NM(x_i)$, called nearest miss in [3]denotes the nearest sample of x_i in classes different from x_i.

Usually, the weights of different samples are set as a uniform value, that is, $w_j = 1$. However, it is well known that the importance of candidate samples varies according to their position in the sample space. Thus an algorithm is developed to optimize the weight vector $W =< w_1, w_2, \cdots, w_j, \cdots, w_n >$. The optimization objective function is

$$C_{\mathbf{B}}{}^{w}(D) = \frac{1}{n}\sum_{i=1}^{n} 1 - \exp\left\{-\frac{w(NM(x_i))^2\|x - NM(x_i)\|^2}{\sigma}\right\} \tag{4}$$

Similar to [6], we can use gradient descent to maximize classification consistency. The maximization of $C_{\mathbf{B}}{}^{w}(D)$ by gradient descent consists in an iterative procedure which updates the weights $w(i)$ by a small amount, in the negative direction of the gradient of $C_{\mathbf{B}}{}^{w}(D)$:

$$w(i) = w(i) - \eta\frac{\partial C_{\mathbf{B}}{}^{w}(D)}{\partial w(i)} \tag{5}$$

The update equation is:

$$w(i) = w(i) + \eta\sum_{x\in s}\left\{\begin{array}{l} 2\times\|x - NM(x)\|^2 \times w(NM(x))\times \\ \exp\left[-w(NM(x))^2 \times \|x - NM(x)\|^2/\delta\right]\end{array}\right\}/\delta$$

Given a set of training samples, we can iteratively search the weight with the following procedure.

{`Sample weight learning via maximizing classification consistency (SWL-MCC)`}
`Procedure` {`Initialize`}
$w =< 1, 1, ..., 1 >, CC = 0, CC1 = 1, \varepsilon > 0.001$
$\forall x \in U$, `compute` $NM(x)$ and $NH(x)$, $\delta = \frac{1}{n}\sum_{x\in U}(\|x - \mathrm{NM}(x)\| - \|x - NH(x)\|)$
`While` $|CC1 - CC| > \varepsilon$
  $CC1 = CC$
  `For` $i = 1, 2, \cdots, n,$
    $w(NM(x)) = w(NM(x)) +$
    $\left\{\begin{array}{l} 2\times\|x - NM(x)\|^2 \times w(NM(x))\times \\ \exp\left[-w(NM(x))\times\|x - NM(x)\|\right]^2/\delta)\end{array}\right\}/\delta$
  `EndFor`
  Compute the classification consistency after samples are weighted
`EndWhile`
`EndProcedure`

The values of η are referred to as learning rates or learning step factors. It can be a constant for all samples or may vary on different samples or in each step. In this context, η is set as a positive value 0.1. The Parameter σ used in the Gaussian function which is related to the computation of classification consistency, may affect the learning performance of the proposed technique. The membership of fuzzy lower approximation may arrive at its maximal value 1 and

its minimal value 0 if a very little positive constant and a very great positive constant are assigned to respectively. Therefore, it is difficult to be optimized. Thus, σ should be set as an appropriate value.

$$\underline{k_S}d_i(x) = \inf_{y \notin d_i} \left(1 - \exp(-||x-y||^2/\sigma)\right)$$

$$if \sigma \to \infty, \underline{k_S}d_i(x) \to 0, \mathrm{i} = 1, 2, ..., \mathrm{n}, C_{\mathbf{B}}(D) = \frac{|\bigcup_{i=1}^{N} \underline{k}d_i|}{|U|} \to 0$$

$$if \sigma \to 0, \underline{k_S}d_i(x) \to 1, \mathrm{i} = 1, 2, ..., \mathrm{n}, C_{\mathbf{B}}(D) = \frac{|\bigcup_{i=1}^{N} \underline{k}d_i|}{|U|} \to 1$$

As a matter of fact, classification consistency can be understood as the average distance of samples to the nearest samples from other classes. If we expect that classification consistency can vary in an appropriate interval for any task by setting the value of parameter σ, the value of σ should be set according to the average distance of samples to the nearest samples from other classes, that is, $\sigma = \frac{1}{n} \sum_{x \in U} ||x - NM(x)||$.

Essentially, SWL-MCC mainly assigns greater weights to the boundary samples whose membership to its lower approximation is very small and does not change weights of non-boundary samples. In fact, the smaller the membership to its lower approximation is, the closer a sample is to the classification boundary, and the positive gradients $\Delta(w(NM(x)))$ of more heterogeneous samples are added to the weight of this sample. Meanwhile, the proposed algorithm performs quite well in the presence of noisy samples. In Fig.1, there is a noisy sample O in the center and it is the nearest miss of the samples A, B, C and D, who would be misclassified when all the samples get equal importance. When we operate SWL-MCC on this data, the weight of the sample A would be greatly enlarged because the positive gradients of the samples A, B, C and D are added to $w(O)$. The small circle and large circle denotes the distance between the sample A and

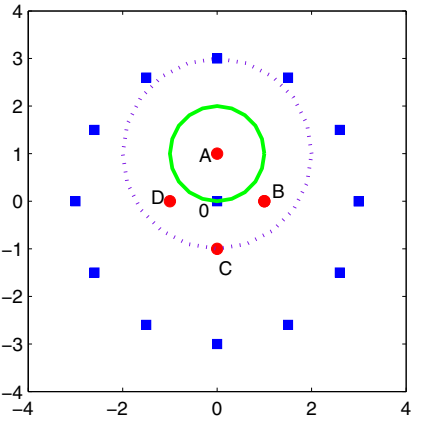


Fig. 1. A toy example with noisy samples

its nearest miss before and after samples are weighted. We can find that the membership of the sample A to its lower approximation increases significantly and can be correctly classified after samples are weighted. Hence, SWL-MCC can automatically indentify the noisy samples and assign greater weights to them.

The computational complexity of SWL-MCC is $O(n^2m)$, where m is the number of features and n is the size of the sample U.

4 Experimental Analysis

In this section, the capability of the sample-weighted technique has been empirically assessed through experiments on two artificial datasets and some real-world datasets.

4.1 Synthetic Dataset

We generate two sets of 26 sample points satisfying Gaussian distribution in a 2-dimensional real space, as shown in Fig. 2. It contains only two output classes in order to make the graphing and visualization easier. The weights of the twenty-six samples are obtained with the proposed technique operated on the synthetic dataset and labeled in Fig.2. In addition, the samples A, B $\cdots$ I, whose weights are greater than one are listed in Tab.1. Meanwhile, we calculate the membership to its lower approximation as to each instance with samples treated as equal, and

Table 1. Memberships and weights of samples whose weights are greater than 1

samples	A	B	C	D	E	F	G	H	I
memberships	0.006	0.006	0.016	0.016	0.0981	0.251	0.266	0.337	0.337
weights	4.211	5.047	2.549	4.151	4.000	4.976	4.243	5.433	1.082

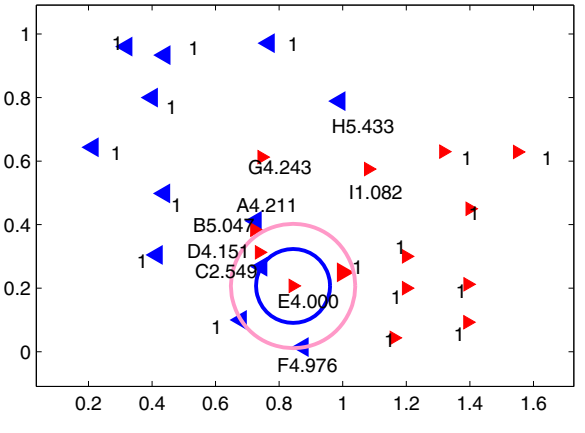


Fig. 2. A toy example of binary classification task

sort the samples in ascending order of the membership. We find that the first nine of the twenty-five samples are the samples A, B $\cdots$ I. As we know, as to the sample, the membership to its lower approximation can be understood as the distance between and its nearest miss . Hence, we can conclude that the proposed technique can assign greater weights to the samples on or close to the boundary, whose memberships to their fuzzy approximation is very small, and do not change the weights of the non-boundary samples. By assigning different weights to different samples according to the position of the samples in the feature space, the fuzzy lower approximation and classification consistency can be greatly enlarged. As shown in Fig.2, the fuzzy lower approximation of the sample E before and after samples are weighted are denoted by the blue circle and red circle respectively. It is obvious that the sample E, whose membership to its lower approximation is enlarged, can be correctly classified after samples are weighted.

4.2 Effect of Noise

The aim of this experiment is to evaluate the performance of the proposed methods in the presence of noise. This is done by randomly changing the class labels of the instances in the training set to an incorrect value (with equal probability for each of the incorrect classes) at different levels, which represents the percentage of the changed instances. We test the ability of SWL-MCC to tolerate the noise on a dataset WPBC (198 samples and 34 features) at different noise levels. From Table2 we can see that the classification consistency of SWL-MCC is much higher than the original nearest neighbor classifier and A-1-NN at all levels. The classification consistency is improved by 6.5% and 4.2% compared to 1-NN and A-1-NN respectively. Overall, the proposed technique via maximizing classification accuracy has a strong ability to tolerate the noise.

Table 2. Classification consistency of SWL-MCC compared with other methods on the noisy dataset

Noise level	1-NN	SWL-MCC	A-1-NN
3%	70.8±8.5	76.3±6.5	71.8±12.3
5%	70.3±7.7	75.3±4.6	71.7±8.8
7%	70.2±6.3	75.2±1.8	72.2±7.8
9%	63.2±7.4	70.0±5.3	65.3±7.9
11%	59.1±12.4	69.2±5.6	64.2±10.6
Average	66.7	73.2	69.0

4.3 Real World Problems

We gathered twelve datasets from UCI machine learning repository [16]. The classification consistency before and after samples are weighted is listed in Table 3 (CC and WCC denotes classification consistency and weighted classification consistency respectively; S/F/C denotes the numbers of the samples,

Table 3. Variance of classification consistency and classification accuracy of SWL-MCC compared to other methods

Data	S/F/C	CC	WCC	1-NN	SWL-MCC	A-1-NN
WPBC	198/33/2	0.5591	0.8118	70.6±6.8	75.3±5.9	72.7±10.3
german	1000/21/2	0.5855	0.7602	68.8±3.2	71.1±3.7	71.3±2.8
Crx	690/16/2	0.4300	0.7200	79.1±11.6	83.3±15.9	80.7±11.7
heart	270/13/2	0.5388	0.8284	76.6±9.4	81.5±5.5	77.0±5.5
hepatitis	155/19/2	0.6001	0.8152	82.5±7.6	86.0±8.0	83.2±9.4
hors	368/22/2	0.6069	0.7980	87.2±4.2	90.4±4.4	88.1±3.1
iono	351/34/2	0.4911	0.6220	86.4±4.9	90.5±5.6	90.1±4.1
WDBC	569/30/2	0.4959	0.6392	95.4±3.3	97.4±2.2	96.5±2.5
Breast	84/9217/5	0.6082	0.6543	77.0±18.3	85.0±14.2	82.0+13.3
derm	366/35/6	0.5614	0.6201	96.1±5.7	97.9±2.9	96.3±0.5
iris	150/5/3	0.4928	0.5840	96.0±5.6	97.3±4.6	96.0±5.6
Gene5	72/7130/3	0.6135	0.6495	78.9±17.2	82.9±15.0	87.1±14.8
Average	/	/	/	82.8	86.6	85.0

features and classes respectively). From Table 3, we can see that the classification consistency of all data sets has been greatly enlarged after samples are weighted.

In order to evaluate the performance of the proposed technique on the classification, we compare the SWL-MCC to original nearest neighbor classifier and A-1-NN on the thirteen datasets, as shown in Table 3 We can see that average classification accuracy of SWL-MCC is higher than 1-NN and A-1-NN by 3.8% and 1.6% respectively and the proposed technique outperforms the other two methods on almost all the datasets.

5 Conclusion

The performance of the nearest neighbor (NN) classification depends significantly on the distance functions used to compute similarity between examples. In this paper, we propose a technique that can assign different weights to each instance via maximizing the classification consistency. By automatically indentifying the noisy samples and assigning greater weights to the noisy samples, the proposed technique is insensitive to noise. In comparison with 1-NN and A-1-NN, we show that SWL-MCC achieves better accuracy results. We also show that SWL-MCC performs quite well in the presence of noisy data.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grants 60703013 and 10978011.

References

1. Yao, Y., Zhao, Y.: Attribute reduction in decision-theoretic rough set models. Information Sciences 78(17), 3356–3373 (2008)
2. Hart, P., Cover, T.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory 13, 21–27 (1967)
3. Gilad-Bachrach, R., Navot, A., Tishby, N.: Margin based feature selection - theory and algorithms. In: ICML 2004 (2004)
4. Wang, J., Neskovic, P., Cooper, L.N.: Improving nearest neighbor rule with a simple adaptive distance measure. Pattern Recognition Letters 28, 207–213 (2007)
5. Weinberger, K., Blitzer, J., Saul, L.: Distance metric learning for large margin nearest neighbor classification. In: Advances in Neural Information Processing Systems (NIPS), vol. 18
6. Paredes, R., Vidal, E.: Learning weighted metrics to minimize nearest-neighbor classification error. IEEE Trans. Pattern Anal. Mach. Intell. 28, 1100–1114 (2006)
7. Hastie, T., Tibshirani, R.: Discriminant Adaptive Nearest Neighbor Classification and Regression. In: Advances in Neural Information Processing Systems, vol. 8, pp. 409–415 (1996)
8. Howe, N., Cardie, C.: Examining Locally Varying Weights for Nearest Neighbor Algorithms. In: Leake, D.B., Plaza, E. (eds.) ICCBR 1997. LNCS, vol. 1266, pp. 455–466. Springer, Heidelberg (1997)
9. Kohavi, R., Langley, P., Yung, Y.: The Utility of Feature Weighting in Nearest-Neighbor Algorithms. In: van Someren, M., Widmer, G. (eds.) ECML 1997. LNCS, vol. 1224, pp. 455–466. Springer, Heidelberg (1997)
10. Wilson, D.: Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. IEEE Trans. Systems, Man, and Cybernetics 2, 408–421 (1972)
11. Hu, Q.H., Xie, Z.X., Yu, D.R.: Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation. Pattern Recognition 40(12), 3509–3521 (2007)
12. Hu, X., Cercone, N.: Data mining via discretization, generalization and rough set feature selection. Knowledge and Information Systems 1(1), 33–60 (1999)
13. Morsi, N.N., Yakout, M.M.: Axiomatics for fuzzy rough set. Fuzzy Sets System 100, 327–342 (1998)
14. Perou, C.M., Srlie, T., Eisen, M.B., et al.: Molecular portraits of human breast tumours. Nature 406, 747–752 (2000)
15. Slezak, D.: Degrees of conditional (in)dependence: A framework for approximate Bayesian networks and examples related to the rough set-based feature selection. Information Sciences 1789(3), 197–209 (2009)
16. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html

Rough Set-Based Incremental Learning Approach to Face Recognition

Xuguang Chen and Wojciech Ziarko

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

Abstract. The article reports our implementation of a rough set-based incremental learning algorithm involving the application of the hierarchy of probabilistic decision tables to face recognition. The implementation, the related theoretical background such as the basics of the variable precision rough set theory, the algorithm, the classifier structure and experiments with balanced and imbalanced data sets are presented.

1 Introduction

Face recognition has received wide attention from researchers applying diverse methodologies to this problem. Face recognition methods can be classified into holistic matching methods and feature-based matching methods [4] [12]. They however share a common limitation: when constructing a face recognition system, it is often hard to account for all variations of face photos of each participant. Face of the same person can change greatly depending on expression, illumination conditions, or presence or absence of make-up [5]. Therefore, it is desirable to construct a recognition system that could improve its performance continuously by learning additional information via process of *incremental learning* from the newly added training objects (facial photos). Incremental mode of learning would be particularly advantageous since it involves gradual modification of the learned data structure [10].

Rough set theory was proposed by Pawlak [1] and extended, among others, by Ziarko [2]. It can be used to determine the most appropriate attributes for a given information system [1]. The theory has been applied to many areas, including face recognition. In [7], a face representation and rough set-based recognition methodology, called *soft-cut and probabilistic distance-based classifier* (*soft-cut classifier* for short), are described. In this paper, we discuss how the extended rough set-based approach can be applied to incremental learning of hierarchical structures of probabilistic decision tables in the context of face recognition application.

The article is organized as follows. In Section 2, the development of a hierarchy of learnt decision tables, based on accumulated training data (pictures of faces), involving variable precision rough set theory [2] and the incremental learning algorithm methodology are presented. Section 3 describes how to apply soft-cut classifier approach [7] to incremental learning [10]. Section 4 presents the experimental results, and a brief summary is included in Section 5.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 356–365, 2010.

2 Adaptive Decision Table-Based Approach

The rough set theory was introduced by Pawlak [1], and the variable precision model (VPRSM) of rough sets broadens its the basic ideas. In the VPRSM, two model precision-control parameters are used, denoted as the *lower limit* l, and the *upper limit* u [2], respectively, such that $0 \leq l < P(X) < u \leq 1$. $P(X)$ is a *prior probability* of the target set X.

Let E denote an elementary set of an approximation space on the universe U [1] and let $P(X|E)$ denote a conditional probability of the set X occurrence given that the set E occurred. In the VPRSM, the negative region is defined as $NEG_l(X) = \cup\{E \subseteq U : P(X|E) \leq l\}$. Objects are classified into the negative region of the set X if the probability of the membership in the set X is significantly lower, as expressed by the *lower limit* l, than the prior probability $P(X)$. The positive region is defined as $POS_u(X) = \cup\{E \subseteq U : P(X|E) \geq u\}$. Objects are classified into the positive region of the set X if the probability of the membership in the set X is significantly higher, as expressed by the *upper limit* u, than the prior probability $P(X)$. The objects that are not classified into either the positive region nor the negative region are classified into the boundary region of the decision category X. The boundary region is defined as $BND_{l,u}(X) = \cup\{E \subseteq U : l < P(X|E) < u\}$.

The probabilistic decision tables and their hierarchies extend the notion of decision table acquired from data as introduced by Pawlak [1]. The probabilistic decision table approximately represents the stochastic relation between condition and decision attributes via a set of uniform size probabilistic rules. That is, the probabilistic decision table is a mapping that assigns each vector of condition attribute values, corresponding to an elementary set E, to its unique designation of one of VPRSM approximation regions $POS_u(X)$, $NEG_l(X)$ or $BND_{l,u}(X)$, along with associated elementary set E probabilities $P(E)$ and conditional probabilities $P(X|E)$. They can be conveniently represented in a tabular form.

In the VPRSM, the boundary region $BND_{l,u}(X)$ is a definable subset of the universe U, which can be precisely specified by its elementary sets. The basic idea behind the hierarchies of probabilistic decision table construction is to treat the boundary region as an independent sub-universe of the universe U. Such a sub-universe can have its local collection of condition attributes to form a new approximation sub-space, from which a "child" decision table can be derived [4]. By repeating the step of parent-child decision table formation recursively, until either the boundary region is eliminated or some other attribute-based termination criteria are satisfied, a hierarchy of probabilistic decision table can be constructed [3].

For the purpose of classifier evaluation, an inter-attribute probabilistic dependency measure, called the $\lambda - dependency$ was adopted in our experiments [8]. The $\lambda - dependency$ is defined as the normalized expected degree of deviation of the conditional probability of the target set X, $P(X|E)$, from its prior probability $P(X)$, that is $\lambda(X|C) = \frac{\sum_{E \in U/C} P(E)|P(X|E) - P(X \cap U)|}{2P(X)(1-P(X))}$, where U/C is a collection of elementary sets induced by the set of condition attributes C and $2P(X)(1 - P(X))$ is a normalization factor.

2.1 Hierarchy Adaptation Strategies

The generated hierarchy of decision tables is a subject of growth and adaptive modification with the arrival of new training objects. In [3], an incremental algorithm satisfying the properties specified in [6] was proposed. Its basic ideas, as as summarized below, have been adopted in our classifier system.

Given a new image I_{new}, it is said to be *inconsistent* if I_{new} can match the pattern of condition attributes values of an elementary set belonging to the positive region but cannot match the value of the decision attribute. Otherwise, I_{new} is *consistent*. For the *j-th* layer of the hierarchy of decision tables DT_j having n elementary sets, its universe will be denoted by U_j; the *POS, BND*, and *NEG* regions on layer j will be denoted as POS_j, BND_j, and NEG_j; the $\lambda - dependency$ of the partition of the universe U hierarchy is λ_{total}; the elementary set i on the level j is denoted by E_{ij}, where *i=1...n*. Depending on the approximation region of the *j-th* layer of the hierarchy decision tables a new case I_{new} would fall, the following adaptation strategies are applied when adjusting the hierarchy structure and recomputing λ_{total}:

1. if $I_{new} \in E_{ij} \subset BND_j \subset DT_j$, then $|E_{ij}| = |E_{ij}| + 1$, $|U_j| = |U_j| + 1$;
2. if a consistent $I_{new} \in E_{ij} \subset POS_j \subset DT_j$, then $|E_{ij}| = |E_{ij}| + 1$, $|U_j| = |U_j| + 1$;
3. if an inconsistent $I_{new} \in E_{ij} \subset POS_j \subset DT_j$, then $E_{ij} = E_{ij} \cup I_{new}$ and $BND_j = BND_j \cup E_{ij}$; $U_{j+1} = U_j \cup E_{ij}$ and then recursively apply the adaptation rules starting from the layer DT_{j+1} followed by re-computation of all affected subordinate layers;
4. if $I_{new} \notin E_{ij}$, where *i=1...n*, then create a new elementary set $E_{(n+1)j} = \{I_{new}\} \subset POS_j$, $|U_j| = |U_j| + 1$, $DT_j = DT_j \cup E_{(n+1)j}$.

The next section details how these adaptation strategies are used to control the growth of the decision table hierarchies.

3 Soft-Cut Classifier and Incremental Learning

Before constructing probabilistic decision tables of the hierarchy, photos in the training set need to be pre-processed for feature acquisition. The details of this step are described in [7]. In our method, due to its simplicity and small computation costs, resulting from the sparsity of the transform matrices and the small number of significant wavelet coefficients [11], we first transform each photo I_i by Haar-wavelet transformation and represent each I_i by a small group of m selected Haar-wavelet coefficients from a particular part of a certain level of the Haar-wavelet transformation, denoted as $x^i_{Haar,m}$. All of N photos in the training set form an $N \times m$ pattern matrix X, which are then processed by principal component analysis (PCA). The result is that each photo is represented by an r-dimensional ($r \leq n$) PCA feature patterns $x^i_{pca,r}$. After that, each $x^i_{pca,r}$ is firstly transformed by a sigmoid function and then discretized from real-valued into binary-valued. More details of this step are provided in [7].

To avoid elementary sets with spurious attributes, if one or more features of the feature vector $x^i_{pca,r}$ could not be discretized due to its too close proximity to the selected cut point, the photo was automatically classified into $BND_{l,u}(X)$. Such a photo was then considered again when working on the $BND_{l,u}(X)$ to build the classification of the next layer of decision table in the hierarchy. After the discretization was completed, only those photos, the selected principal components of which have been completely discretized, were retained. The retained photos were represented by a $p \times Q$ pattern matrix denoted as X_{discr}, where Q is the number of photos with p discretized attributes.

After X_{discr} was formed, it was evaluated by using λ-dependency measure in a hill-climbing fashion to find the most adequate combination of attributes (columns of the matrix) for the recognition task on each layer of the hierarchy. The selected attributes were represented by the matrix X_{simp} used to construct the decision table for the current layer in the hierarchy according to the steps below:

1. **Select the first column of X_{simp} from X_{discr}.**
   First, we define the local λ-dependency restricted to a specific level of the hierarchy: $\lambda_{loc}(X|C) = \frac{\sum_{E \in U/C} P(E)|P(X|E) - P(X \cap BND_{l,u}(X))|}{2P(X \cap BND_{l,u}(X))(1 - P(X \cap BND_{l,u}(X)))}$.
   The local λ-dependency is used to select the column from X_{discr} that generates the highest value of λ_{loc}. The selected first column is then moved from X_{discr} to X_{simp}. The matrix X_{discr} becomes a $(p-1) \times Q$ pattern matrix.
2. **Select the remaining columns from X_{discr}.** Each time, one column of X_{discr} and all of columns of X_{simp} are combined to construct a trial decision table of the current layer. Its λ_{loc} is computed. The column of X_{discr} generating the highest λ_{loc} is then permanently moved from X_{discr} to X_{simp}.
3. **Set the maximum number of k columns in X_{simp}.**
   We heuristically select k=*6* as the threshold value: once a X_{simp} having k columns is reached, we immediately stop. Based on our experiments, a X_{simp} having k=*2* or k=*3* columns is sufficient to generate an appropriate decision table.

Following selection of proper columns of X_{simp}, the probabilistic decision table was constructed for the given layer of the decision table hierarchy. The photos were classified into elementary sets that were assigned to rough approximation regions $POS_u(X)$, $NEG_l(X)$, or $BND_{l,u}(X)$ based on the region definitions. Starting from the top decision table, the above process was recursively repeated for photos classified into the boundary area to build next layers of probabilistic decision tables until all photos were classified into either $POS_u(X)$ or $NEG_l(X)$, or all of Haar-wavelet coefficients have been utilized. The end-result was a hierarchy of probabilistic decision tables. With photos of N persons, we would build N initial hierarchies. For each hierarchy, photos from one specific person would correspond to the recognition target, the set X.

3.1 Incremental Update of Probabilistic Decision Tables

When a new case (a photo) I_{new} becomes available, it is added to the training set, and the corresponding hierarchies of probabilistic decision tables are updated.

Before being added to the training set, the new case I_{new} must be pre-processed in the same way as other photos already in the training set. That is, it is to be transformed by the Haar-wavelet transformation and then some of its wavelets coefficients are selected for further processing. The wavelet coefficients must be selected in the same way as from the photos in the training set. For example, when building the first decision table that has four condition attributes in a hierarchy, if the four selected coefficients are from the vertical part of level 2 Haar-wavelet transformation and their indexes are No.1, No.3, No.4, and No.6, I_{new} must also be represented by the coefficients having the same indexes and from the same area of Haar-wavelet transform.

Next, these selected wavelet coefficients of I_{new} are transformed by PCA, using features such as average image, eigenvectors etc. that are generated when processing the photos for the training set to build the initial hierarchies. Finally, these selected PCA-based features of the new photo are subject to transformation via the same sigmoid function as applied to the photos in the initial training set.

To associate a new photo with an elementary set, starting with the top decision table in the hierarchy, Euclidean distance $EDis_i$ between the new photo I_{new} and each elementary set E_i of the first decision table of the hierarchy is calculated based on the formula (the details are provided in [7]): $EDis_i = \|I_{new} - E_i\|$, where I_{new} is represented by the selected PCA-based features as described in the previous section, and E_i is represented by the vector of its binary condition attribute values. The minimum value of the distance among all elementary sets is denoted as $EDis_{min}$. We require that $EDis_{min}$ be less than a pre-defined threshold value τ, the value of which can be identified heuristically. The threshold value is introduced to prevent classification of "distant" new case into an existing elementary set. That is, if $EDis_i = EDis_{min} < \tau$, for the elementary set E_i, then I_{new} is classified into the elementary set E_i. If $EDis_{min} \geq \tau$, I_{new} will be discretized by crisp cut function into a binary attribute-value vector to form a new elementary set. In this case, for each PCA-based feature x_i^{pca}, its value is compared to arithmetic average C_i of that selected PCA-based feature of all photos in the initial training set. If $x_i^{pca} > C_i$, then 1 is assigned as the corresponding attribute value; otherwise 0 is assigned.

After classifying the new photo into an appropriate elementary set, or creating a new elementary set to contain the new photo on the top level of the hierarchy, we need to recursively modify the subordinate levels of hierarchy of the decision tables to propagate the change, if the introduction of the new case affects the boundary area on the given level. For that purpose, the adaptation strategies listed in previous section were employed, while taking into consideration the following factors:

1. The condition attribute values of the new photo I_{new} and the condition attribute values of the elementary set E_k absorbing the new photo;
2. The region of an elementary set is located in;
3. The λ-dependency of the hierarchy after the decision tables of the hierarchy have been modified.

Factor 1 and factor 2 are primarily used to determine how to modify each lower layer of the decision table in the hierarchy, starting from the top table. For example, if the condition attribute values of I_{new} is the same as the condition attribute values of the elementary set E_k located in the positive area of the top decision table of the hierarchy, then I_{new} is consistent and the second strategy is applied. The change propagation is terminated. The cardinality of E_k and the cardinality of the universe should be modified before recalculating λ-dependency of the hierarchy. Similarly, if the condition attribute values of I_{new} is not the same as the condition attribute values of the elementary set E_k located in the positive area of the first decision table of the hierarchy, then I_{new} is inconsistent and in this case the third strategy is applied, resulting in the creation of a new elementary set and termination of change propagation.

In practical applications, the hierarchy of decision tables is usually learned from photos in the initial training set. That is to say, for the photos in the the initial training set, the condition attributes of each decision table are deemed to be the most appropriate features, as selected by λ-dependency-based criterion. After a new photo was added, it is possible that some of the condition attributes of some of the decision tables in the hierarchy would cease to be the most appropriate features. Therefore, after the hierarchy has been updated, we still need to evaluate the attribute structure of the updated hierarchy by the λ-dependency criterion [3] to ensure that its structure is still stable.

If the value of the new λ-dependency λ_{new} is much lower than that of before adding the new photo, it indicates that the attribute structure of the hierarchy is no longer appropriate, and we have to completely re-generate the hierarchy. The re-generation can be accomplished by employing all of the steps described earlier, with the training set expanded by the newly added photos. On the other hand, if the value of λ_{new} is not much lower than the initial value, we can keep the updated hierarchy and update it again when another new photo is available. The steps described above can be continued until no new photos are available for the purpose of training.

3.2 Recognition Process

Once the process of incremental learning is finished, the hierarchy can be tested by the method called *probabilistic distance-based classification method* [7]. With this method, each test photo I_{test} is first pre-processed by Haar-wavelet transformation and PCA. Then, starting from the top decision table of the first hierarchy (corresponding to the first recognition target X), we evaluate the distance between each elementary set and I_{test}, to match I_{test} with the nearest elementary set E_{min}, subject to a pre-defined threshold value τ limitation.

The outcome of the matching process is either the conditional probability $P(X|E_{min})$, if the test case was matched by the elementary set E_{min}, or the prior probability $P(X)$ of the decision category X, if no matching elementary set was identified on any level of the decision table hierarchy. The matching is repeated for each hierarchy of decision tables, that is for each recognition target X, producing a ranking of probability values for different recognition targets.

The highest ranked recognition target is then selected as the recognition result of the classifier.

4 Experimental Results

In this Section, some recognition learning experiments are described. Photos were collected from 16 students and staff (8 men and 8 women, 96 photos/per person) in the Science Faculty, University of Regina [7]. These photos were taken at three office type locations with different facial expressions (neutral, smile, anger, and scream). The background of the photos was always plain and white with similar but uncontrolled lighting. The purpose was to observe how the hierarchy of decision tables would vary during the process of incremental learning, to evaluate the performance of the soft-cut classifier with balanced and unbalanced datasets, and to identify the impact of the factors such as facial expression, gender ratio, and number of participants in the training set on the recognition accuracy.

4.1 Experiments with Balanced Data Sets

According to [9], we define a training set as balanced if the number of photos of each participant is the same; the total number of photos of the male participants and the female participants is the same; and the number of photos with different facial expressions is the same. Based on the number of participants in the data sets, experiments described in this section can be classified into two categories: *4-person sets* and *sets having photos of more than 4 participants*.

There were 784 4-person experimental data sets constructed initially, and then each experimental set was divided into a training set (24 photos/per person), an add-on set (24 photos/per person), and a test set (24 photos/per person). Each time, 4 initial hierarchies were built based on one training set, and then when updating them, photos of four different people were selected randomly, one by one from the add-on set and gradually were added to the training set in the process of incremental learning. Each selected photo was first pre-processed according to the steps described in Section 3 and then classified into an appropriate elementary set based on its derived attribute values. After that, the hierarchy of decision tables was adapted, as described before.

After all of photos in the add-on set have been utilized, the photos in test set were used to test the updated hierarchies with *probabilistic distance-based classification method*. The number of photos correctly classified and the accuracy rate were recorded. Once all 784 sets were utilized, the results were averaged. The average accuracy rate of the initial series of recognition experiments was about 93.31%.

The experiments were repeated twice. In the second series of experiments, the participants of each experimental set have not been changed, but the photos of each participant were used for different purposes. For example, a photo initially used for training purpose was selected to test the hierarchy or used in the add-on set. The average recognition accuracy rate in these two experiment cycles

was about 93.15% and 94.93%. The total accuracy of all experiments was about 93.8%.

During the process of incremental learning, we observed that the hierarchy was totally re-generated once a certain number of new photos randomly selected from different people was added. This fact underlines the importance of the appropriate selection of cut points and attributes in the process of classifier construction, because some of the condition attributes of some of the decision tables in the hierarchy would turn out not be the most appropriate features after new photos were added. Moreover, we also observed that the number of randomly selected photos causing total regeneration was not constant in practice: it can be a few or many photos. In the example shown in Figure 1, the regeneration points are clearly visible as the discontinuity points of the curve representing the variation of the λ-dependency of the hierarchy in the process of incremental learning. Once a random photo was added, the λ-dependency of the updated hierarchy always changed: either slightly increased or decreased. After a certain number of additions, the λ-dependency would deteriorate. Such a deterioration would cause total regeneration of the hierarchy. The value of the λ-dependency of the regenerated hierarchy would become relatively higher than that of the λ-dependency before total regeneration.

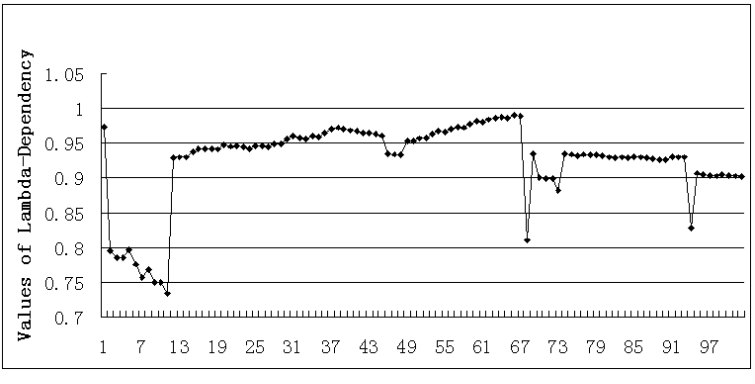


Fig. 1. Variation of Lambda Dependency

Experiments with data sets having more than 4-person were similar to the experiments with 4-person data sets. The average accuracy rate with 6-person data sets was about 88.63%, with 8-person sets was 83.96%, and with 10-person sets was 82.46%.

4.2 Experiments with Unbalanced Data Sets

When experimenting with unbalanced data sets, the procedural steps of these experiments are almost the same as described earlier. The experimental sets had photos from 6 or 8 participants. In order to identify the impact of facial expression on the accuracy rate, two groups of experimental sets were constructed.

They consisted of photos with neutral expression for the initial training set, with a smile for the test set, and with either angry or screaming expression for the add-on set. Similarly, in order to identify the impact of gender ratio on the accuracy rate, two groups of experimental sets were constructed, which consisted of photos with various facial expressions but unbalanced gender ratio. The results are summarized in Tables 1 and 2.

Table 1. Results with Data Sets with Specific Facial Expression

Group	Initial Train Set	Add-on Set	Test Set	Participants	Average Accuracy(%)
1	Neutral	Angry	Smile	6	92.41
2	Neutral	Scream	Smile	6	92.41
3	Neutral	Angry	Smile	8	91.04
4	Neutral	Scream	Smile	8	87.47

Table 2. Results with Data Sets with Unbalanced Gender Ratio

Gender Ratio	2M4F	4M2F	2M6F	6M2F
Accuracy(%)	88.13	84.84	76.41	76.99

Based on the results, we can summarize the performance of the soft-cut classifier when used in the mode of incremental learning as follows:

1. when experimental sets are balanced, the number of participants seems to be inversely proportional to the accuracy rate;
2. when experimental sets only have specific facial expressions, the accuracy would improve;
3. unlike the number of participants in the training set or the choice of facial expression, the unbalanced gender ratio in the training set has no apparent impact on the accuracy.

Therefore, we can conclude that during the incremental learning, the number of participants in the training set is the important factor affecting the accuracy rate, whereas the facial expression and the gender ratio do not appear to have a great impact on the accuracy rate. It seems that the performance of the soft-cut classifier would improve if the training set only had photos of a few, for example four, participants, no matter whether that training set is balanced or not.

5 Final Remarks

In this paper, we discussed the application of a new, variable precision rough set-based method called soft-cut and probabilistic distance-based classifier to the process of incremental learning for the purpose of face recognition. The

theoretical background, the application procedure, and some experiments were presented. We observed that during the process of incremental learning, when a photo was added, the λ-dependency of the updated hierarchy did not change consistently: it either slightly increased or decreased. Adding some more photos was eventually causing a significant drop in the value of λ-dependency, forcing the total regeneration of the hierarchy of decision tables. The testing by new photos revealed that the performance of the soft-cut classifier would deteriorate when the number of participants in the training set increased.

Acknowledgment. The support of the Natural Sciences and Engineering Research Council of Canada in partial funding the research presented in this article is gratefully acknowledged.

References

[1] Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer, Dordrecht (1991)
[2] Ziarko, W.: Variable Precision Rough Sets Model. Journal of Computer and System Sciences 46(1), 39–59 (1993)
[3] Ziarko, W.: Incremental Learning and Evaluation of Structures of Rough Decision Tables. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets IV. LNCS, vol. 3700, pp. 162–177. Springer, Heidelberg (2005)
[4] Zhao, W., Chellpappa, R., Philiips, P.J., Rosenfeld, A.: Face Recognition: A Literature Survey. ACM Computing Surveys 35(4), 399–458 (2003)
[5] Kidera, T., Ozawa, S., Abe, S.: An Incremental Learning Algorithm of Ensemble Classifier Systems. In: 2006 International Joint Conference on Neural Networks, Vancouver, BC, Canada, July 16-21 (2006)
[6] Polikar, R., Upda, L., Upda, S.S., Honavar, V.: Learn++: an Incremental Learning Algorithm for Supervised Neural Networks. IEEE Transactions on Systems, Man and Cybernetics 31(4), 497–508 (2001)
[7] Chen, X., Ziarko, W.: Experiments with Rough Set Approach to Face Recognition. In: Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) RSCTC 2008. LNCS (LNAI), vol. 5306, pp. 399–408. Springer, Heidelberg (2008)
[8] Ziarko, W.: Partition Dependencies in Hierarchies of Probabilistic Decision Tables. In: Wang, G.-Y., Peters, J.F., Skowron, A., Yao, Y. (eds.) RSKT 2006. LNCS (LNAI), vol. 4062, pp. 42–49. Springer, Heidelberg (2006)
[9] Linear Models Analyzing Unbalanced Data, Basic Methods, http://faculty.ucr.edu/~hanneman/linear_models/c5.html
[10] Shan, N., Ziarko, W.: Data-Based Acquisition and Incremental Modification of Classification Rules. Computational Intelligence: an International Journal 11(2), 357–370 (1995)
[11] Hariharana, G., Kannana, K., Sharmab, K.R.: Haar Wavelet in Estimating Depth Profile of Soil Temperature. Applied Mathematics and Computation 210(1), 119–125 (2009)
[12] Abatea, A.F., Nappi, M., Riccioa, D., Sabatinoa, G.: 2D and 3D Face Recognition: A Survey. Pattern Recognition Letters 28(14), 1885–1906 (2007)

A Comparison of Dynamic and Static Belief Rough Set Classifier

Salsabil Trabelsi[1], Zied Elouedi[1], and Pawan Lingras[2]

[1] Larodec, Institut Superieur de Gestion de Tunis, Tunisia
[2] Saint Mary's University Halifax, Canada

Abstract. In this paper, we propose a new approach of classification based on rough sets denoted Dynamic Belief Rough Set Classifier (D-BRSC) which is able to learn decision rules from uncertain data. The uncertainty appears only in decision attributes and is handled by the Transferable Belief Model (TBM), one interpretation of the belief function theory. The feature selection step of the construction procedure of our new technique of classification is based on the calculation of dynamic reduct. The reduction of uncertain and noisy decision table using dynamic approach which extracts more relevant and stable features yields more significant decision rules for the classification of the unseen objects. To prove that, we carry experimentations on real databases using the classification accuracy criterion. We also compare the results of D-BRSC with those obtained from Static Belief Rough Set Classifier (S-BRSC).

Keywords: rough sets, belief function theory, uncertainty, dynamic reduct, classification.

1 Introduction

The rough set theory proposed by Pawlak [6] constitutes a sound basis for data mining. It offers solutions to the problem of discretization, decision rule generation and solves the problem of attribute selection. For the latter, one of the ideas was to consider as relevant features those in reduct of the information system [5,6,8]. In fact, a reduct is a minimal set of attributes that preserves the ability to perform classifications as the whole attribute set does. Another issue in real world applications is the uncertain, imprecise or incomplete data. This kind of uncertainty exists in real-world applications like in marketing, finance, management and medicine. For example, some condition or decision attribute values in a client's database, used by the bank to plan a loan policy, are partially uncertain. Nevertheless, finding reducts from uncertain and noisy data leads to results which are unstable and sensitive to the sample data. Using dynamic reducts [1] allows getting better performance in very large datasets. In fact, rules induced by means of dynamic reducts are more appropriate to classify new objects, since these reducts are more stable and appear more frequently in sub-decision systems created by random samples of a given decision system.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 366–375, 2010.

For these reasons, we have previously developed an approach of feature selection based on rough set theory namely dynamic reduct [12] computed from uncertain data. In our context, the uncertainty appears only in decision attributes and is represented through the Transferable Belief Model (TBM), one interpretation of the belief function theory. In fact, this theory is considered as a useful model to represent quantified beliefs because it allows experts express partial beliefs in a much more flexible way than probability functions do. The belief function theory [7] is very applied in real world applications related to decision making and classification.

Due to the advantages of our dynamic feature selection approach [12], we propose in this paper a new approach of classification based on rough sets denoted Dynamic Belief Rough Set Classifier (D-BRSC) which is able to generate more stable decision rules from uncertain data which are better to classify unseen cases. The uncertainty exists only in decision attributes and is handled by the TBM. The feature selection step of the construction procedure of our new technique of classification is based on the calculation of dynamic reduct proposed originally in [12]. To evaluate our D-BRSC, we will carry experimentations on real databases using the classification accuracy criterion. Besides, we will compare the results with those obtained from Static Belief Rough Set Classifier (S-BRSC) [13].

This paper is organized as follows: Section 2 provides an overview of the rough set theory. Section 3 introduces the belief function theory as understood in the TBM. In Section 4, we propose under the belief function framework a new approach of classification called Dynamic Belief Rough Set Classifier (D-BSRC) based on dynamic approach of feature selection which is induced from uncertain data. Finally, we report results of experiments on real databases relative to our new approach Dynamic Belief Rough Set Classifier (D-BRSC) comparing with Static Belief Rough Set Classifier (S-BRSC) from [13].

2 Rough Sets

In this Section, we recall some basic notions related to information systems and rough sets [6]. An information system is a pair $A = (\mathrm{U}, \mathrm{C})$, where U is a non-empty, finite set called the *universe* and C is a non-empty, finite set of attributes. We also consider a special case of information systems called decision tables. A decision table is an information system of the form $A = (\mathrm{U}, \mathrm{C} \cup \{\mathrm{d}\})$, where $d \notin C$ is a distinguished attribute called *decision*. In this paper, the notation $c_i(o_j)$ is used to represent the value of a condition attribute $c_i \in C$ for $o_j \in U$.

For every set of attributes $B \subseteq C$, an equivalence relation denoted by IND_B and called the B-indiscernibility relation, is defined by

$$IND_B = U/B = \{[o_j]_B | o_j \in U\} \tag{1}$$

Where

$$[o_j]_B = \{o_i | \forall c \in B\, c(o_i) = c(o_j)\} \tag{2}$$

Let $B \subseteq C$ and $X \subseteq U$. We can approximate X by constructing the $B-lower$ and $B-upper\ approximations$ of X, denoted $\underline{\mathrm{B}}(\mathrm{X})$ and $\bar{\mathrm{B}}(\mathrm{X})$, respectively, where

$$\underline{\mathrm{B}}(X) = \{o_j | [o_j]_B \subseteq X\}\ and\ \bar{\mathrm{B}}(X) = \{o_j | [o_j]_B \cap X \neq \emptyset\} \tag{3}$$

2.1 Reduct and Core

A subset $B \subseteq C$ is a reduct of C with respect to d, iff B is minimal and:

$$Pos_B(\{d\}) = Pos_C(\{d\}) \tag{4}$$

Where $Pos_C(\{d\})$, called a positive region of the partition $U/\{d\}$ with respect to C.

$$Pos_C(\{d\}) = \bigcup_{X \in U/\{d\}} \underline{\mathrm{C}}(X) \tag{5}$$

The core is the most important subset of attributes, it is included in every reduct.

$$Core(A,\, d) = \bigcap RED(A,\, d) \tag{6}$$

Where $RED(A, d)$ is the set of all reducts of A relative to d.

2.2 Dynamic Reduct

If $A = (U,\, C \cup \{d\})$ is a decision table, then any system $B = (U',\, C \cup \{d\})$ such that $U' \subseteq U$ is called a subtable of A. Let F be a family of subtables of A [1].

$$DR(A,\, F) = RED(A, d) \cap \bigcap_{B \in F} RED(B,\, d) \tag{7}$$

Any element of $DR(A,\, F)$ is called an F-dynamic reduct of A. From the definition of dynamic reducts, it follows that a relative reduct of A is dynamic if it is also a reduct of all subtables from a given family F. This notation can be sometimes too restrictive, so we apply a more general notion of dynamic reduct. They are called $(F,\, \varepsilon)$-dynamic reducts, where $0 \leq \varepsilon \leq 1$. The set $DR_\varepsilon(A,\, F)$ of all $(F,\, \varepsilon)$-dynamic reducts is defined by:

$$DR_\varepsilon(A,\, F) = \left\{ R \in RED(A,\, d) : \frac{|\{B \in F : R \in RED(B,\, d)\}|}{|F|} \geq 1 - \varepsilon \right\} \tag{8}$$

3 Belief Function Theory

The belief function theory is proposed by Shafer [7] as a useful tool to represent uncertain knowledge. Here, we introduce only some basic notations related to the TBM [9], one interpretation of the belief function theory. Let Θ, frame of discernment, be a finite set of exhaustive elements to a given problem. All the subsets of Θ belong to the power set of Θ, denoted by 2^Θ. The bba (basic belief

assignment) is a function representing the impact of a piece of evidence on the subsets of the frame of discernment Θ and is defined as follows:

$$m : 2^{\Theta} \rightarrow [0, 1]$$

$$\sum_{E \subseteq \Theta} m(E) = 1 \tag{9}$$

Where *m(E)*, named a basic belief mass (bbm), shows the part of belief exactly committed to the element E. The bba's induced from distinct pieces of evidence are combined by the rule of combination [9].

$$(m_1 \textcircled{\cap} m_2)(E) = \sum_{F,G \subseteq \Theta : F \cap G = E} m_1(F) \times m_2(G) \tag{10}$$

4 Dynamic Belief Rough Set Classifier (D-BRSC)

In this Section, we propose a new approach of classification called Dynamic Belief Rough Set Classifier (D-BRSC) based on dynamic approach of feature selection. This classifier is built from uncertain data under the belief function framework. The uncertainty appears only in decision attribute and is handled by the TBM. Before describing the main steps of the construction procedure of D-BRSC especially the feature selection, we need at first to present the modified basic concepts of rough sets under uncertainty [11] such as decision table, tolerance relation, set approximation, positive region, reduct and core.

4.1 Basic Concepts of Rough Sets under Uncertainty

Uncertain decision table Our uncertain decision system is given by $A = (U, C \cup \{ud\})$, where $U = \{o_j : 1 \leq j \leq n\}$ is characterized by a set of certain condition attributes C=$\{c_1, c_2,...,c_k\}$, and an uncertain decision attribute ud. We represent the uncertainty of each object o_j by a bba m_j expressing beliefs on decisions defined on the frame of discernment Θ=$\{ud_1, ud_2,...,ud_s\}$ describing the possible values of ud. These bba's are given by an expert.

Example: Let us use Table 1 to describe our uncertain decision table. It contains eight objects, three certain condition attributes C=$\{Hair, Eyes, Height\}$ and an uncertain decision attribute ud with two possible values $\{ud_1, ud_2\}$ representing Θ. For the object o_3, 0.7 of beliefs are exactly committed to the decision ud_1, whereas 0.3 of beliefs is assigned to the whole of frame of discernment Θ (ignorance). With bba, we can represent the certain case, like for the objects o_2, o_5 and o_7. The decision rules induced from the uncertain decision table are denoted belief decision rules where the decision is represented by a bba: *If* $Hair = Blond$ *and* $Eyes = Brown$ *and* $Height = Short$ *Then* $m_3(\{ud_1\}) = 0.7 \quad m_3(\Theta) = 0.3$.

Table 1. Uncertain decision table

U	Hair	Eyes	Height	ud
o_1	Dark	Brown	Short	$m_1(\{ud_2\}) = 0.5 \quad m_1(\Theta) = 0.5$
o_2	Blond	Blue	Middle	$m_2(\{ud_2\}) = 1$
o_3	Blond	Brown	Short	$m_3(\{ud_1\}) = 0.7 \quad m_3(\Theta) = 0.3$
o_4	Blond	Brown	Tall	$m_4(\{ud_1\}) = 0.95 \quad m_4(\{ud_2\}) = 0.05$
o_5	Dark	Brown	Short	$m_5(\{ud_2\}) = 1$
o_6	Blond	Blue	Middle	$m_6(\{ud_2\}) = 0.95 \quad m_6(\Theta) = 0.05$
o_7	Dark	Brown	Tall	$m_7(\{ud_1\}) = 1$
o_8	Dark	Brown	Middle	$m_8(\{ud_1\}) = 0.975 \; m_8(\Theta) = 0.025$

Tolerance relation. The indiscernibility relation for the decision attribute $U/\{ud\}$ is not the same as in the certain case. The decision value is represented by a bba. In our case, it will be denoted *tolerance relation*. So, we need to assign each object to the right tolerance class. The idea is to use the distance between the two bba's m_j and a certain bba m (such that $m(\{ud_i\}) = 1$). Many distance measures between two bba's were developed. Some of them are based on pignistic transformation [3,14]. This kind of distances may lose information given by the initial bba's. However, the distance measures developed in [2,4] are directly defined on bba's. In our case, we choose the distance measure proposed in [2] which satisfies more properties such as non-negativity, non-degeneracy and symmetry. For every ud_i, we define a tolerance class as follows:

$$X_i = \{o_j | dist(m, m_j) < 1 - threshold\} \tag{11}$$

Besides, we define a tolerance relation as follows:

$$IND_{\{ud\}} = U/\{ud\} = \{X_i | ud_i \in \Theta\} \tag{12}$$

Where *dist* is a distance measure between two bba's.

$$dist(m_1, m_2) = \sqrt{\frac{1}{2}(\| \overrightarrow{m_1} \|^2 + \| \overrightarrow{m_2} \|^2 - 2 < \overrightarrow{m_1}, \overrightarrow{m_2} >)} \tag{13}$$

Where $< \overrightarrow{m_1}, \overrightarrow{m_2} >$ is the scalar product defined by:

$$< \overrightarrow{m_1}, \overrightarrow{m_2} > = \sum_{i=1}^{|2^\Theta|} \sum_{j=1}^{|2^\Theta|} m_1(A_i) m_2(A_j) \frac{|A_i \cap A_j|}{|A_i \cup A_j|} \tag{14}$$

with $A_i, A_j \in 2^\Theta$ for $i, j = 1, 2, \cdots, |2^\Theta|$. $\| \overrightarrow{m_1} \|^2$ is then the square norm of $\overrightarrow{m_1}$.

Remark: It should be noted here that we replace the term equivalence class from the certain decision attribute case by tolerance class for the uncertain decision attribute, because the resulting classes may overlap.

Example: Let us continue with the same example to compute the tolerance classes based on the uncertain decision attribute $U/\{ud\}$. For the uncertain

decision value ud_1: (if we take threshold equal to 0.1, we btain these results)
$dist(m(\{ud_1\}) = 1, m_1) = 0.67$ (using eq. 13) < 0.9
$dist(m(\{ud_1\}) = 1, m_3) = 0.34 < 0.9$
$dist(m(\{ud_1\}) = 1, m_4) = 0.0735 < 0.9$
$dist(m(\{ud_1\}) = 1, m_7) = 0 < 0.9$
$dist(m(\{ud_1\}) = 1, m_8) = 0.065 < 0.9$
So, $X_1 = \{o_1, o_3, o_4, o_7, o_8\}$
The same for the uncertain decision value ud_2.
So, $X_2 = \{o_1, o_2, o_3, o_5, o_6\}$
$U/\{ud\} = \{\{o_1, o_3, o_4, o_7, o_8\}, \{o_1, o_2, o_3, o_5, o_6\}\}$

Set approximation. In the uncertain context, the two subsets *lower* and *upper* approximations are redefined using two steps:

1. We combine the bba's for each equivalence class from U/C using the operator mean which is more suitable than the rule of combination in eq. 10 which is proposed especially to combine different beliefs on decision for one object and not different bba's for different objects.
2. We compute the new *lower* and *upper* approximations for each tolerance class X_i from $U/\{ud\}$ based on uncertain decision attribute ud_i as follows:

$$\underline{C}X_i = \{o_j | [o_j]_C \cap X_i \neq \emptyset \, and \, dist(m, m_{[o_j]_C}) \leq threshold\} \tag{15}$$

$$\bar{C}X_i = \{o_j | [o_j]_C \cap X_i \neq \emptyset\} \tag{16}$$

We find in the new *lower* approximation all equivalence classes from U/C included in X_i where the distance between the combined bba $m_{[o_j]_C}$ and the certain bba m (such that $m(\{ud_i\}) = 1$) is less than a *threshold*. However, the *upper* approximation is computed in the same manner as in the certain case.

Example: We continue with the same example to compute the new *lower* and *upper* approximations. After the first step, we obtain the combined bba for each equivalence class from U/C using operator mean. Table 2 represents the combined bba for the equivalence classes $\{o_1, o_5\}$ and $\{o_2, o_6\}$. Next, we compute the *lower* and *upper* approximations for each tolerance class $U/\{ud\}$. We will use $threshold = 0.1$. For the uncertain decision value ud_1, let $X_1 = \{o_1, o_3, o_4, o_7, o_8\}$. The subsets $\{o_3\}$, $\{o_4\}$, $\{o_7\}$ and $\{o_8\}$ are included to X_1. We should check the distance between their bba and the certain bba $m(\{ud_1\}) = 1$.
$dist(m(\{ud_1\}) = 1, m_3) = 0.34 > 0.1$
$dist(m(\{ud_1\}) = 1, m_4) = 0.0735 < 0.1$
$dist(m(\{ud_1\}) = 1, m_7) = 0 < 0.1$
$dist(m(\{ud_1\}) = 1, m_8) = 0.065 < 0.1$
$\underline{C}(X_1) = \{o_4, o_7, o_8\}$ and $\bar{C}(X_1) = \{o_1, o_3, o_4, o_5, o_7, o_8\}$
The same for uncertain decision value ud_2, let $X_2 = \{o_1, o_2, o_3, o_5, o_6\}$
$\underline{C}(X_2) = \{o_2, o_6\}$ and $\bar{C}(X_2) = \{o_1, o_2, o_3, o_5, o_6\}$

Table 2. The combined bba for the subsets $\{o_1, o_5\}$ and $\{o_2, o_6\}$

Object	$m(\{ud_1\})$	$m(\{ud_2\})$	$m(\Theta)$
o_1	0	0.5	0.5
o_5	0	1	0
$m_{1,5}$	0	0.75	0.25
o_2	0	1	0
o_6	0	0.95	0.05
$m_{2,6}$	0	0.975	0.025

Positive region. With the new *lower* approximation, we can redefine the positive region:

$$UPos_C(\{ud\}) = \bigcup_{X_i \in U/\{ud\}} \underline{C}X_i \tag{17}$$

Example: Let us continue with the same example, to compute the positive region of A. $UPos_C(\{\text{ud}\})$=$\{o_2, o_4, o_6, o_7, o_8\}$.

Reduct and core. Using the new formalism of positive region, we can redefine the reduct of A as a minimal set of attributes $B \subseteq C$ such that:

$$UPos_B(\{ud\}) = UPos_C(\{ud\}) \tag{18}$$

$$UCore(A, ud) = \bigcap URED(A, ud) \tag{19}$$

Where $URED(A, ud)$ is the set of all reducts of A relative to ud.

Example: Using our example, we find that $UPos_{\{Hair, Height\}}(\{ud\})$= $UPos_{\{Eyes, Height\}}(\{ud\})$= $UPos_C(\{ud\})$. So, we have two possible reducts $\{Hair, Height\}$ and $\{Eyes, Height\}$. The attribute $Height$ is the relative core.

4.2 The Construction Procedure of the D-BRSC

1. **Feature selection:** It is the more important step which consists of removing the superfluous condition attributes that are not in reduct. This leaves us with a minimal set of attributes that preserve the ability to perform same classification as the original set of attributes. However, our decision table shown in subsection 4.1 is characterized by a high level of uncertain and noisy data. One of the issues with such a data is that the resulting reducts are not stable, and are sensitive to sampling. The belief decision rules generated are not suitable for classification. The solution to this problem is to redefine the concept of dynamic reduct in the new context as we have done in [12]. The rules calculated by means of dynamic reducts are better predisposed to classify unseen objects, because they are the most frequently appearing reducts in sub-decision systems created by random samples of a

given decision system. According to the uncertain context, we can redefine the concept of dynamic reduct as follows:

$$UDR(A,\, F) = URED(A, ud) \cap \bigcap_{B \in F} URED(B,\, ud) \tag{20}$$

Where F be a family of subtables of A. This notation can be sometimes too restrictive so we apply a more general notion of dynamic reduct. They are called $(F,\, \varepsilon)$-dynamic reducts, where $1 \geq \varepsilon \geq 0$. The set $UDR_\varepsilon(A,\, F)$ of all $(F,\, \varepsilon)$-dynamic reducts is defined by: $UDR_\varepsilon(A,\, F) =$

$$\left\{ R \in URED(A,\, ud) : \frac{|\{B \in F : R \in URED(B,\, ud)\}|}{|F|} \geq 1 - \varepsilon \right\} \tag{21}$$

2. **Eliminate the redundant objects:** After removing the superfluous condition attributes, we find redundant objects. They may not have the same bba on decision attribute. So, we use their combined bba's using the operator mean.
3. **Eliminate the superfluous condition attribute values:** In this step, we compute the reduct value for each belief decision rule R_j of the form: **If** $C(o_j)$ **then** m_j. For all $B \subset C$, let $X = \{o_k \mid B(o_j) = B(o_k)\}$ **If** $Max(dist(m_j, m_k)) \leq$ threshold **then** B is a reduct value of R_j.
Remark: In the case of uncertainty, the *threshold* gives more flexibility to the calculation of tolerance class, set approximations and reduct value. It is fixed by the user and it should be the same value to be coherent.

5 Experimentations

In our experiments, we have performed several tests on real databases obtained from the U.C.I. repository[1] to evaluate D-BRSC. A brief description of these databases is presented in Table 3. These databases are artificially modified in order to include uncertainty in decision attribute. We take different degrees of uncertainty (Low, Middle and High) based on increasing values of probabilities P used to transform the actual decision value d_i of each object o_j to a bba $m_j(\{d_i\}) = 1 - P$ and $m_j(\Theta) = P$. A larger P gives a larger degree of uncertainty.

- Low degree of uncertainty: we take $0 < \mathrm{P} \leq 0.3$
- Middle degree of uncertainty: we take $0.3 < \mathrm{P} \leq 0.6$
- High degree of uncertainty: we take $0.6 < \mathrm{P} \leq 1$

The relevant criterion used to evaluate the performance of D-BRSC is the classification accuracy (PCC[2]) of the generated belief decision rules. To further evaluate the new classifier, we will compare the experimental results relative to D-BRSC with those obtained from Static Belief Rough Set Classifier (S-BRSC) proposed originally in [13].

[1] http://www.ics.uci.edu/ mlearn/MLRepository.html
[2] Percent of Correct Classification.

Table 3. Description of databases

Database	#instances	#attributes	#decision values
W. Breast Cancer	690	8	2
Balance Scale	625	4	3
C. Voting records	497	16	2
Zoo	101	17	7
Nursery	12960	8	3
Solar Flares	1389	10	2
Lung Cancer	32	56	3
Hyes-Roth	160	5	3
Car Evaluation	1728	6	4
Lymphography	148	18	4
Spect Heart	267	22	2
Tic-Tac-Toe Endgame	958	9	2

From Table 4, we can conclude that reducing uncertain and noisy database using dynamic feature selection approach is more suitable for classification of the unseen cases than the static approach. It is true for all chosen databases and for all degrees of uncertainty. For example, the PCC for Car Evaluation database under high degree of uncertainty is 84.17% with dynamic reduct and 72.77% with static reduct. We further note that the PCC slightly increases when the uncertainty decreases for the both approaches.

Table 4. Experimentation results relative to D-BRSC and S-BRSC

Database	D-BRSC PCC (%)			S-BRSC PCC (%)		
	Low	Middle	High	Low	Middle	High
W. Breast Cancer	86.87	86.58	86.18	83.41	83.39	82.17
Balance Scale	83.46	83.21	83.03	77.3	77.83	77.76
C. Voting records	98.94	98.76	98.52	97.91	97.76	97.71
Zoo	96.52	96.47	95.87	90.22	90.41	90.37
Nursery	96.68	96.21	96.07	94.34	94.13	94.11
Solar Flares	88.67	88.61	88.56	85.72	85.61	85.46
Lung Cancer	75.77	75.50	75.33	66.43	66.08	66.08
Hyes-Roth	97.96	97.15	96.75	83.66	83.31	82.14
Car Evaluation	84.46	84.01	84.17	73.39	73.22	72.77
Lymphography	83.24	83.03	82.67	79.25	78.97	78.94
Spect Heart	85.34	85.28	85.07	83.54	83.21	82.17
Tic-Tac-Toe Endgame	86.26	86.21	86.18	83.93	83.72	83.47

6 Conclusion and Future Work

In this paper, we have proposed a new approach of classification called Dynamic Belief Rough Set Classifier (D-BRSC) based on rough sets induced from uncertain data under the belief function framework. This technique of classification

is based on dynamic approach for feature selection. We have done experimentations on real databases to evaluate our proposed classifier based on classification accuracy criterion. To further evaluate our approach, we compare the results with those obtained from Static Belief Rough Set Classifier (S-BRSC). According to the experimentation results, we find that generating belief decision rules based on dynamic approach of feature selection is more suitable for classification process than static one.

References

1. 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 (LNAI), vol. 869, pp. 346–355. Springer, Heidelberg (1994)
2. Bosse, E., Jousseleme, A.L., Grenier, D.: A new distance between two bodies of evidence. Information Fusion 2, 91–101 (2001)
3. Elouedi, Z., Mellouli, K., Smets, P.: Assessing sensor reliability for multisensor data fusion within the transferable belief model. IEEE Trans. Syst. Man Cybern. 34(1), 782–787 (2004)
4. Fixen, D., Mahler, R.P.S.: The modified Dempster-Shafer approach to classification. IEEE Trans. Syst. Man Cybern. 27(1), 96–104 (1997)
5. Modrzejewski, M.: Feature selection using rough sets theory. In: Proceedings of the 11th International Conference on Machine Learning, pp. 213–226 (1993)
6. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishing, Dordrecht (1991)
7. Shafer, G.: A mathematical theory of evidence. Princeton University Press, Princeton (1976)
8. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Slowinski, R. (ed.) Intelligent Decision Support, pp. 331–362. Kluwer Academic Publishers, Boston (1992)
9. Smets, P., Kennes, R.: The transferable belief model. Artificial Intelligence 66(2), 191–234 (1994)
10. Tessem, B.: Approximations for efficient computation in the theory of evidence. Artif. Intell. 61(2), 315–329 (1993)
11. Trabelsi, S., Elouedi, Z.: Learning decision rules from uncertain data using rough sets. In: The 8th International FLINS Conference on Computational Intelligence in Decision and Control, Madrid, Spain, September 21-24, pp. 114–119. World scientific, Singapore (2008)
12. Trabelsi, S., Elouedi, Z., Lingras, P.: Dynamic reduct from partially uncertain data using rough sets. In: Sakai, H., Chakraborty, M.K., Hassanien, A.E., Ślęzak, D., Zhu, W. (eds.) RSFDGrC 2009. LNCS (LNAI), vol. 5908, pp. 160–167. Springer, Heidelberg (2009)
13. Trabelsi, S., Elouedi, Z., Lingras, P.: Belief rough set classifier. In: Gao, Y., Japkowicz, N. (eds.) Canadian AI 2009. LNCS (LNAI), vol. 5549, pp. 257–261. Springer, Heidelberg (2009)
14. Zouhal, L.M., Denoeux, T.: An evidence-theory k-NN rule with parameter optemization. IEEE Trans. Syst. Man Cybern. C 28(2), 263–271 (1998)

Rule Generation in Lipski's Incomplete Information Databases

Hiroshi Sakai[1], Michinori Nakata[2], and Dominik Ślęzak[3,4]

[1] Mathematical Sciences Section, Department of Basic Sciences,
Faculty of Engineering, Kyushu Institute of Technology
Tobata, Kitakyushu 804, Japan
sakai@mns.kyutech.ac.jp
[2] Faculty of Management and Information Science,
Josai International University
Gumyo, Togane, Chiba 283, Japan
nakatam@ieee.org
[3] Institute of Mathematics, University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
[4] Infobright Inc., Poland
Krzywickiego 34 pok. 219, 02-078 Warsaw, Poland
slezak@infobright.com

Abstract. *Non-deterministic Information Systems* (*NISs*) are well known as systems for handling information incompleteness in data. In our previous work, we have proposed *NIS-Apriori* algorithm aimed at extraction of decision rules from *NISs*. *NIS-Apriori* employs the minimum and the maximum supports for each descriptor, and it effectively calculates the criterion values for defining rules. In this paper, we focus on *Lipski's Incomplete Information Databases* (*IIDs*), which handle non-deterministic information by means of the sets of values and intervals. We clarify how to understand decision rules in *IIDs* and appropriately adapt our *NIS-Apriori* algorithm to generate them. Rule generation in *IIDs* turns out to be more flexible than in *NISs*.

Keywords: Lipski's incomplete information databases, Rule generation, Apriori algorithm, External and internal interpretations, Rough sets.

1 Introduction

In our previous research, we focused on rule generation in *Non-deterministic Information Systems* (*NISs*)[12]. In contrast to *Deterministic Information Systems* (*DISs*)[11,15], *NISs* were proposed by Pawlak [11] and Orłowska [10] in order to better handle information incompleteness in data. Incompleteness can be here understood as related to null values, unknown values, missing values, but also to partially defined values represented by the subsets of possible values. Since the emergence of incomplete information research and applications [4,7,8,10], *NISs* have been playing an important role, both with regards to their mathematical foundations and algorithmic frameworks.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 376–385, 2010.

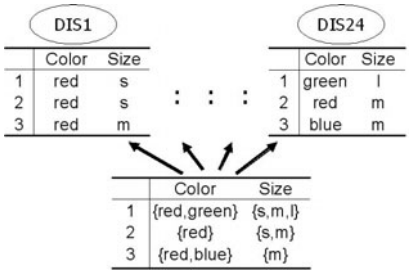


Fig. 1. A NIS and 24 derived $DISs$. The number of derived $DISs$ is finite. However, it usually increases in the exponential order.

In this paper, we consider rule generation in *Incomplete Information Databases* ($IIDs$) proposed by Lipski [8,9]. $IIDs$ are more flexible than $NISs$. This is because definition of attribute values in $IIDs$ is more general than in $NISs$. We adapt our previous results related to *Rough Non-deterministic Informa- tion Analysis* ($RNIA$) [12,13,14] in $NISs$ in order to address the problem of rule generation in $IIDs$. The obtained methodology can be treated as a step towards more general rule-based data analysis, where both data values and descriptors take various forms of incompleteness, vagueness or non-determinism.

The paper is organized as follows: Section 2 recalls data representation and rule generation in $DISs$ and $NISs$. Sections 3, 4 and 5 introduce the same for $IIDs$. Section 6 presents open challenges. Section 7 concludes the paper.

2 Rule Generation in DISs and NISs

We omit formal definitions of $DISs$ and $NISs$. Instead, we show an example in Fig. 1. We identify a DIS with a standard table. In a NIS, each attribute value is a set. If the value is a singleton, there is no incompleteness. Otherwise, we interpret it as a set of possible values, i.e., we assume that it includes the actual value but we do not know which of them is the actual one.

A rule (more correctly, a candidate for a rule) is an implication τ in the form of $Condition\_part \Rightarrow Decision\_part$. We employ $support(\tau)$ and $accuracy(\tau)$ to express the rule's appropriateness as follows [1,11] (see also Fig. 2.):

Specification of the rule generation task in a DIS
For threshold values α and β $(0 < \alpha, \beta \leq 1)$, find each implication τ satisfying $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$.

The *Apriori* algorithm proposed to search for such rules by Agrawal in [1] is now one of the most representative methods in data mining [2].

In both $DISs$ and $NISs$, the same τ may be generated by different tuples. We use notation τ^x to express that τ is generated by an object x. Let $DD(\tau^x)$ denote a set of derived $DISs$ such that τ^x holds.

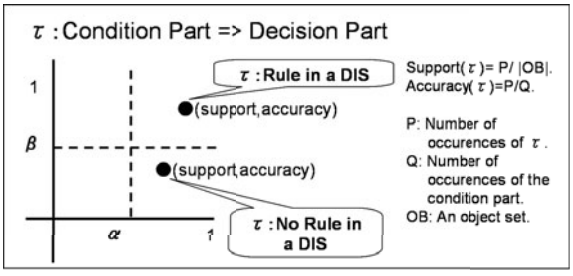


Fig. 2. A pair (*support*,*accuracy*) corresponding to the implication τ

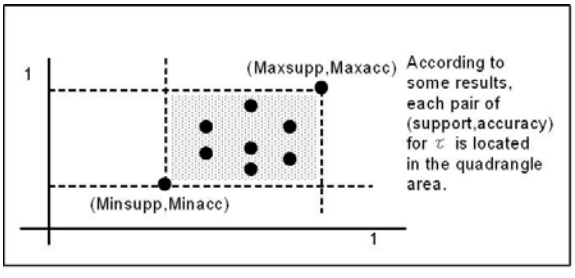


Fig. 3. A distribution of pairs (*support*,*accuracy*) for τ^x. There exists $\psi_{min} \in DD(\tau^x)$ which makes both $support(\tau^x)$ and $accuracy(\tau^x)$ the minimum. There exists $\psi_{max} \in DD(\tau^x)$ which makes both $support(\tau^x)$ and $accuracy(\tau^x)$ the maximum. We denote such quantities as *minsupp*, *minacc*, *maxsupp* and *maxacc*, respectively.

Example 1. In Fig. 1, $\tau : [Color, red] \Rightarrow [Size, m]$ occurs for all objects. We have $|DD(\tau^1)| = 4$, $|DD(\tau^2)| = 12$, $|DD(\tau^3)| = 12$. In $DD(\tau^1)$, there is a derived DIS such that $support(\tau^1) = 1$ and $accuracy(\tau^1) = 1$. In another derived DIS, $support(\tau^1) = 1/3$ and $accuracy(\tau^1) = 1/2$ hold.

Specification of the rule generation task in a NIS
(The lower system) Find each implication τ such that there exists an object x such that $support(\tau^x) \geq \alpha$ and $accuracy(\tau^x) \geq \beta$ hold in each $\psi \in DD(\tau^x)$.
(The upper system) Find each implication τ such that there exists an object x such that $support(\tau^x) \geq \alpha$ and $accuracy(\tau^x) \geq \beta$ hold in some $\psi \in DD(\tau^x)$.

Both above systems depend on $|DD(\tau^x)|$. In [14], we proved simplifying results illustrated by Fig. 3. We also showed how to effectively compute $support(\tau^x)$ and $accuracy(\tau^x)$ for ψ_{min} and ψ_{max} independently from $|DD(\tau^x)|$.

Equivalent specification of the rule generation task in a NIS
(The lower system) Find each implication τ such that there exists an object x such that $minsupp(\tau^x) \geq \alpha$ and $minacc(\tau^x) \geq \beta$ (see Fig. 3).
(The upper system) Find each implication τ such that there exists an object x such that $maxsupp(\tau^x) \geq \alpha$ and $maxacc(\tau^x) \geq \beta$ (see Fig. 3).

Table 1. An example of Lipski's Incomplete Information Database (IID).

OB	Age	$Dept\#$	$Hireyear$	Sal
x_1	$[60, 70]$	$\{1, \cdots, 5\}$	$\{70, \cdots, 75\}$	$\{10000\}$
x_2	$[52, 56]$	$\{2\}$	$\{72, \cdots, 76\}$	$(0, 20000]$
x_3	$\{30\}$	$\{3\}$	$\{70, 71\}$	$(0, \infty)$
x_4	$(0, \infty)$	$\{2, 3\}$	$\{70, \cdots, 74\}$	$\{22000\}$
x_5	$\{32\}$	$\{4\}$	$\{75\}$	$(0, \infty)$

In [13,14], we extended rule generation onto $NISs$ and implemented a software tool called NIS-$Apriori$. NIS-$Apriori$ does not depend upon the number of derived $DISs$. We are now working on its SQL-based version called SQL-NIS-$Apriori$ [16]. We also continue discussions on various challenges in front of Data Mining and Data Warehousing that are related to complex, inexact data types [6]. The following sections are one of the next steps along this path.

3 Lipski's Incomplete Information Databases

Now, we advance from $NISs$ to $IIDs$. In $NISs$, each attribute value is given as a subset of a domain. In $IIDs$, in case of ordered sets, we also handle intervals. Lipski coped with mathematical foundations of the question answering systems in $IIDs$ and proposed some software solutions in [8,9].

Table 1 is an example of IID cited from [8]. For Age whose domain is $(0, \infty)$, information about two persons x_3 and x_5 is definite. Information on three persons x_1, x_2 and x_4 is indefinite. For each of these cases, information is given as an interval. For $Dept\#$, each attribute value is not an interval but a subset of a set of all department numbers. In Table 1, attributes Age and Sal require intervals and attributes $Dept\#$ and $Hireyear$ require sets. We call each of the former an *interval-attribute*, and each of the latter – a *set-attribute*.

For simplicity, we restrict in this paper only to one type of interval-attributes, namely the attributes with numeric values. Of course, there are also other cases where we can define intervals, e.g., the sets of words with lexicographic order.

Definition 1. *(A revised definition of [8]) An Incomplete Information Database (IID) is a quadruplet* $(OB, AT, \{VAL_A | A \in AT\}, \{g_A | A \in AT\})$. *$OB$ and AT are finite sets of objects and attributes. g_A is defined as follows:*

(1) If $A \in AT$ is a set-attribute, VAL_A is a finite set and g_A is a mapping from OB to $P(VAL_A)$ (a power set of VAL_A).
(2) If $A \in AT$ is an interval-attribute, VAL_A is a set of numerical values with a standard order $<$ and g_A is a mapping from OB to $\{[l, u), (l, u], (l, u) | l, u \in VAL_A, l < u\} \cup \{[l, u] | l, u \in VAL_A, l \leq u\}$.

For an interval-attribute $A \in AT$ and an object $x \in OB$, we denote by $\overline{g_A(x)} = (u - l)$ the length of interval $g_A(x)$.

Table 2. A complete extension ψ of IID from Table 1. Here, $\gamma_{Age} = 2$, $\gamma_{Sal} = 1000$. A complete extension takes the same role for IID as a derived DIS for NIS. For complete extensions of an IID, we can imagine a chart analogous to Fig. 1.

OB	Age	$Dept\#$	$Hireyear$	Sal
x_1	$[63, 65]$	$\{1\}$	$\{71\}$	$\{10000\}$
x_2	$[54, 56]$	$\{2\}$	$\{76\}$	$[19000, 20000]$
x_3	$\{30\}$	$\{3\}$	$\{71\}$	$[13000, 14000]$
x_4	$[50, 52]$	$\{3\}$	$\{71\}$	$\{22000\}$
x_5	$\{32\}$	$\{4\}$	$\{75\}$	$[14000, 15000]$

Each $g_A(x)$ is either a set or an interval, and it is interpreted as that the actual value is in the set or the interval. In case of intervals, there may be uncountable number of sub-intervals. For example, there are uncountably many sub-intervals for $(0, 1) = \{x \in R \text{ (a set of real numbers)} | \, 0 < x < 1\}$.

Definition 2. *For an interval attribute $A \in AT$, let us fix a threshold value $\gamma_A > 0$. We say that an interval is definite, if its length is not higher than γ_A. Otherwise, we say that it is indefinite. We call γ_A a resolution of VAL_A.*

Definition 3. *(A revised definition of [8]) Let us consider an $IID = (OB, AT, \{VAL_A | \, A \in AT\}, \{g_A | \, A \in AT\})$ and a set $ATR \subseteq AT$. Consider ψ in the form of $(OB, ATR, \{VAL_A | \, A \in ATR\}, \{h_A | \, A \in ATR\})$. If ψ satisfies (1) or (2) for each $A \in ATR$ and $x \in OB$, we say that it is an extension of IID:*

(1) For a set-attribute A: $h_A(x) \neq \emptyset$ and $h_A(x) \subseteq g_A(x)$;
(2) For an interval-attribute A: $h_A(x) \geq \gamma_A$ and $h_A(x) \subseteq g_A(x)$.

If $h_A(x)$ is either a singleton or a definite interval for each $A \in ATR$ and $x \in OB$, we say that ψ is a complete extension of IID.

4 The External and Internal Modes in IIDs

Lipski originally proposed two interpretations: the *external* and the *internal interpretation* [8]. The external interpretation does not allow for expressing information incompleteness in descriptors. The internal interpretation allows it. We illustrate it in Fig. 4.

We define two modes depending upon the usage of descriptors. In the external mode, we employ descriptors of the following form:
$(EXT\ 1)$ $[A, val]$ $(val \in VAL_A)$ for a set-attribute;
$(EXT\ 2)$ $[A, [l, u]]$ $((u - l) = \gamma_A)$ for an interval-attribute with resolution γ_A.
In the internal mode, we employ descriptors of the following form:
$(INT\ 1)$ $[A, SET]$ $(SET \subseteq VAL_A)$ for a set-attribute;
$(INT\ 2)$ $[A, [l, u]]$ $(\gamma_A \leq (u - l))$ for an interval-attribute with resolution γ_A.

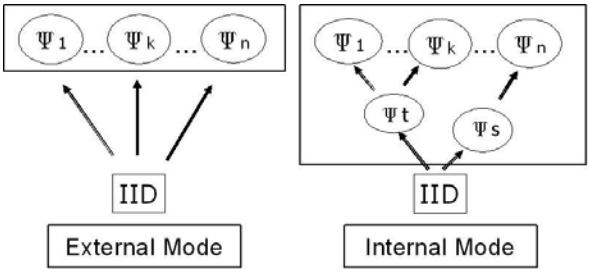


Fig. 4. In the external mode, one handles complete information and complete extensions $\Psi_1, ..., \Psi_n$. In the internal mode, one handles incomplete information and all extensions. In our rough non-deterministic information analysis ($RNIA$), we have coped with the analogy of external mode. We do not handle the internal mode yet.

As for descriptors of the interval-attributes, the selection of intervals is a very important issue. We discuss this problem in the subsequent section. Here, we assume that the descriptors are given. From now on, we employ the notation $DESC_{Attribute}$ for expressing descriptors. The problem is to obtain appropriate implications in the form of $\wedge_i DESC_i \Rightarrow DESC_{DEC}$.

5 Rule Generation in the External Mode

As mentioned before, a complete extension ψ is analogous to a derived DIS. Basing on this analogy, we can reconsider *support* and *accuracy* in $IIDs$.

5.1 Support and Accuracy of Implications in the External Mode

Definition 4. *Let us consider a complete extension ψ of an IID. We define the satisfiability of descriptors in ψ as follows:*

(1) For a set-attribute $A \in AT$, an object x in ψ satisfies the descriptor $[A, val]$, if $h_A(x) = \{val\}$.
(2) For an interval-attribute $A \in AT$, an object x in ψ satisfies the descriptor $[A, [l, u]]$, if $h_A(x) \subseteq [l, u]$.

Definition 5. *Let us consider a complete extension ψ of an IID and an implication $\tau : \wedge_i DESC_i \Rightarrow DESC_{DEC}$. Let us define the following:*

(1) $OBJ_\psi(CON) = \{x \in OB | \, x$ in ψ satisfies each descriptor $DESC_i\}$;
(2) $OBJ_\psi(CON, DEC) = \{x \in OB | \, x$ in ψ satisfies each descriptor in $\tau\}$.

If $OBJ_\psi(CON, DEC) \neq \emptyset$, we say τ is definable in ψ. Then we define:

(3) $support_\psi(\tau) = |OBJ_\psi(CON, DEC)|/|OB|$;
(4) $accuracy_\psi(\tau) = |OBJ_\psi(CON, DEC)|/|OBJ_\psi(CON)|$.

If τ is not definable, we do not define (3) and (4).

Example 2. For Table 1, consider $\gamma_{Age} = 2$ and $\gamma_{Sal} = 1000$. Consider complete extension ψ in Table 2. Consider $\tau : [Age, [30, 32]] \Rightarrow [Sal, [13000, 14000]]$. We have $OBJ_\psi(\{Age\}) = \{x_3, x_5\}$ and $OBJ_\psi(\{Age\}, \{Sal\}) = \{x_3\}$. According to Definition 5, we have $support_\psi(\tau) = 1/5$ and $accuracy_\psi(\tau) = 1/2$. Further, consider $\tau' : [Age, [50,\ 52]] \Rightarrow [Sal,\ [20000, 21000]]$. We have $OBJ_\psi(\{Age\}) = \{x_4\}$ and $OBJ_\psi(\{Age\}, \{Sal\}) = \emptyset$. Thus, τ' is not definable in ψ.

Now, we are ready to formulate the task of rule generation in the external mode. Please note how it depends on the resolution settings.

Specification of the rule generation task in the external mode
For α, β $(0 < \alpha, \beta \leq 1)$ and fixed resolutions γ_A of interval-attributes $A \in AT$:
(The lower system) Find each implication τ such that $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$ hold in each complete extension ψ, where τ is definable.
(The upper system) Find each implication τ such that $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$ hold in a complete extension ψ, where τ is definable.

Although in the above specification $support(\tau)$ could be taken into account also for extensions where τ is not definable, namely as $support(\tau) = 0$, we do not do it because it might lead to ignoring potentially meaningful implications.

Example 3. For Table 1, let us fix $\gamma_{Age} = 40$, $\gamma_{Sal} = 12000$, $\alpha = 0.3$ and $\beta = 0.5$. Then, $g_{Age}(x_4)$, $g_{Sal}(x_2)$, $g_{Sal}(x_3)$, $g_{Sal}(x_5)$ are indefinite intervals. Let us consider $\tau : [Age, [30, 70]] \Rightarrow [Sal, [10000, 22000]]$. In this case, $\{x_1, x_2, x_3, x_5\} \subseteq OBJ_\psi(\{Age\})$ for each ψ, and $x_4 \in OBJ_{\psi'}(\{Age\})$ for some ψ'. Given that $OBJ_\psi(\{Age\}, \{Sal\}) = \{x_1, x_2\}$ for each ψ, we have $support(\tau) \geq 2/5 > 0.3$ and $accuracy(\tau) \geq 2/4 \geq 0.5$. (If x_4 satisfies $[Age, [30, 70]]$, then we have $accuracy(\tau) = 0.6$.) Therefore, τ is a rule in the lower system. The lower system defines certain rules, and τ expresses certain information in Table 1.

5.2 External Apriori Algorithm

Algorithm 1 on the next page is analogous to *NIS-Apriori*. It works properly thanks to similar observations as those reported in Section 2 for *NISs*.

Definition 6. *Consider IID. We define sets inf and sup for each descriptor:*

(1) For a set-attribute $A \in AT$ and a descriptor $[A, val]$,
$inf([A, val]) = \{x \in OB |\, g_A(x) = \{val\}\}$,
$sup([A, val]) = \{x \in OB |\, val \in g_A(x)\}$.
(2) For an interval-attribute $A \in AT$ with a resolution γ_A and $[A, [l, u]]$,
$inf([A, [l, u]]) = \{x \in OB |\, g_A(x) \subseteq [l, u]\}$,
$sup([A, [l, u]]) = \{x \in OB |\, [l, u] \subseteq g_A(x)\}$.
(3) For a conjunction of descriptors $\wedge_i DESC_i$,
$inf(\wedge_i DESC_i) = \cap_i inf(DESC_i)$,
$sup(\wedge_i DESC_i) = \cap_i sup(DESC_i)$.

Algorithm 1. External Apriori Algorithm for Lower System

Input : An IID, a decision attribute DEC, threshold values α and β, as well as resolutions γ_A and descriptors $[A, [l_k, u_k]]$ for all interval-attributes.
Output: All rules defined by the lower system.

for (*each_attribute* $A \in AT$) **do**
  Generate $inf([A, val])$ and $sup([A, val])$
  or $inf([A, [l_k, u_k]])$ and $sup([A, [l_k, u_k]])$;
end

Generate set $CANDIDATE(1)$ with elements $DESC$ satisfying (A) or (B):

(A) $|inf(DESC)| \geq NUM$; // where $NUM = \lceil \alpha \cdot |OB| \rceil$
(B) $|inf(DESC)| = (NUM - 1)$ and $(sup(DESC) - inf(DESC)) \neq \emptyset$;

Generate set $CANDIDATE(2)$ according to the following procedures:

(Proc 2-1) For every $DESC$ and $DESC_{DEC}$ in $CANDIDATE(1)$
  generate conjunction of descriptors $DESC \wedge DESC_{DEC}$;
(Proc 2-2) For each generated conjunction, examine (A) and (B);
  If either (A) or (B) holds and $minacc(\tau) \geq \beta$
    display an implication τ as a rule;
  If either (A) or (B) holds and $minacc(\tau) < \beta$
    add this descriptor to $CANDIDATE(2)$;

Assign 2 to n;

while $CANDIDATE(n) \neq \emptyset$ **do**
  Generate $CANDIDATE(n+1)$ according to the following procedures:
    (Proc 3-1) For each matching pair in $CANDIDATE(n)$
      generate the corresponding longer conjunction;
    (Proc 3-2) Examine the same procedure as (Proc 2-2);
  Assign $n+1$ to n;
end

Due to Definition 6, it is possible to define two sets inf and sup over a set OB, and it is possible to apply *NIS-Apriori* algorithm. We can similarly define the algorithm for the upper system. These algorithms do not depend upon the number of complete extensions. We can calculate rules defined in the external mode even if there are uncountably many complete extensions.

6 Open Challenges: Internal Mode and Descriptors

In the internal mode, we employ descriptors $[A, SET]$ ($SET \subseteq VAL_A$) for a set-attribute and $[A, [l, u]]$ ($\gamma_A \leq (u - l)$) for an interval-attribute. We can consider the rule generation problem as follows:

Specification of the rule generation task in the internal mode
(The lower system) Find each implication τ such that $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$ hold in each extension ψ, where τ is definable.
(The upper system) Find each implication τ such that $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$ hold in an extension ψ, where τ is definable.

The external mode is defined only by complete extensions. On the other hand, the internal mode is defined by a much wider class of extensions, wherein the complete ones are just special cases. Investigation of the rule generation problems in the internal mode is one of our nearest future research items.

Another challenge is a selection of descriptors. In $NISs$, all attributes are finite set-attributes. Therefore, we employ finite descriptors $[A, val]$. In $IIDs$, we can similarly define finite descriptors for set-attributes. However, we need to introduce some additional assumptions for interval-attributes. For example:

(Assumption) There is a set of disjoint intervals $[l_i, u_i]$ satisfying $\cup_i [l_i, u_i] = VAL_A$. Furthermore, any interval $[l_s, u_t]$ is defined by $\cup_{i=s}^{t} [l_i, u_i]$. We call each $[l_i, u_i]$ an *atomic* interval.

The idea of atomic intervals looks like analogous to discretization of numerical data [3,4,5]. On the other hand, we would like to tend towards models that are more general than basic discretization. Appropriate selection of descriptors for interval-attributes is one more item on our future research roadmap.

7 Concluding Remarks

In this paper, we proposed how to formulate and solve the rule generation problem for Lipski's Incomplete Information Databases. We attempted to clarify all necessary details for both the external and the internal interpretations introduced in [8], although it is obvious that the internal interpretation requires far more study. In this way, we extended our previous research on Non-deterministic Information Systems towards a more general framework for dealing with information incompleteness in data.

One should remember that this is just a preliminary report. In particular, we need to keep working on the challenges outlined in Section 6. We need to proceed with experimental verification of the proposed algorithms. We also need to investigate how to extend our framework further towards more complex attribute types, for which the sets of values and intervals may be yet insufficient.

Acknowledgements. The first author was partially supported by the Grant-in-Aid for Scientific Research (C) (No.16500176, No.18500214), Japan Society for the Promotion of Science. The third author was partially supported by the grants N N516 368334 and N N516 077837 from the Ministry of Science and Higher Education of the Republic of Poland.

References

1. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: Proceedings of the 20th Very Large Data Base, pp. 487–499 (1994)
2. Ceglar, A., Roddick, J.F.: Association mining. ACM Comput. Surv. 38(2) (2006)
3. Greco, S., Matarazzo, B., Słowiński, R.: Granular Computing and Data Mining for Ordered Data: The Dominance-Based Rough Set Approach. Encyclopedia of Complexity and Systems Science, 4283–4305 (2009)
4. Grzymała-Busse, J.: Data with Missing Attribute Values: Generalization of Indiscernibility Relation and Rule Induction. Transactions on Rough Sets 1, 78–95 (2004)
5. Huynh, V.-N., Nakamori, Y., Ono, H., Lawry, J., Kreinovich, V., Nguyen, H. (eds.): Interval / Probabilistic Uncertainty and Non-Classical Logics. Advances in Soft Computing, vol. 46. Springer, Heidelberg (2008)
6. Infobright.org Forums, http://www.infobright.org/Forums/viewthread/288/, http://www.infobright.org/Forums/viewthread/621/
7. Kryszkiewicz, M.: Rules in Incomplete Information Systems. Information Sciences 113, 271–292 (1999)
8. Lipski, W.: On Semantic Issues Connected with Incomplete Information Data Base. ACM Trans. DBS 4, 269–296 (1979)
9. Lipski, W.: On Databases with Incomplete Information. Journal of the ACM 28, 41–70 (1981)
10. Orłowska, E., Pawlak, Z.: Representation of Nondeterministic Information. Theoretical Computer Science 29, 27–39 (1984)
11. Pawlak, Z.: Rough Sets. Kluwer Academic Publisher, Dordrecht (1991)
12. Sakai, H., Okuma, A.: Basic Algorithms and Tools for Rough Non-deterministic Information Analysis. Transactions on Rough Sets 1, 209–231 (2004)
13. Sakai, H., Ishibashi, R., Nakata, M.: On Rules and Apriori Algorithm in Non-deterministic Information Systems. Transactions on Rough Sets 9, 328–350 (2008)
14. Sakai, H., Hayashi, K., Nakata, M., Ślęzak, D.: The Lower System, the Upper System and Rules with Stability Factor in Non-deterministic Information Systems. In: Sakai, H., Chakraborty, M.K., Hassanien, A.E., Ślęzak, D., Zhu, W. (eds.) RSFDGrC 2009. LNCS (LNAI), vol. 5908, pp. 313–320. Springer, Heidelberg (2009)
15. Skowron, A., Rauszer, C.: The Discernibility Matrices and Functions in Information Systems. In: Intelligent Decision Support - Handbook of Advances and Applications of the Rough Set Theory, pp. 331–362. Kluwer Academic Publishers, Dordrecht (1992)
16. Ślęzak, D., Sakai, H.: Automatic Extraction of Decision Rules from Non-deterministic Data Systems: Theoretical Foundations and SQL-Based Implementation. In: Ślęzak, D., et al. (eds.) DTA 2009. CCIS, vol. 64, pp. 151–162. Springer, Heidelberg (2009)

A Fast Randomisation Test for Rule Significance*

Ivo Düntsch[1] and Günther Gediga[2]

[1] Dept of Computer Science, Brock University, St. Catherines, ON, Canada
duentsch@brocku.ca
[2] Dept of Psychology, Universität Münster, Fliednerstr. 21, D–48149 Münster
guenther@gediga.de

Abstract. Randomisation is a method to test the statistical significance of a symbolic rule; it is, however, very expensive. In this paper we present a sequential randomisation test which in most cases greatly reduces the number of steps needed for a conclusion.

1 Introduction

One problem of rule based data analysis is that the validity of a rule may be given, while its (statistical) significance is not. For example, if rules are based on a few observations only, the granularity of the system is too high, and the rule may be due to chance; in other words, a rule may be true, but useless. The significance problem does not seem to have received due attention in the rough set community.

In order to test the significance of rules, one can use randomisation methods [3] to compute the conditional probability of the rule, assuming that the null hypothesis

"Objects are randomly assigned to decision classes"

is true. These procedures seem to be particularly suitable to non-invasive techniques of data mining such as rough set data analysis, since randomisation tests do not assume that the available data is a representative sample. This assumption is a general problem of statistical data mining techniques; the reason for this is the huge state complexity of the space of possible rules, even when there is only a small number of features. However, a drawback of randomisation is its costliness, and it is of great value to have a less expensive procedure which has few model assumptions, and still gives us a reliable significance test.

In [2] we have developed two procedures, both based on randomisation techniques, which evaluate the significance of prediction rules obtained in rough set dependency analysis. In the present paper, we show how Wald's sequential probability ratio randomisation test [5] can be applied which is cheap and reliably determines the statistical significance of a rough set rule system.

* Equal authorship is implied. Ivo Düntsch gratefully acknowledges support from the Natural Sciences and Engineering Research Council of Canada. Günther Gediga is also adjunct professor in the Department of Computer Science, Brock University.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 386–391, 2010.

2 Rule Systems

We use the terminology of rough set data analysis [4], and briefly explain the basic concepts.

An *information system* is a tuple $\mathcal{I} = \langle U, \Omega, V_a \rangle_{a \in \Omega}$, where

1. U is a finite set of objects.
2. Ω is a finite set of mappings $a : U \to V_a$. Each $a \in \Omega$ is called an *attribute* or *feature*.

If $x \in U$ and $Q \subseteq \emptyset$, we denote by $Q(x)$ the feature vector of x determined by the attributes in Q. Each non-empty subset Q of Ω induces an equivalence relation θ_Q on U by

$$x \equiv_{\theta_Q} y \text{ iff } a(x) = a(y) \text{ for all } a \in Q,$$

i.e.

$$x \equiv_{\theta_Q} y \text{ iff } Q(x) = Q(y).$$

Objects which are in the same equivalence class cannot be distinguished with the knowledge of Q.

Equivalence relations θ_Q, θ_P are used to obtain rules in the following way: Let $Q \to P$ be the relation

$$\langle X, Y \rangle \in Q \to P \text{ iff } X \text{ is a class of } \theta_Q,\ Y \text{ is a class of } \theta_P, \text{ and } X \cap Y \neq \emptyset.$$

A pair $\langle X, Y \rangle \in Q \to P$ is called a Q,P – *rule* (or just a rule, if Q and P are understood) and usually written as $X \to Y$. By some abuse of language we shall also call $Q \to P$ a rule when there is no danger of confusion.

Each equivalence class X of θ_Q corresponds to a vector $\boldsymbol{X}$ of Q-features, and analogously for P. Thus, if the class X of θ_Q intersects exactly the classes $Y_1, \ldots, Y_n$ of θP, then we obtain the rule

$$\text{If } Q(y) = \boldsymbol{X}, \text{ then } P(y) = \boldsymbol{Y_1} \text{ or } \ldots \text{ or } P(y) = \boldsymbol{Y_n}. \tag{2.1}$$

A class X of θ_Q is called P – deterministic, if $n = 1$ in (2.1), i.e. if there is exactly one class Y of P which intersects, and thus contains, X. We define the *quality of an approximation* of a an attribute set Q with respect to an attribute set P by

$$\gamma(Q \to P) = \frac{|\bigcup\{X : X \text{ is a } P\text{–deterministic class of } \theta_Q\}|}{|U|}. \tag{2.2}$$

The statistic $\gamma(Q \to P)$ measures the relative frequency of correctly P – classified objects with the data provided by Q.

3 Randomisation

Suppose that $\emptyset \neq Q, P \subseteq \Omega$, and that we want to evaluate the statistical significance of the rule $Q \to P$. Let Σ be the set of all permutations of U, and $\sigma \in \Sigma$. We define new attribute functions a^σ by

$$a^\sigma(x) \stackrel{\text{def}}{=} \begin{cases} a(\sigma(x)), & \text{if } a \in Q, \\ a(x), & \text{otherwise.} \end{cases}$$

The resulting information system $\mathcal{I}_\sigma$ permutes the Q–columns according to σ, while leaving the P–columns constant; we let Q^σ be the result of the permutation in the Q–columns, and $\gamma(Q^\sigma \to P)$ be the approximation quality of the prediction of P by Q^σ in $\mathcal{I}_\sigma$.

The value

$$p(\gamma(Q \to P)|H_0) := \frac{|\{\gamma(Q^\sigma \to P) \geq \gamma(Q \to P) : \sigma \in \Sigma\}|}{|U|!} \tag{3.1}$$

now measures the significance of the observed approximation quality. If $p(\gamma(Q \to P)|H_0)$ is low, traditionally below 5%, then the rule $Q \to P$ is deemed significant, and the (statistical) hypothesis "$Q \to P$ is due to chance" can be rejected.

A simulation study done in [2] indicates that the randomisation procedure has a reasonable power if the rule structure of the attributes is known.

Since there are $|U|!$ permutations of U, we see from equation (3.1) that the computational cost of obtaining the significance is feasible only for small values of $|U|$. A fairly simple tool to shorten the processing time of the randomisation test is the adaptation of a sequential testing scheme to the given situation. Because this sequential testing scheme can be used as a general tool in randomisation analysis, we present the approach in a more general way.

Suppose that θ is a statistic with realizations θ_i, and a fixed realization θ_c. We can think of θ_c as $\gamma(Q \to P)$ and θ_i as $\gamma(Q^\sigma \to P)$. Recall that the statistic θ is called α – *significant*, if the true value $p(\theta \geq \theta_c|H_0)$ is smaller than α. Traditionally, $\alpha = 0.05$, and in this case, one speaks just of *significance*.

An evaluation of the hypothesis $\theta \geq \theta_c$ given the hypothesis H_0 can be done by using a sample of size n from the θ distribution, and counting the number k of θ_i for which $\theta_i \geq \theta_c$. The evaluation of $p(\theta \geq \theta_c|H_0)$ can now be done by the estimator $\hat{p}_n(\theta \geq \theta_c|H_0) = \frac{k}{n}$, and the comparison $\hat{p}_n(\theta \geq \theta_c|H_0) < \alpha$ will be performed to test the significance of the statistic. For this to work we have to assume that the simulation is asymptotically correct, i.e. that

$$\lim_{n\to\infty} \hat{p}_n(\theta \geq \theta_c|H_0) = p(\theta \geq \theta_c|H_0). \tag{3.2}$$

In order to find a quicker evaluation scheme of the significance, it should be noted that the results of the simulation k out of n can be described by a binomial distribution with parameter $p(\theta \geq \theta_c|H_0)$. The fit of the approximation of $\hat{p}_n(\theta \geq \theta_c|H_0)$ can be determined by the confidence interval of the binomial distribution.

In order to control the fit of the approximation more explicitly, we introduce another procedure within our significance testing scheme. Let

$$H_b : \ p(\theta \geq \theta_c | H_0)) \in [0, \alpha) \tag{3.3}$$

$$H_a : \ p(\theta \geq \theta_c | H_0)) \in [\alpha, 1] \tag{3.4}$$

be another pair of statistical hypotheses, which are strongly connected to the original ones: If H_b holds, we can conclude that the test is α–significant, if H_a holds, we conclude that it is not.

Because we want to do a finite approximation of the test procedure, we need to control the precision of the approximation; to this end, we define two additional error components:

1. r = probability that H_a is true, but H_b is the outcome of the approximative test.
2. s = probability that H_b is true, but H_a is the outcome of the approximative test.

The pair (r, s) is called the *precision* of the approximative test. To result in a good approximation, the values r, s should be small (e.g. $r = s = 0.05$); at any rate, we assume that $r + s \lesssim 1$, so that $\frac{s}{1-r} \lesssim \frac{1-s}{r}$, which will be needed below.

Using the Wald-procedure [5], we define the likelihood ratio

$$LQ(n) = \frac{\sup_{p \in [0,\alpha)} p^k (1-p)^{n-k}}{\sup_{p \in [\alpha,1]} p^k (1-p)^{n-k}}, \tag{3.5}$$

and we obtain the following approximative sequential testing scheme:

1. If
$$LQ(n) \lesssim \frac{s}{1-r},$$
then H_a is true with probability at most s.
2. If
$$LQ(n) \gtrsim \frac{1-s}{r},$$
then H_b is true with probability at most r.
3. Otherwise
$$\frac{s}{1-r} \leq LQ(n) \leq \frac{1-s}{r},$$
and no decision with precision (r, s) is possible. In this case, the simulation must continue.

With this procedure, which is implemented in our rough set engine GROBIAN[1] [1], the computational effort for the significance test can be greatly reduced in most cases.

[1] http://www.cosc.brocku.ca/~duentsch/grobian/grobian.html

4 Simulation Studies

In order to validate the procedure we conducted a small scale simulation study. We check 6 different situations:

Situation	n	description	granules / decision classes	γ
1	8	granules	11112233	0.500
		d-classes	11223344	
2	8	granules	11223344	1.000
		d-classes	11223344	
3	8	granules	12345566	1.000
		d-classes	11223344	
4	8	granules	12345677	1.000
		d-classes	11223344	
5	8	granules	12345678	1.000
		d-classes	11223344	
6	30	granules	11111111112222222222222223333	0.467
		d-classes	11111111112222222222222334444	

In situation 1 there is a moderate approximation quality using granules which consists of more than 1 observation. In situations 2 to 5 the approximation quality is perfect ($\gamma = 1$), but the size of the granules gets smaller from situation 2 (maximum size of the granules) to situation 5 (minimum size of the granules. Situation 6 shows a moderate approximation quality, but a larger sample size and granules which are larger than those in situation 1 to 5.

If we apply the sequential randomisation test (SRT) to these situations, we observe the following results (based upon 1000 simulations, $\alpha = r = s = 0.05$, and 10,000 as the maximum number of randomisations within the SRT procedure):

Situation	γ	p	expected number of randomisations
1	0.500	0.034	572
2	1.000	0.014	176
3	1.000	0.033	556
4	1.000	0.233	39
5	1.000	0.999	11
6	0.467	0.008	121

First of all, the situations 4 and 5 show no significant results, although $\gamma = 1$. This is due to fact the all granules are extremely small and this high approximation quality will be observed by random as well. The second observation is that the expected number of randomisations grows when p approaches α. An inspection of the decision rule of the SRT procedure shows us that this needs to be the case. Note that the maximum number of 10,000 randomisations is not needed at all in this situation – in contrast to simple randomisation tests, in which the precision of the test is based on the number of the simulations.

Whereas these simulations demonstrate the usefulness of the procedure when there is a substantial γ-value, it needs to be demonstrated that the procedure is

statistical valid as well. The SRT procedure should only be used, if the reproduced α value, given the "no dependence" assumption, is not greater than the nominal α. The test would be optimal if the reproduced α equals the nominal α. In order to check this for the SRT, we start with the same situation, but permute the values of the decision attribute before using the SRT. Obviously, we result in a random prediction situation. In this setting we count the number of decisions of the SRT against the "no dependence" assumption. The relative number of these decisions can be used as an estimator for the reproduced α value.

Situation	Expected γ	Reproduced α (given $\alpha = 0.05$)	Expected number of randomisations
1	0.070	0.024	23
2	0.140	0.008	18
3	0.576	0.032	180
4	0.798	0.001	16
5	1.000	0.000	11
6	0.070	0.040	253

The result shows that using the SRT procedure will not exceed the nominal α - the test seems to be valid. The higher the granularity of the granules, the higher is the expected value of γ (given no dependency) and the lower is the reproduced α of the SRT. Situation 6 demonstrates that the nominal α-value should be achieved when the granules consists of a reasonable number of observations.

References

1. Düntsch, I., Gediga, G.: The rough set engine GROBIAN. In: Sydow, A. (ed.) Proc. 15th IMACS World Congress, Berlin, vol. 4, pp. 613–618. Wissenschaft und Technik Verlag (1997)
2. Düntsch, I., Gediga, G.: Statistical evaluation of rough set dependency analysis. International Journal of Human–Computer Studies 46, 589–604 (1997)
3. Manly, B.F.J.: Randomization and Monte Carlo Methods in Biology. Chapman and Hall, Boca Raton (1997)
4. Pawlak, Z.: Rough sets. Internat. J. Comput. Inform. Sci. 11, 341–356 (1982)
5. Wald, A.: Sequential tests of statistical hypotheses. The Annals of Mathematical Statistics 16(2), 117–186 (1945)

Ordinal Classification with Monotonicity Constraints by Variable Consistency Bagging

Jerzy Błaszczyński[1], Roman Słowiński[1,2], and Jerzy Stefanowski[1]

[1] Institute of Computing Science, Poznań University of Technology, 60-965 Poznań, Poland
[2] Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland
{jurek.blaszczynski,roman.slowinski,jerzy.stefanowski}@cs.put.poznan.pl

Abstract. We propose an ensemble method that solves ordinal classification problem with monotonicity constraints. The classification data is structured using the Variable Consistency Dominance-based Rough Set Approach (VC-DRSA). The method employs a variable consistency bagging scheme to produce bootstrap samples that privilege objects (i.e., classification examples) with relatively high values of consistency measures used in VC-DRSA. In result, one obtains an ensemble of rule classifiers learned on bootstrap samples. Due to diversification of bootstrap samples controlled by consistency measures, the ensemble of classifiers gets more accurate, which has been acknowledged by a computational experiment on benchmark data.

1 Introduction

The paper concerns construction of rule classifiers for *ordinal classification problems with monotonicity constraints*. We propose an ensemble classification method that is based on a generalization of the bagging scheme, called *variable consistency bagging* (VC-bagging) [5]. In VC-bagging, the sampling of objects is controlled by so-called consistency measures that are also used in Variable-Consistency Dominance-based Rough Set Approach (VC-DRSA) [3]. VC-DRSA is, in turn, a generalization of the Dominance-based Rough Set Approach (DRSA) proposed by Greco, Matarazzo and Słowiński [11,14] for ordinal classification with monotonicity constraints. In ordinal classification, objects are described by attributes with ordered value sets; such attributes are called criteria.

Ordinal classification with monotonicity constraints means that classes are ordered and there exists a monotonic relationship between evaluation of an object on a criterion and its assignment to a class, such that if object a has evaluations on all considered criteria not worse than object b (i.e., a dominates b), then a is expected to be assigned to a class not worse than that of b. The last principle is called *dominance principle*. Objects violating the dominance principle are called *inconsistent*.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 392–401, 2010.

DRSA builds lower and upper approximations of unions of ordered classes using granules which are dominance cones in the space of considered criteria. In result, DRSA is structuring the ordinal classification data (classification examples). Lower approximations are used to induce certain decision rules. The lower approximations are composed of consistent objects only, which appears to be too restrictive in practical applications. Because real data concerning ordinal classification with monotonicity constraints are often strongly inconsistent, the lower approximations get easily empty, which prevents inducing certain decision rules from these approximations. Therefore, different versions of dominance-based lower approximations relaxed by probabilistic conditions were proposed (see [3] for review). In this paper, we rely on the version of VC-DRSA characterized in [3]. It makes use of so-called *object consistency measures* to quantify the evidence for membership of an object to a union of ordered classes. These measures permit to control the degree of consistency of objects admitted to the probabilistic lower approximations. These approximations provide positive examples for induction of decision rules which are basic components of a classifier.

In this paper, we propose to construct an ensemble composed of rule classifiers induced on bootstrap samples of objects (classification examples) controlled by consistency measures and structured using VC-DRSA. This approach extends the *bagging* scheme proposed by Breiman [7]. Let us remark that in the standard bagging, several classifiers, called component or base classifiers, are induced using the same learning algorithm over different distributions of input objects, which are bootstrap samples obtained by *uniform sampling* with replacement. Bagging has been extended in a number of ways in attempt to improve its the predictive accuracy. These extensions focused mainly on increasing diversity of component classifiers. *Random forest* [8], which is using attribute subset randomized decision tree component classifiers, is a well known example. Other extensions of bagging take advantage of random selection of attributes. In some cases, the random selection of attributes was combined with standard bootstrap sampling (see [13]).

The motivation of VC-bagging is also to increase diversity of component classifiers by changing the sampling phase. We take, however, into account the postulate saying that base classifiers used in bagging are expected to have sufficiently high predictive accuracy apart from being diversified [8]. Our hypothesis is that this requirement can be satisfied by privileging consistent objects when generating bootstrap samples. In our opinion, the inconsistent objects may lead to overfitting of the base classifies which decreases their classification accuracy.

Following our hypothesis, we change the standard bootstrap sampling, where each object is sampled with the same probability, into more focused sampling, where consistent objects are more likely to be selected than inconsistent ones. To identify consistent objects we use the same consistency measures as those used to define probabilistic lower approximations in VC-DRSA [5]. The supporting intuition is that decreasing a chance for selecting inconsistent objects should lead to constructing more accurate and more diversified base classifiers in the bagging scheme. In addition to [5], we also consider consistency of objects with respect to description of objects by a random subset of attributes (criteria), instead of the

whole set only. We considered this extension in [4], but as it was also the case in [5], only for non-ordinal classification problems. In consequence, the consistency will be measured in the dominance cones constructed on random subsets of the set of attributes. The motivation of this extension of VC-bagging is twofold. First, it introduces another level of randomization into method, which should lead to more diversified samples. Second, when consistent objects are identified on subsets of attributes, the intuition of sampling is the same as in the case of the whole set of attributes, but then it is expressed with respect to objects that can become basis for construction of classification patterns, e.g., decision rules.

Remark finally, that our bootstrap sampling controlled by consistency measures takes place in a pre-processing phase, before learning, similarly to data structuring by VC-DRSA, which takes place before induction of rules from probabilistic lower approximations. Moreover, in the way typical for bagging, consistency of objects is calculated independently for each of bootstrap samples.

The paper is organized as follows. In the next section, we recall basic elements of VC-DRSA. Then in the two following sections, we present the bagging scheme and its variable consistency extension (VC-bagging). In the following section the computational experiment is reported, and the last section groups conclusions.

2 Basic Elements of VC-DRSA

Ordinal data being structured by the Dominance-based Rough Set Approach (DRSA) [11,14] concern a finite universe U of objects described by a finite set of attributes A with ordered value sets. It has the form of a decision table, where rows correspond to objects from U and columns to attributes from A. Attributes with preference-ordered value sets are called *criteria*, while attributes whose value sets are not preference-ordered are called *regular attributes*. Moreover, A is divided into disjoint sets of condition attributes C and decision attributes D. The value set of attribute $q \in C \cup D$ is denoted by V_q, and $V_P = \prod_{q=1}^{|P|} V_q$ is called P-evaluation space, $P \subseteq C$. For simplicity, we assume that D is a singleton $D = \{d\}$, and values of d are ordered class labels coded by integers from 1 to n.

When among condition attributes from C there is at least one criterion, and there exists a monotonic relationship between evaluation of objects on criteria and their values (class labels) on the decision attribute, then the classification problem falls into the category of ordinal classification with monotonicity constraints. In order to make a meaningful representation of classification decisions, one has to consider the *dominance relation* in the evaluation space. For each object $y \in U$, two dominance cones are defined with respect to (w.r.t.) $P \subseteq C$. The P-positive dominance cone $D_P^+(y)$ is composed of objects that for each $q_i \in P$ are not worse than y. The P-negative dominance cone $D_P^-(y)$ is composed of objects that for each $q_i \in P$ are not better than y. The decision attribute makes a partition of objects from U into ordered decision classes $X_1, X_2, \ldots, X_n$, such that if $i < j$, then class X_i is considered to be worse than X_j. The dominance-based

approximations concern unions of decision classes: upward unions $X_i^{\geq} = \bigcup_{t \geq i} X_t$, where $i = 2, \ldots, n$, and downward unions $X_i^{\leq} = \bigcup_{t \leq i} X_t$, where $i = 1, \ldots, n-1$.

In order to simplify notation, we will use symbol X to denote a set of objects belonging to union of classes $X_i^{\geq}$ or $X_i^{\leq}$, unless it would lead to misunderstanding. Moreover, we will use $E_P(y)$ to denote any dominance cone $D_P^+(y)$ or $D_P^-(y)$, $y \in U$. If X and $E_P(y)$ are used in the same equation, then for X representing $X_i^{\geq}$ (resp. $X_i^{\leq}$), $E_P(y)$ stands for dominance cone $D_P^+(y)$ (resp. $D_P^-(y)$).

Variable-consistency (probabilistic) rough set approaches aim to extend lower approximation of set X by inclusion of objects with sufficient evidence for membership to X. This evidence can be quantified by *object consistency measures*. In [3], we distinguished gain-type and cost-type object consistency measures.

Let us give a generic definition of probabilistic P-lower approximation of set X. For $P \subseteq C, X \subseteq U, y \in U$, given a gain-type (resp. cost-type) object consistency measure $\Theta_X^P(y)$ and a gain-threshold (resp. cost-threshold) θ_X, the P-lower approximation of set X is defined as:

$$\underline{P}^{\theta_X}(X) = \{y \in X : \Theta_X^P(y) \propto \theta_X\}, \tag{1}$$

where $\propto$ denotes $\geq$ in case of a gain-type object consistency measure and a gain-threshold, or $\leq$ for a cost-type object consistency measure and a cost-threshold. In the above definition, $\theta_X \in [0, A_X]$ is a technical parameter influencing the degree of consistency of objects belonging to lower approximation of X.

In [3], we also introduced and motivated four *monotonicity properties* required from object consistency measures used in definition (1); they were denoted by $(m1)$, $(m2)$, $(m3)$, and $(m4)$. The object consistency measure that we consider in this paper is a cost-type measure $\epsilon_X^P(y)$. For $P \subseteq C, X, \neg X \subseteq U$, it is defined as:

$$\epsilon_X^P(y) = \frac{|E_P(y) \cap \neg X|}{|\neg X|}. \tag{2}$$

As proved in [3], this measure has properties $(m1)$, $(m2)$ and $(m4)$.

The probabilistic lower approximations of unions of decision classes are basis for induction of a set of decision rules. VC-DomLEM [6] algorithm can be applied to this end. It induces sets of probabilistic rules that preserve monotonicity constraints in a degree expressed by the same consistency measure as the one used to identify sufficiently consistent objects for the probabilistic lower approximations.

Once the set of rules has been constructed, it can be used to classify objects. Classification methods are used at this stage to solve situations when the classified object is covered by multiple rules that suggest assignment to different unions of classes. In the standard DRSA classification method, an object covered by a set of rules is assigned to a class (or a set of contiguous classes) resulting from intersection of unions of decision classes suggested by the rules. The new classification method proposed for DRSA and VC-DRSA in [2], is based on a notion of score coefficient associated with a set of rules covering an object and with classes suggested by these rules for the considered object. The score coefficient reflects relevance between rules and a particular class to which they assign

the object. A vector of values of score coefficients calculated for an object with respect to each class can be interpreted as a distribution of relevance between rules covering the object and the classes.

3 Bagging Scheme

Bagging was introduced by Breiman [7]. Its idea is quite simple. Bagging combines base classifiers generated by the same learning algorithm from different bootstrap samples of the input training set. The outputs of these classifiers are aggregated by an equal weight voting to make a final classification decision.

The diversity results from using different training samples. Each *bootstrap sample* is obtained by sampling objects uniformly with replacement. Each sample contains $n \leq |U|$ objects, however, some objects do not appear in it, while others may appear more than once. The same probability $1/n$ of being sampled is assigned to each object. The probability of an object being selected at least once is $1 - (1 - 1/n)^n$. For a large n, this is about $1 - 1/e$. Each bootstrap sample contains, on the average, 63.2% unique objects from the training set [7]. Bagging has one parameter m, which is the number of repetitions.

Bagging is a learning framework in which almost any learning algorithm can be used. Many experimental results show a significant improvement of the classification accuracy, in particular, using decision tree classifiers and rule classifiers. However, the choice of a base classifier is not indifferent. According to Breiman [7], what makes a base classifier suitable is its *unstability*, i.e., small changes in the training set causing major changes in the classifier.

4 Variable Consistency Bagging for Ordinal Classification with Monotonicity Constraints

The goal of variable consistency bagging (VC-bagging) is to increase predictive accuracy of bagged classifiers by using additional information about inconsistency of objects. The resulting bagged classifiers are trained on bootstrap samples slightly shifted towards more consistent objects [4,5].

The VC-bagging learning algorithm presented as Algorithm 1 is almost the same as the standard bagging scheme. The difference lies in consistency sampling, which is a modified procedure of bootstrap sampling on random subsets of attributes P of specified size $p = |P|$, line 3. This procedure is using consistency of object calculated on random subsets of attributes to construct more consistent bootstrap samples. The cardinality p of random subsets of attributes $P \subseteq C$ is limited by the size of the set of condition attributes (criteria) describing objects. This parameter controls the size of patterns that are identified by the consistency measures in the sampling procedure. It is worth noting that, random subsets of attributes are used only to calculate consistency of objects. Objects with complete description are drawn into bootstrap samples and then used during learning of component classifiers.

Algorithm 1. VC-bagging scheme for ordinal classification with monotonicity constraints

Input : LS training set; TS testing set; `OMCLA` learning algorithm that constructs ordinal classifiers preserving monotonicity constraints; Θ^P consistency measure; p number of attributes used in consistency sampling; m number of bootstrap samples;
Output: C^* final classifier

1 *Learning phase*;
2 **for** $i := 1$ *to* m **do**
3 $S_i :=$ `bootstrap sample of objects, which are drawn by consistency sampling from` LS `with measure` Θ^P `calculated on randomly selected set of attributes` P`, such that` $|P| = p$ {sample objects with replacement according to measure Θ^P } ;
4 $C_i :=$ `OMCLA` (S_i) {generate a base classifier} ;
5 *Classification phase*;
6 **foreach** y *in* TS **do**
7 $C^*(y) :=$ `combination of the responses of` $C_i(y)$`, where` $i = 1, \ldots, m$ {the suggestion of the classifier for object y is a combination of suggestions of component classifiers C_i} ;

To apply VC-bagging to ordinal classification with monotonicity constraints, the algorithm that constructs component classifiers needs to be an ordinal one that preserves monotonicity constraints. Moreover, this requirement also applies to the consistency measures used in bootstrap consistency sampling. It is thus possible to use measures defined in VC-DRSA. Consistency measures are used to tune the probability of object y being drawn to a bootstrap sample, e.g., by calculating a product of $\Theta_X^P(y)$ and $1/|U|$. The cost-type object consistency measures need to be transformed to gain-type (it can be done by subtracting the value of consistency measure from the highest value that it can take).

In the sampling, objects that are inconsistent on the selected random subset P have decreased probability of being sampled. Objects that are more consistent (i.e., have higher value of a consistency measure) are more likely to appear in the bootstrap sample. Different object consistency measures may result in different probability of inconsistent object y being sampled. The consistency measures that have property $(m1)$, i.e., that are monotonic with respect to the set of attributes, when are applied in consistency sampling on subsets of attributes P, they allow to identify consistent patterns of at least size p , such that $|P| = p$. The object consistency measures that do not have property $(m1)$ allow to identify consistent patterns of exactly size p.

The responses of component classifiers are combined in line 7 of the algorithm. When all responses indicate single class, in case of non-ordinal classification problem, majority voting is the method of combining responses in an ensemble [7]. This may be attributed to the fact that mode is the measure of central

tendency for non-ordinal nominal scale. Thus, majority voting minimizes the number of misclassifications. On the other hand, in case of ordinal classification problem, median of responses is the natural choice. This may be attributed to the fact that median is the measure of central tendency for ordinal scales. Median does not depend on a distance between class labels, so the scale of the decision attribute does not matter, only the order is taken into account. It minimizes the difference of ranks of the class to which the classified object belong and to which it is classified. Moreover, the responses indicating a set of contiguous classes (as it may be in case of the standard DRSA classifier), may be weighted according to the cardinality of the set of contiguous classes. Weighted median of responses is applied to combine such responses.

5 Experiments and Discussion

The first goal of the experiment was to check the predictive accuracy of the VC-bagging in ordinal classification with monotonicity constraints. To this end, we measured mean absolute error (MAE) on fourteen ordinal data sets listed in Table 1. We considered single monotonic VC-DomLEM[1] with the standard and the new classification methods. Results of these classifiers are used as a baseline for comparison of VC-DomLEM with the standard DRSA classification method used in bagging and in VC-bagging on random subsets of attributes with 50% cardinality. The cardinality of the random subset of attributes was chosen according to the results of our previous experiments with this type of ensembles [4]. We used ϵ measure in this type of ensemble because it has preferable properties and it is the same measure that is used by VC-DomLEM component classifiers. The choice of the standard DRSA classification method in the ensembles was made due to computational complexity of the new classification method. Moreover, we used for comparison two ordinal classifiers that preserve monotonicity constraints: Ordinal Learning Model (OLM) [1] and Ordinal Stochastic Dominance Learner (OSDL) [9]. The predictive accuracy was estimated by stratified 10-fold cross-validation, which was repeated several times. The results are shown in Table 2. The table contains values of MAE together with their standard deviations. Moreover, for each data set, we calculated a rank of the result of a classifier when compared to the other classifiers. The rank is presented in brackets (the smaller the rank, the better). Last row of the table shows the average rank obtained by a given classifier. The second aim of the experiment was to identify differences in bootstrap samples created by standard bagging and VC-bagging. These differences should (at least to some extent) transform to the differences of the component classifiers constructed by the two versions of bagging. The results of comparison of the bootstrap samples are presented in Table 3.

We applied Friedman test to globally compare performance of six different classifiers on multiple data sets [10]. The null-hypothesis in this test was that all compared classifiers perform equally well. We analyzed the ranks from Table 2. The p-value in Friedman test performed for this comparison was lower

[1] see http://www.cs.put.poznan.pl/jblaszczynski/Site/jRS.html

Table 1. Characteristics of data sets

Id	Data set	Objects	Attributes	Classes
1	balance	625	4	3
2	breast-c	286	8	2
3	breast-w	699	9	2
4	car	1296	6	4
5	cpu	209	6	4
6	bank-g	1411	16	2
7	fame	1328	10	5
8	denbosch	119	8	2
9	ERA	1000	4	9
10	ESL	488	4	9
11	housing	506	13	4
12	LEV	1000	4	5
13	SWD	1000	10	4
14	windsor	546	10	4

Table 2. MAE resulting from repeated 10-fold cross validation

Data set	VC-DomLEM std. class.	VC-DomLEM new. class.	bagging std. class.	VC-bagging std. class.	OLM	OSDL
balance	0.1621 (1.5) $\pm$0.001996	0.1621 (1.5) $\pm$0.001996	0.2011 (4) $\pm$0.003771	0.1973 (3) $\pm$0.01433	0.6384 (5) $\pm$0.01713	0.7003 (6) $\pm$0.004588
breast-c	0.2331 (1.5) $\pm$0.003297	0.2331 (1.5) $\pm$0.003297	0.2448 (3) $\pm$0.008565	0.2459 (4) $\pm$0.008722	0.324 (6) $\pm$0.01835	0.3065 (5) $\pm$0.001648
breast-w	0.03815 (4) $\pm$0.0006744	0.03720 (3) $\pm$0.002023	0.03577 (2) $\pm$0.001168	0.03243 (1) $\pm$0.001349	0.1764 (6) $\pm$0.00552	0.04149 (5) $\pm$0.001168
car	0.04090 (4) $\pm$0.00126	0.03421 (1) $\pm$0.0007275	0.03652 (2) $\pm$0.0007275	0.03832 (3) $\pm$0.002623	0.09156 (6) $\pm$0.005358	0.04141 (5) $\pm$0.0009624
cpu	0.1037 (4) $\pm$0.01846	0.08293 (2) $\pm$0.01479	0.08453 (3) $\pm$0.005968	0.07656 (1) $\pm$0.003907	0.3461 (6) $\pm$0.02744	0.3158 (5) $\pm$0.01034
bank-g	0.05481 (4) $\pm$0.001456	0.04536 (3) $\pm$0.001531	0.04489 (2) $\pm$0.001205	0.04158 (1) $\pm$0.001205	0.05528 (5) $\pm$0.001736	0.1545 (6) $\pm$0
fame	0.3803 (4) $\pm$0.001627	0.3406 (3) $\pm$0.001878	0.3230 (2) $\pm$0.006419	0.32 (1) $\pm$0.007993	1.577 (5) $\pm$0.03791	1.592 (6) $\pm$0.007555
denbosch	0.1261 (3) $\pm$0.006861	0.1232 (2) $\pm$0.01048	0.1289 (4) $\pm$0.01048	0.1092 (1) $\pm$0.006861	0.2633 (6) $\pm$0.02206	0.1541 (5) $\pm$0.003961
ERA	1.386 (5.5) $\pm$0.003682	1.386 (5.5) $\pm$0.003682	1.263 (1) $\pm$0.004497	1.271 (2) $\pm$0.002625	1.321 (4) $\pm$0.01027	1.280 (3) $\pm$0.00704
ESL	0.4447 (5) $\pm$0.01045	0.3702 (4) $\pm$0.01352	0.3477 (3) $\pm$0.006762	0.3374 (1) $\pm$0.004211	0.474 (6) $\pm$0.01114	0.3422 (2) $\pm$0.005019
housing	0.3564 (4) $\pm$0.008887	0.3235 (3) $\pm$0.01133	0.2984 (2) $\pm$0.002795	0.2793 (1) $\pm$0.00796	0.3867 (5) $\pm$0.01050	1.078 (6) $\pm$0.00796
LEV	0.4877 (5) $\pm$0.004497	0.4813 (4) $\pm$0.004028	0.4353 (3) $\pm$0.001700	0.409 (2) $\pm$0.003742	0.615 (6) $\pm$0.0099	0.4033 (1) $\pm$0.003091
SWD	0.462 (5) $\pm$0.003742	0.454 (4) $\pm$0.004320	0.443 (3) $\pm$0.003742	0.4297 (1) $\pm$0.002867	0.5707 (6) $\pm$0.007717	0.433 (2) $\pm$0.002160
windsor	0.5354 (5) $\pm$0.008236	0.5024 (1) $\pm$0.006226	0.5299 (4) $\pm$0.006743	0.5043 (2) $\pm$0.006044	0.5757 (6) $\pm$0.006044	0.5153 (3) $\pm$0.006044
average rank	3.96	2.75	2.71	1.71	5.57	4.29

than 0.0001. This result and observed differences in average ranks between the compared classifiers allow us to conclude that there is a significant difference between compared classifiers. Moreover, we checked significance of difference in predictive accuracy for each pair of classifiers. We applied to this end Wilcoxon test with null-hypothesis that the medians of results on all data sets of the two compared classifiers are equal. We observed significant difference (p-values lower

Table 3. Consistency and similarity of bootstrap samples created by standard bagging and by VC-bagging with ϵ calculated on subsets of attributes with 50% cardinality

	bagging			VC-bagging		
Id	% inconsistent	similarity all	similarity inconsistent	% inconsistent	similarity all	similarity inconsistent
1	100	0.7507	0.7507	100	0.4428	0.4426
2	93.05	0.7564	0.7561	91.76	0.7531	0.7506
3	15.93	0.7519	0.7492	9.77	0.7527	0.5808
4	67.01	0.7512	0.7508	56.29	0.7231	0.6489
5	55.17	0.7534	0.7541	53.49	0.7554	0.7429
6	6.38	0.7521	0.7492	3.34	0.7512	0.5472
7	59.47	0.7499	0.7502	57.99	0.7499	0.7422
8	39.6	0.7525	0.7573	3.76	0.7314	0.1431
9	100	0.7514	0.7514	100	0.7385	0.7387
10	99.25	0.7526	0.7526	99.05	0.7462	0.7463
11	33.23	0.7542	0.7554	14.76	0.7207	0.4745
12	100	0.7514	0.7514	100	0.6530	0.6532
13	99.89	0.7514	0.7514	99.87	0.7407	0.7409
14	92.53	0.7508	0.7507	89.58	0.7358	0.7295

than 0.05) between VC-bagging and any other classifier. These results allow us to state that VC-bagging with monotonic VC-DomLEM obtains the best results among compared classifiers. To our best knowledge, these results are also comparable to the results obtained by statistical ensembles of classifiers that solve ordinal classification with monotonicity constraints found in the literature [12].

Finally, we checked the similarity and consistency of bootstrap samples drawn by standard bagging and VC-bagging. The purpose of this analysis is to show differences between sampling used in the two versions of bagging. The average percentages of inconsistent objects in Table 3, indicate that samples used by VC-bagging are more consistent than those drawn in standard bagging. Similarity of bootstrap samples created by standard bagging is always close to 0.75, regardless of whether it is calculated for all objects or for inconsistent ones. We consider this result as a base line for our comparison. We can see that similarity measured for objects drawn in bootstrap samples created by VC-bagging is usually lower than in case of standard bagging. Moreover, for most of the data sets, similarity of inconsistent objects is even lower. These results are concordant with our analysis of consistency and similarity of bootstrap samples created by bagging and VC-bagging on non-ordinal data sets [4].

6 Conclusions

The main contribution of this paper is application of variable consistency bagging (VC-bagging) to ordinal classification problem with monotonicity constraints. The component classifiers in such bagging ensemble are composed of decision rules induced from bootstrap samples of objects structured using the Variable-Consistency Dominance-based Rough Set Approach (VC-DRSA). In VC-bagging, the generation of bootstrap samples is controlled by consistency measures which privilege objects being more consistent with respect to the dominance principle. The results of experiments indicate that VC-bagging improved the predictive

accuracy, i.e., reduced MAE, of rule classifiers induced from data structured by VC-DRSA. Comparison of consistency and similarity between samples drawn in VC-bagging and standard bagging shows that our proposal allows to construct bootstrap samples which are more consistent and more diversified, particularly with respect to inconsistent objects.

Acknowledgment

The authors wish to acknowledge financial support from the Ministry of Science and Higher Education, grantN N519 314435.

References

1. Ben-David, A., Sterling, L., Pao, Y.-H.: Learning and classification of monotonic ordinal concepts. Computational Intelligence 5(1), 45–49 (1989)
2. Błaszczyński, J., Greco, S., Słowiński, R.: Multi-criteria classification - a new scheme for application of dominance-based decision rules. European Journal of Operational Research 181(3), 1030–1044 (2007)
3. Błaszczyński, J., Greco, S., Słowiński, R., Szeląg, M.: Monotonic variable consistency rough set approaches. International Journal of Approximate Reasoning 50(7), 979–999 (2009)
4. Błaszczyński, J., Słowiński, R., Stefanowski, J.: Feature set-based consistency sampling in bagging ensembles. In: From Local Patterns To Global Models (LEGO), ECML/PKDD Workshop, pp. 19–35 (2009)
5. Błaszczyński, J., Słowiński, R., Stefanowski, J.: Variable consistency bagging ensembles. In: Peters, J.F., Skowron, A., Wolski, M., Chakraborty, M.K., Wu, W.-Z. (eds.) Transactions on Rough Sets X. LNCS, vol. 5656. Springer, Heidelberg (2009)
6. Błaszczyński, J., Słowiński, R., Szeląg, M.: Sequential covering rule induction algorithm for variable consistency rough set approaches. Submitted to Information Sciences (2009)
7. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
8. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
9. Cao-Van, K.: Supervised ranking - from semantics to algorithms. PhD thesis, Ghent University, CS Department (2003)
10. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
11. Greco, S., Matarazzo, B., Słowiński, R.: Rough sets theory for multicriteria decision analysis. European Journal of Operational Research 129(1), 1–47 (2001)
12. Kotłowski, W., Słowiński, R.: Rule learning with monotonicity constraints. In: Proceedings of ICML, pp. 537–544 (2009)
13. Patrice, L., Olivier, D., Christine, D.: Different ways of weakening decision trees and their impact on classification accuracy of DT combination. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 200–209. Springer, Heidelberg (2000)
14. Słowiński, R., Greco, S., Matarazzo, B.: Rough sets in decision making. In: Meyers, R.A. (ed.) Encyclopedia of Complexity and Systems Science, pp. 7753–7786. Springer, New York (2009)

Learnability in Rough Set Approaches

Jerzy Błaszczyński[1], Roman Słowiński[1,2], and Marcin Szeląg[1]

[1] Institute of Computing Science, Poznań University of Technology,
60-965 Poznań, Poland
{jblaszczynski,rslowinski,mszelag}@cs.put.poznan.pl
[2] Institute for Systems Research, Polish Academy of Sciences,
01-447 Warsaw, Poland

Abstract. We consider learning abilities of classifiers learned from data structured by rough set approaches into lower approximations of considered sets of objects. We introduce two measures, λ and δ, that estimate attainable predictive accuracy of rough-set-based classifiers. To check the usefulness of the estimates for various types of classifiers, we perform a computational experiment on fourteen data sets. In the experiment, we use two versions of the rough-set-based rule classifier, called VC-DomLEM, and few other well known classifiers. The results show that both introduced measures are useful for an a priori identification of data sets that are hard to learn by all classifiers.

1 Introduction

Rough set analysis of data is a step preceding the learning of a classifier. It checks the data for possible inconsistencies by calculation of lower approximations of considered sets of objects. Due to this type of data structuring, one may restrict a priori the set of objects on which the classifier is learned to a subset of sufficiently consistent objects belonging to lower approximations. This restriction is motivated by a postulate for learning from consistent data, so that the knowledge gained from this learning is relatively certain.

The original Rough Set Approach proposed by Pawlak [9] deals with classification data which are not considered to be ordered. The basic relation defining elementary granules of knowledge is an indiscernibility relation, and the sets approximated using these granules are decision classes. This is why we call the original approach Indiscernibility-based Rough Set Approach (IRSA). *Ordinal classification with monotonicity constraints* requires, however, another basic relation in order to handle the ordered domains of attributes and a monotonic relationship between evaluations of objects on the attributes and the assignment of these objects to ordered decision classes. Greco, Matarazzo and Słowiński [8,10] proposed to this end a dominance relation. In their approach, called Dominance-based Rough Set Approach (DRSA), the elementary granules of knowledge are dominance cones, and the sets approximated using these granules are upward and downward unions of ordered decision classes.

In this paper, we evaluate accuracy of prediction of various classifiers. To this end, we use two predictive accuracy measures. The first one is the percentage of

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 402–411, 2010.

correctly classified objects (PCC). The second one is the mean absolute difference between index of the class to which an object is assigned by a classifier and index of the class to which the object belongs. This measure, called mean absolute error (MAE), makes sense when the classes are ordered.

The interesting question is how to determine if the actual predictive accuracy obtained by a given classifier on a given data set is satisfactory. Obviously, one may compare this actual predictive accuracy with those of other classifiers. However, such an approach allows only for relative performance evaluation. It does not show if the classifier has learned from given data as much as it could have learned in the best case. We claim that the actual predictive accuracy of a classifier on a particular data set depends on at least three factors. First, on the complexity of the data (i.e., how large is the number of attributes and classes). Second, on the number of available training objects. Third, on the amount of inconsistencies observed in the data. In the rough set approaches, construction of a classifier is preceded by data structuring aiming at restricting the sets of objects on which the classifier is learned to lower approximations of these sets, composed of sufficiently consistent objects. This implies that the classifier will learn classification patterns on sufficiently consistent objects only, and we do not expect that it will classify correctly inconsistent objects.

In this paper, we concentrate on the estimation of the attainable predictive accuracy of a rough-set-based classifier. This estimation is performed before learning, taking into account inconsistencies detected in the data. Thus, it considers only the third factor, from those that have impact on the actual predictive accuracy. We propose two measures for this estimation. The first one, called λ, estimates attainable PCC of a classifier. The second one, called δ, estimates attainable MAE of a classifier. Obviously, measure δ can be applied only when decision classes are ordered, i.e., in DRSA. Our motivation for introducing the two measures is twofold. First, we observed that in the context of DRSA, the well-known γ measure, called quality of (approximation of) classification [8,10], is not well suited to the estimation of attainable PCC. It can be said that γ is a pessimistic estimate. In case of (highly) inconsistent data sets, value of γ tends to be relatively low (or even zero) and does not correlate with the PCC obtained by a learned classifier (see [7]). Second, we want to estimate attainable predictive accuracy of a classifier assigning objects to ordered decision classes. In such a case, it is important to minimize the difference between the true decision class of an object and the decision class it is assigned to. Even though the proposed measures are designed with concern for rough set classifiers, they can be also useful for classifiers that do not benefit from data structuring by rough set approaches. The inconsistencies in the data, even if they are not detected before learning, may affect the actual predictive accuracy of such classifiers.

We analyze the dependency between values of measures λ and δ calculated on the whole data sets, and average values of the corresponding predictive accuracy measures obtained by various classifiers in k-fold cross-validation experiments. The proposed measures are designed to be optimistic estimates of the attainable predictive accuracy of a classifier that is learned on lower approximations

of considered sets of objects. Actual predictive accuracy of such a classifier is supposed not to exceed the value of λ and δ, unless it uses some "tricks", like, for example, default assignment of unclassified objects to majority decision class. On the other hand, actual predictive accuracy worse than values of λ and δ indicate a weakness of the classifier and/or high complexity of the considered data set.

The concept of learnability considered in this paper concerns predictive accuracy that may be attained by a rough-set-based classifier. It is different from learnability of decision tables introduced by Ziarko [11], which addresses the effective ability to converge to a stable state in a process of incremental learning.

This paper is organized as follows. In Section 2, we remind basic definitions of rough set approaches. In Section 3, we introduce measures which can be used to estimate attainable predictive accuracy of a classifier learned in a rough set framework. Section 4 contains results of a computational experiment in which we compared different classifiers and showed how actual predictive accuracy of these classifiers relates to the values of estimation measures introduced in Section 3. Section 5 concludes the paper.

2 Basic Definitions of Rough Set Approaches

In the rough set approaches [8,9,10], data is presented as a decision table, where rows correspond to objects from universe U and columns correspond to attributes from a finite set A. Attributes with preference-ordered value sets are called *criteria*, while attributes without preference-ordered value sets are called *regular attributes*. Moreover, set A is divided into disjoint sets of condition attributes C and decision attributes D. The value set of attribute $q \in C \cup D$ is denoted by V_q, and $V_P = \prod_{q=1}^{|P|} V_q$ is called P-evaluation space, where $P \subseteq C$. For simplicity, we assume that $D = \{d\}$, and that values of d are class labels. Decision attribute d makes a partition of set U into a finite number of n disjoint sets of objects, called *decision classes*. We denote this partition by $\mathcal{X} = \{X_1, \ldots, X_n\}$. Decision about classification of object $y \in U$ to set X_i depends on its class label known from the decision table and on its relation with other objects from the table. In Indiscernibility-based Rough Set Approach (IRSA), the considered relation is the *indiscernibility relation* in the evaluation space [9]. Consideration of this relation is meaningful when set of attributes A is composed of regular attributes only. Indiscernibility relation makes a partition of universe U into disjoint blocks of objects that have the same description and are considered indiscernible. Such blocks are called *granules*. Moreover, $I_P(y)$ denotes a set of objects indiscernible with object y using set of attributes $P \subseteq C$. It is called a granule of P-indiscernible objects.

When among condition attributes from C there is at least one criterion, decision attribute d has preference-ordered value set, and there exists a monotonic relationship between evaluation of objects on criteria and their values (class labels) on the decision attribute, then, in order to make a meaningful representation of classification decisions, one has to consider the *dominance relation* in the

evaluation space. It has been proposed in [8,10] and the resulting approach was called Dominance-based Rough Set Approach (DRSA). For each object $y \in U$, two *dominance cones* are defined with respect to (w.r.t.) $P \subseteq C$. The P-positive dominance cone $D_P^+(y)$ is composed of objects that for each $q_i \in P$ are not worse than y. The P-negative dominance cone $D_P^-(y)$ is composed of objects that for each $q_i \in P$ are not better than y.

While in IRSA, decision classes $X_i \subseteq U$, $i = 1, \ldots, n$, are not ordered, in DRSA, they are ordered, such that if $i < j$, then class X_i is considered to be worse than X_j. In IRSA, approximations concerns decision classes X_i. In order to handle preference orders, and monotonic relationship between evaluations on criteria and assignment to decision classes, approximations made in DRSA concern the following *unions of decision classes*: upward unions $X_i^{\geq} = \bigcup_{t \geq i} X_t$, where $i = 2, 3, \ldots, n$, and downward unions $X_i^{\leq} = \bigcup_{t \leq i} X_t$, where $i = 1, 2, \ldots, n-1$.

In order to simplify notation, we will use (unless it would lead to misunderstanding) symbol X to denote a set of objects belonging to class X_i, in the context of IRSA, or to union of classes $X_i^{\geq}$, $X_i^{\leq}$, in the context of DRSA. Moreover, we will use symbol $E_P(y)$ to denote any granule $I_P(y)$, $D_P^+(y)$ or $D_P^-(y)$, $y \in U$. If X and $E_P(y)$ are used in the same equation, then for X representing class X_i, $E_P(y)$ denotes granule $I_P(y)$ and for X representing union of ordered classes $X_i^{\geq}$ (resp. $X_i^{\leq}$), $E_P(y)$ stands for dominance cone $D_P^+(y)$ (resp. $D_P^-(y)$).

In IRSA and DRSA, the P-lower approximation of set X, for $P \subseteq C$, $X \subseteq U$, $y \in U$, is defined as:

$$\underline{P}(X) = \{y \in X : E_P(y) \subseteq X\}. \tag{1}$$

This definition of the lower approximation appears to be too restrictive in practical applications. In consequence, lower approximations of sets are often empty, preventing generalization of data in terms of relative certainty. Therefore, various *probabilistic rough set approaches* were proposed which extend the lower approximation of set X by inclusion of objects with sufficient evidence for membership to X (see [4] for review). Probabilistic rough set approaches employing indiscernibility relation are called Variable Consistency Indiscernibility-based Rough Set Approaches (VC-IRSA), while probabilistic rough set approaches employing dominance relation are called Variable Consistency Dominance-based Rough Set Approaches (VC-DRSA). The evidence for membership to set X can be quantified by different *object consistency measures* (see [4] for review).

In VC-IRSA and VC-DRSA, probabilistic P-lower approximation of set X, for $P \subseteq C$, $X \subseteq U$, $y \in U$, given a gain-type (resp. cost-type) object consistency measure $\Theta_X^P(y)$ and a gain-threshold (resp. cost-threshold) θ_X, is defined as:

$$\underline{P}^{\theta_X}(X) = \{y \in X : \Theta_X^P(y) \propto \theta_X\}, \tag{2}$$

where $\propto$ denotes $\geq$ in case of a gain-type object consistency measure and a gain-threshold, or $\leq$ for a cost-type object consistency measure and a cost-threshold. In the above definition, $\theta_X \in [0, A_X]$ is a technical parameter influencing the degree of consistency of objects belonging to the lower approximation of X.

Let us observe that the probabilistic P-lower approximation of set X defined according to (2) is a superset of the P-lower approximation defined according to (1). Moreover, given any object consistency measure $\Theta_X^P(y)$, for the most restrictive value of threshold θ_X, $\underline{P}^{\theta_X}(X)$ is the same as $\underline{P}(X)$. Therefore, in the following, we will use more general notation of VC-IRSA and VC-DRSA, having in mind that introduced definitions are also applicable in IRSA and DRSA.

Let us remind definitions of positive, negative and boundary regions of X in the evaluation space ([3]). First, let us note that each set X has its complement $\neg X = U - X$. The P-positive region of X, for $P \subseteq C$, $X \subseteq U$, is defined as:

$$POS_P^{\theta_X}(X) = \bigcup_{y \in \underline{P}^{\theta_X}(X)} E_P(y), \tag{3}$$

where θ_X comes from (2). One can observe that $POS_P^{\theta_X}(X)$ extends $\underline{P}^{\theta_X}(X)$ by inclusion of some "inevitable" inconsistent objects. Moreover, in case of IRSA and DRSA, $POS_P^{\theta_X}(X)$ boils down to $\underline{P}^{\theta_X}(X)$. Basing on definition (3), we can define P-negative and P-boundary regions of the approximated sets:

$$NEG_P^{\theta_X}(X) = POS_P^{\theta_X}(\neg X) - POS_P^{\theta_X}(X), \tag{4}$$

$$BND_P^{\theta_X}(X) = U - POS_P^{\theta_X}(X) - NEG_P^{\theta_X}(X). \tag{5}$$

3 Estimation of Attainable Predictive Accuracy

In this section, we introduce two measures that estimate attainable predictive accuracy of a classifier learned on (probabilistic) P-lower approximations of considered sets of objects (i.e., classes in IRSA or unions of ordered classes in DRSA).

In (VC-)IRSA, a classifier learned on P-lower approximations *may* correctly assign object $y \in X_i$ to class X_i if y belongs to the P-positive region of X_i. Measure λ that estimates the ratio of objects in the data table that may be learned by the classifier is defined as:

$$\lambda_P^{\boldsymbol{\theta}_X}(\mathcal{X}) = \frac{\bigcup_{i=1}^{n} \left| X_i \cap POS_P^{\theta_{X_i}}(X_i) \right|}{|U|}, \tag{6}$$

where $P \subseteq C$, $\boldsymbol{\theta}_X = \{\theta_{X_1}, \ldots, \theta_{X_n}\}$.

Since in the context of (VC-)IRSA, $X_i \cap POS_P^{\theta_{X_i}}(X_i) = \underline{P}^{\theta_{X_i}}(X_i)$, one can observe that in this context measure λ boils down to the quality of approximation of classification $\mathcal{X}$ by set of attributes P [8,10], denoted by $\gamma_P^{\boldsymbol{\theta}_X}(\mathcal{X})$.

In (VC-)DRSA, a classifier learned on P-lower approximations *may* correctly assign object $y \in X_i$ to class X_i if $y \in POS_P^{\theta_{X_i^{\geq}}}(X_i^{\geq})$ or $y \in POS_P^{\theta_{X_i^{\leq}}}(X_i^{\leq})$. Measure λ that estimates the ratio of objects in the data table that may be learned by the classifier is defined as:

$$\lambda_P^{\boldsymbol{\theta}_X}(\mathcal{X}) = \frac{\left|X_1 \cap POS_P^{\theta_{X_1^{\leq}}}(X_1^{\leq})\right|}{|U|} + \tag{7}$$

$$+ \frac{\bigcup_{i=2}^{n-1}\left|X_i \cap (POS_P^{\theta_{X_i^{\geq}}}(X_i^{\geq}) \cup POS_P^{\theta_{X_i^{\leq}}}(X_i^{\leq}))\right|}{|U|} + \frac{\left|X_n \cap POS_P^{\theta_{X_n^{\geq}}}(X_n^{\geq})\right|}{|U|},$$

where $P \subseteq C$, $\boldsymbol{\theta}_X = \{\theta_{X_1^{\leq}}, \ldots, \theta_{X_{n-1}^{\leq}}, \theta_{X_2^{\geq}}, \ldots, \theta_{X_n^{\geq}}\}$.

Let us observe that $X_1 \cap POS_P^{\theta_{X_1^{\leq}}}(X_1^{\leq})$ may be written as $\underline{P}^{\theta_{X_1^{\leq}}}(X_1^{\leq})$, $X_n \cap POS_P^{\theta_{X_n^{\geq}}}(X_n^{\geq})$ may be written as $\underline{P}^{\theta_{X_n^{\geq}}}(X_n^{\geq})$, and $X_i \cap POS_P^{\theta_{X_i^{\geq}}}(X_i^{\geq})$ may be simplified as $X_i \cap \underline{P}^{\theta_{X_i^{\geq}}}(X_i^{\geq})$. Moreover, in (VC-)DRSA measure λ does not boil down to $\gamma_P^{\boldsymbol{\theta}_X}(\mathcal{X})$. It can be also shown that $\lambda_P^{\boldsymbol{\theta}_X}(\mathcal{X}) \geq \gamma_P^{\boldsymbol{\theta}_X}(\mathcal{X})$. In fact, $\gamma_P^{\boldsymbol{\theta}_X}(\mathcal{X})$ treats inconsistencies in the data very restrictively – each object $y \in X_i$ that does not belong to $\underline{P}^{\theta_{X_i^{\geq}}}(X_i^{\geq})$ or does not belong to $\underline{P}^{\theta_{X_i^{\leq}}}(X_i^{\leq})$ decreases the value of this measure. On the other hand, measure $\lambda_P^{\boldsymbol{\theta}_X}(\mathcal{X})$ decreases if object $y \in X_i$ belongs neither to $\underline{P}^{\theta_{X_i^{\geq}}}(X_i^{\geq})$ nor to $\underline{P}^{\theta_{X_i^{\leq}}}(X_i^{\leq})$.

In (VC-)DRSA, a classifier learned on P-lower approximations *may* assign object $y \in X_i$ to class X_k if y belongs to the P-positive region of $X_k^{\geq}$ or $X_k^{\leq}$. Measure δ that estimates the average minimal absolute difference between index of the class to which an object may be assigned and index of the class to which the object belongs, for $i : y_j \in X_i$, is defined as:

$$\delta_P^{\boldsymbol{\theta}_X}(\mathcal{X}) = \frac{1}{|U|}\sum_{j=1}^{|U|} \min_{k\,:\,y_j \in POS_P^{\theta_{X_k^{\geq}}}(X_k^{\geq}) \vee y_j \in POS_P^{\theta_{X_k^{\leq}}}(X_k^{\leq})} |i-k|, \tag{8}$$

where $P \subseteq C$, $\boldsymbol{\theta}_X = \{\theta_{X_1^{\leq}}, \ldots, \theta_{X_{n-1}^{\leq}}, \theta_{X_2^{\geq}}, \ldots, \theta_{X_n^{\geq}}\}$.

4 Results of the Computational Experiment

We considered rough-set-based classifier called VC-DomLEM[1] [5] in two variants: monotonic (i.e., with consistency measure ϵ [4]) and non-monotonic (i.e., with consistency measure μ' [3]). Moreover, we used ordinal classifiers that preserve monotonicity constraints: Ordinal Learning Model (OLM) [2] and Ordinal Stochastic Dominance Learner (OSDL) [6]. We also used well known non-ordinal classifiers: Naive Bayes, SVM with linear kernel, RIPPER, and C4.5.

The aim of the experiment was to compare actual predictive accuracy of considered classifiers with the proposed estimates of attainable predictive accuracy calculated before learning. We measured the percentage of correctly classified objects (PCC) and mean absolute error (MAE) on fourteen ordinal data sets listed in Table 1. Data sets: ERA, ESL, LEV and SWD were taken from [1]. Other data sets come from the UCI repository[2] and other public repositories.

[1] See http://www.cs.put.poznan.pl/jblaszczynski/Site/jRS.html
[2] See http://www.ics.uci.edu/~mlearn/MLRepository.html

Table 1. Characteristics of data sets

Data set	#Objects	#Attributes	#Classes
balance	625	4	3
bank-g	1411	16	2
breast-c	286	8	2
breast-w	699	9	2
car	1296	6	4
cpu	209	6	4
denbosch	119	8	2
ERA	1000	4	9
ESL	488	4	9
fame	1328	10	5
housing	506	13	4
LEV	1000	4	5
SWD	1000	10	4
windsor	546	10	4

In Table 2, we show the values of: quality of classification γ, measure λ (7), and measure δ (8), calculated on the whole data sets. For each measure, we present values for the most restrictive VC-DRSA consistency thresholds (i.e., for $\boldsymbol{\theta}^*_X = \boldsymbol{\epsilon}^*_X = \mathbf{0}$, $\boldsymbol{\theta}^*_X = \boldsymbol{\mu}^{'*}_X = \mathbf{1}$), and values calculated for the VC-DRSA consistency thresholds ϵ_X, μ'_X used during learning of VC-DomLEM classifiers. All these values can be compared to PCC and MAE achieved by the classifiers.

Table 2. Values of γ, λ, and δ measures for $\boldsymbol{\theta}^*_X = \boldsymbol{\epsilon}^*_X = \mathbf{0}$, $\boldsymbol{\theta}^*_X = \boldsymbol{\mu}^{'*}_X = \mathbf{1}$, as well as for $\boldsymbol{\epsilon}_X$ and $\boldsymbol{\mu}'_X$ used to obtain VC-DomLEM results shown in Tables 3 & 4

Data set	$\gamma_C^{\boldsymbol{\theta}^*_X}(\mathcal{X})$	$\lambda_C^{\boldsymbol{\theta}^*_X}(\mathcal{X})$	$\delta_C^{\boldsymbol{\theta}^*_X}(\mathcal{X})$	ϵ_X	$\gamma_C^{\boldsymbol{\epsilon}_X}(\mathcal{X})$	$\lambda_C^{\boldsymbol{\epsilon}_X}(\mathcal{X})$	$\delta_C^{\boldsymbol{\epsilon}_X}(\mathcal{X})$	μ'_X	$\gamma_C^{\boldsymbol{\mu}'_X}(\mathcal{X})$	$\lambda_C^{\boldsymbol{\mu}'_X}(\mathcal{X})$	$\delta_C^{\boldsymbol{\mu}'_X}(\mathcal{X})$
balance	100	100	0	0.01	100	100	0	0.99	100	100	0
bank-g	98.02	98.02	0.0198	0.001	98.87	98.87	0.0113	0.99	98.72	98.72	0.0128
breast-c	23.78	23.78	0.7622	0.45	98.6	98.6	0.014	0.55	90.21	90.21	0.0979
breast-w	97.57	97.57	0.0243	0.001	97.57	97.57	0.0243	0.95	100	100	0
car	97.22	98.61	0.0162	0.01	99.46	99.46	0.0054	0.85	100	100	0
cpu	100	100	0	0.001	100	100	0	0.99	100	100	0
denbosch	89.92	89.92	0.1008	0.05	99.16	99.16	0.0084	0.9	100	100	0
ERA	0	11.3	2.826	0.025	11.3	80.8	0.28	0.75	23.9	87.3	0.129
ESL	18.24	85.04	0.1578	0.025	62.09	100	0	0.95	77.46	98.98	0.0102
fame	89.38	98.27	0.0211	0.001	90.21	99.17	0.0113	0.6	100	100	0
housing	100	100	0	0.01	100	100	0	0.99	100	100	0
LEV	0.7	41.2	0.801	0.025	54	97.7	0.023	0.9	44.6	88.7	0.113
SWD	1.8	48.7	0.68	0.15	96.5	100	0	0.85	53.9	80.4	0.196
windsor	34.8	69.6	0.4066	0.05	87.55	97.44	0.0256	0.9	71.98	80.04	0.1996

The values of λ and δ in Table 2 show the attainable predictive accuracy of a rough-set-based classifier. Thus, they also show the consistency of analyzed data sets. The values of γ are always lower than the values of λ. Basing on values of λ and δ, we can identify three fully consistent data sets: balance, cpu, and housing. Then, we can distinguish four data sets that have high consistency: breast-w, car, bank-g, and fame. Also not bad in terms of consistency are: denbosch and ESL. Data sets: breast-c, ERA, LEV, SWD, and windsor are highly inconsistent. We can also observe that application of VC-DRSA led to

considerable improvement of both measures for inconsistent data sets. Thus, VC-DRSA allowed to include fair amount of inconsistent objects into extended lower approximations.

The actual predictive accuracy was calculated by stratified 10-fold cross-validation repeated several times. The results are shown in Tables 3 & 4. Both tables contain values of the actual predictive accuracy and their standard deviations. For each data set, the best value of the actual predictive accuracy, and values included in the standard deviation of the best one, are marked in bold. Actual predictive accuracies obtained by VC-DomLEM are at least comparable to those of other classifiers (which is concordant with the results from [5]).

Table 3. Percentage of correctly classified objects (PCC)

Data set	$\lambda_C^{\theta_X}(\mathcal{X})$	monotonic VC-DomLEM	non-monotonic VC-DomLEM	Naive Bayes	SVM	RIPPER	C4.5	OLM	OSDL
balance	100	86.61 $\pm$0.5891	86.93 $\pm$0.3771	**90.56** $\pm$0.1306	87.47 $\pm$0.1508	81.5 $\pm$0.5439	78.45 $\pm$0.7195	61.28 $\pm$1.287	57.81 $\pm$0.3288
bank-g	98.02	**95.46** $\pm$0.1531	95.13 $\pm$0.0884	88.54 $\pm$1.371	87.2 $\pm$0.1205	95.11 $\pm$0.352	94.85 $\pm$0.5251	94.47 $\pm$0.1736	84.55 $\pm$0
breast-c	23.78	**76.69** $\pm$0.3297	75.64 $\pm$0.7185	74.36 $\pm$0.5943	67.83 $\pm$1.244	70.4 $\pm$1.154	75.76 $\pm$0.3297	67.6 $\pm$1.835	69.35 $\pm$0.1648
breast-w	97.57	96.28 $\pm$0.2023	95.42 $\pm$0.3504	96.04 $\pm$0.06744	**96.76** $\pm$0.06744	95.52 $\pm$0.4721	94.47 $\pm$0.751	82.36 $\pm$0.552	95.85 $\pm$0.1168
car	98.61	**97.15** $\pm$0.063	**97.1** $\pm$0.1311	84.72 $\pm$0.1667	92.18 $\pm$0.2025	84.41 $\pm$1.309	89.84 $\pm$0.1819	91.72 $\pm$0.4425	96.53 $\pm$0.063
cpu	100	**91.7** $\pm$1.479	**90.75** $\pm$1.579	83.41 $\pm$0.9832	56.62 $\pm$1.579	84.69 $\pm$1.409	88.52 $\pm$1.409	68.58 $\pm$2.772	72.41 $\pm$1.479
denbosch	89.92	**87.68** $\pm$1.048	**87.11** $\pm$1.428	**87.11** $\pm$1.428	78.71 $\pm$0.3961	82.63 $\pm$2.598	83.47 $\pm$1.048	73.67 $\pm$2.206	84.6 $\pm$0.3961
ERA	11.30	26.9 $\pm$0.3742	22.17 $\pm$0.1247	25.03 $\pm$0.2494	24.27 $\pm$0.2494	20 $\pm$0.4243	**27.83** $\pm$0.4028	23.97 $\pm$0.4643	24.7 $\pm$0.8165
ESL	85.04	66.73 $\pm$1.256	62.43 $\pm$1.139	67.49 $\pm$0.3483	62.7 $\pm$0.6693	61.61 $\pm$1.555	66.33 $\pm$0.6966	55.46 $\pm$0.7545	**68.3** $\pm$0.3483
fame	98.27	**67.55** $\pm$0.4642	**67.1** $\pm$0.4032	56.22 $\pm$0.2328	**67.1** $\pm$0.2217	63.55 $\pm$0.5635	64.33 $\pm$0.5844	27.43 $\pm$0.7179	22.04 $\pm$0.128
housing	100	**72** $\pm$0.6521	**71.61** $\pm$0.09316	59.03 $\pm$0.3727	69.24 $\pm$0.4061	67.59 $\pm$0.9815	68.12 $\pm$1.037	67.65 $\pm$0.796	27.14 $\pm$0.3359
LEV	41.20	55.63 $\pm$0.3771	52.73 $\pm$0.1700	56.17 $\pm$0.3399	58.87 $\pm$0.3091	60.83 $\pm$0.6128	60.73 $\pm$1.271	45.43 $\pm$0.8179	**63.03** $\pm$0.2625
SWD	48.70	56.43 $\pm$0.4643	52.8 $\pm$0.4320	56.57 $\pm$0.4784	**58.23** $\pm$0.2055	57.63 $\pm$0.66	57.1 $\pm$0.4320	47.83 $\pm$0.411	**58.6** $\pm$0.4243
windsor	69.60	54.58 $\pm$0.7913	53.05 $\pm$1.349	53.6 $\pm$0.2284	51.83 $\pm$1.813	44.08 $\pm$0.8236	47.99 $\pm$2.888	49.15 $\pm$0.7527	**55.37** $\pm$0.3763

We compared the values from Tables 3 and 4 to the values of λ and δ presented in Table 2. We included the most restrictive values of the respective measures from Table 2 in Tables 3 and 4 to facilitate the comparison. Remember that PCC and MAE were calculated by averaged 10-fold cross validation, while λ and δ were calculated on the whole data sets. Nevertheless, we can observe that thresholds $\lambda_C^{\epsilon_X}(\mathcal{X})$, $\lambda_C^{\mu'_X}(\mathcal{X})$, $\delta_C^{\epsilon_X}(\mathcal{X})$, and $\delta_C^{\mu'_X}(\mathcal{X})$ were never reached during learning. This is not surprising since they are defined as limit values of what can be achieved in learning. On the other hand, the values of $\gamma_C^{\epsilon_X}(\mathcal{X})$ and/or $\gamma_C^{\mu'_X}(\mathcal{X})$ were exceeded for data sets: ERA, ESL, and LEV.

Table 4. Mean absolute error (MAE)

Data set	$\delta_C^{\boldsymbol{\theta}_X^*}(\mathcal{X})$	monotonic VC-DomLEM	non-monotonic VC-DomLEM	Naive Bayes	SVM	RIPPER	C4.5	OLM	OSDL
balance	0	0.1621 ±0.001996	0.1659 ±0.002719	**0.1104** ±0.002613	0.1723 ±0.003017	0.2917 ±0.01088	0.3088 ±0.02174	0.6384 ±0.01713	0.7003 ±0.004588
bank-g	0.0198	**0.04536** ±0.001531	0.04867 ±0.000884	0.1146 ±0.01371	0.1280 ±0.001205	0.0489 ±0.00352	0.0515 ±0.005251	0.05528 ±0.001736	0.1545 ±0
breast-c	0.7622	**0.2331** ±0.003297	0.2436 ±0.007185	0.2564 ±0.005943	0.3217 ±0.01244	0.2960 ±0.01154	0.2424 ±0.003297	0.324 ±0.01835	0.3065 ±0.001648
breast-w	0.0243	0.03720 ±0.002023	0.04578 ±0.003504	0.03958 ±0.0006744	**0.03243** ±0.0006744	0.04483 ±0.004721	0.05532 ±0.00751	0.1764 ±0.00552	0.04149 ±0.001168
car	0.0162	**0.03421** ±0.0007275	0.03524 ±0.0009624	0.1757 ±0.002025	0.08668 ±0.002025	0.2029 ±0.01302	0.1168 ±0.003108	0.09156 ±0.005358	0.04141 ±0.0009624
cpu	0	**0.08293** ±0.01479	**0.0925** ±0.01579	0.1707 ±0.009832	0.4386 ±0.01579	0.1611 ±0.01372	0.1196 ±0.01790	0.3461 ±0.02744	0.3158 ±0.01034
denbosch	0.1008	**0.1232** ±0.01048	**0.1289** ±0.01428	**0.1289** ±0.01428	0.2129 ±0.003961	0.1737 ±0.02598	0.1653 ±0.01048	0.2633 ±0.02206	0.1541 ±0.003961
ERA	2.826	1.307 ±0.002055	1.364 ±0.006018	1.325 ±0.003771	1.318 ±0.007257	1.681 ±0.01558	1.326 ±0.006018	1.321 ±0.01027	**1.280** ±0.00704
ESL	0.1578	0.3702 ±0.01352	0.4146 ±0.005112	**0.3456** ±0.003864	0.4262 ±0.01004	0.4296 ±0.01608	0.3736 ±0.01089	0.474 ±0.01114	**0.3422** ±0.005019
fame	0.0211	**0.3406** ±0.001878	0.3469 ±0.004	0.4829 ±0.002906	**0.3406** ±0.001775	0.3991 ±0.003195	0.3863 ±0.005253	1.577 ±0.03791	1.592 ±0.007555
housing	0	0.3235 ±0.01133	**0.3083** ±0.00559	0.5033 ±0.006521	0.3551 ±0.005187	0.3676 ±0.007395	0.3676 ±0.01556	0.3867 ±0.01050	1.078 ±0.00796
LEV	0.801	0.4813 ±0.004028	0.5187 ±0.002867	0.475 ±0.004320	0.4457 ±0.003399	0.4277 ±0.00838	0.426 ±0.01476	0.615 ±0.0099	**0.4033** ±0.003091
SWD	0.68	0.454 ±0.004320	0.4857 ±0.005249	0.475 ±0.004320	0.4503 ±0.002867	0.452 ±0.006481	0.4603 ±0.004497	0.5707 ±0.007717	**0.433** ±0.002160
windsor	0.4066	**0.5024** ±0.006226	0.5201 ±0.003956	0.5488 ±0.005662	0.5891 ±0.02101	0.6825 ±0.03332	0.652 ±0.03721	0.5757 ±0.006044	0.5153 ±0.006044

The nine data sets that were distinguished by $\lambda_C^{\boldsymbol{\theta}_X^*}(\mathcal{X})$ and $\delta_C^{\boldsymbol{\theta}_X^*}(\mathcal{X})$ as at least not bad in terms of consistency, and thus, easier to learn, are also those on which classifiers showed good actual predictive accuracy. Exception to this rule are data sets: ESL, fame, and housing. This may be caused by the fact that these data sets are described by many attributes and/or classes. It is thus visible that measures λ and δ allowed to distinguish the data sets which are just hard to learn (ESL, fame, and housing) from those which are inconsistent and hard to learn (breast-c, ERA, LEV, SWD, windsor). It can be also seen that for the highly inconsistent data sets: breast-c, ERA, LEV and SWD, all classifiers performed better than the values of $\lambda_C^{\boldsymbol{\theta}_X^*}(\mathcal{X})$ and $\delta_C^{\boldsymbol{\theta}_X^*}(\mathcal{X})$. The only exception is PCC of OLM for data set SWD. This indicates that the classifiers were able to overcome the inconsistencies present in the highly inconsistent data sets.

5 Conclusions

We have introduced two measures, λ and δ, that estimate attainable predictive accuracy of classifiers learned in indiscernibility-based and dominance-based rough set approaches. We have shown that λ is a better estimate than the well known quality of classification γ. Values of λ and δ were compared to actual predictive accuracies calculated in a computational experiment on fourteen data

sets. The results show that both introduced measures are useful for an a priori identification of data sets that are hard to learn by all classifiers. Moreover, they can be used to identify the data sets that are consistent and just hard to learn.

Acknowledgment

The authors wish to acknowledge financial support from the Ministry of Science and Higher Education, grant N N519 314435.

References

1. Ben-David, A.: Monotonicity maintenance in information-theoretic machine learning algorithms. Machine Learning 19(1), 29–43 (1995)
2. Ben-David, A., Sterling, L., Pao, Y.-H.: Learning and classification of monotonic ordinal concepts. Computational Intelligence 5(1), 45–49 (1989)
3. Błaszczyński, J., Greco, S., Słowiński, R., Szeląg, M.: On variable consistency dominance-based rough set approaches. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 191–202. Springer, Heidelberg (2006)
4. Błaszczyński, J., Greco, S., Słowiński, R., Szeląg, M.: Monotonic variable consistency rough set approaches. Int. Journ. of Approx. Reason. 50(7), 979–999 (2009)
5. Błaszczyński, J., Słowiński, R., Szeląg, M.: Sequential covering rule induction algorithm for variable consistency rough set approaches. Information Sciences (submitted in 2009)
6. Cao-Van, K.: Supervised ranking – from semantics to algorithms. PhD thesis, Ghent University, CS Department (2003)
7. Gediga, G., Düntsch, I.: Approximation quality for sorting rules. Computational Statistics & Data Analysis 40, 499–526 (2002)
8. Greco, S., Matarazzo, B., Słowiński, R.: Rough sets theory for multicriteria decision analysis. European Journal of Operational Research 129(1), 1–47 (2001)
9. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
10. Słowiński, R., Greco, S., Matarazzo, B.: Rough sets in decision making. In: Meyers, R.A. (ed.) Encyclopedia of Complexity and Systems Science, pp. 7753–7786. Springer, New York (2009)
11. Ziarko, W.: On learnability of decision tables. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 394–401. Springer, Heidelberg (2004)

Upper Bounds on Minimum Cardinality of Exact and Approximate Reducts

Igor Chikalov[1], Mikhail Moshkov[1], and Beata Zielosko[2]

[1] Mathematical and Computer Sciences & Engineering Division
King Abdullah University of Science and Technology
Thuwal 23955-6900, Saudi Arabia
{igor.chikalov,mikhail.moshkov}@kaust.edu.sa
[2] Institute of Computer Science, University of Silesia
39, Będzińska St., Sosnowiec, 41-200, Poland
beata.zielosko@us.edu.pl

Abstract. In the paper, we consider the notions of exact and approximate decision reducts for binary decision tables. We present upper bounds on minimum cardinality of exact and approximate reducts depending on the number of rows (objects) in the decision table. We show that the bound for exact reducts is unimprovable in the general case, and the bound for approximate reducts is almost unimprovable in the general case.

Keywords: Exact reduct, approximate reduct, upper bound on minimum cardinality.

1 Introduction

The paper is devoted to the consideration of upper bounds on minimum cardinality of exact and approximate reducts for binary decision tables. There are different variants of the notion of reduct [6]. We study here decision reducts. One of the main problems of rough set theory is to find reduct with minimum cardinality [4,5,8]. Simple upper bounds on minimum cardinality of reducts can help us to decrease the search range.

For exact reducts we have the following upper bound on minimum cardinality:

$$R(T) \leq N(T) - 1,$$

where $R(T)$ is the minimum cardinality of a reduct for the decision table T and $N(T)$ is the number of rows (objects) in the table T. In the paper we show that this well known bound (we don't know who is the author of this bound) is unimprovable in the general case.

Exact reducts can be overfitted, i.e., depend essentially on the noise or adjusted too much to the existing examples. So last years various kinds of approximate reducts were studied intensively in rough set theory [1,2,3,7,9,10,11]. In this paper, we consider one more definition of an approximate reduct based on an uncertainty measure $P(T)$ of decision table T, which is equal to the number of unordered pairs of rows with different decisions in T.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 412–417, 2010.

Let α be a real number such that $0 \leq \alpha < 1$. A subset of the set of attributes divides the table T into subtables T' in each of which the considered attributes have constant values. This subset is called an α-reduct for the table T if $P(T') \leq \alpha P(T)$ for each subtable T', and we can't remove any attribute from the subset without loss of this property. In the paper we prove that

$$R_\alpha(T) \leq (1-\alpha)N(T) + 1,$$

where $R_\alpha(T)$ is the minimum cardinality of α-reduct for the table T. We show that this bound is almost unimprovable in the general case.

The paper consists of five sections. In Sect. 2, notions of binary decision table, exact test and reduct are considered, and an upper bound on the minimum cardinality of exact reduct is presented. In Sect. 3, notions of approximate test and reduct are discussed, and an upper bound on the minimum cardinality of approximate reduct is proved. In Sect. 4, the quality of upper bounds on minimum cardinality of exact and approximate reducts is studied. Section 5 contains short conclusions.

2 Exact Tests and Reducts

In this section, the notions of binary decision table, exact test (superreduct) and exact reduct are described, and an upper bound on minimum cardinality of exact reduct is presented.

A binary decision table T is a rectangular table which elements belong to the set $\{0, 1\}$. Columns of this table are labeled with names of attributes $f_1, ..., f_n$. Rows of the table are pairwise different, and each row is labeled with a natural number (a decision). A test for T is a subset of columns (attributes) such that at the intersection with these columns any two rows with different decisions are different. A reduct for T is a test for T for which each proper subset is not a test. It is clear that each test has a reduct as a subset. We denote by $R(T)$ the minimum cardinality of a reduct for T, and by $N(T)$ we denote the number of rows in the table T.

Let $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$ and $\delta_1, \ldots, \delta_m \in \{0, 1\}$. We denote by

$$T(f_{i_1}, \delta_1) \ldots (f_{i_m}, \delta_m)$$

a subtable of T that contains only rows of T, which at the intersection with columns $f_{i_1}, \ldots, f_{i_m}$ have numbers $\delta_1, \ldots, \delta_m$ respectively.

Theorem 1. *Let T be a binary decision table. Then*

$$R(T) \leq N(T) - 1.$$

Proof. We prove this inequality by induction on $N(T)$. If $N(T) = 1$ then, evidently, $R(T) = 0$, since there are no pairs of rows with different decisions.

Let $m \geq 1$ and for any decision table T with $N(T) \leq m$ the inequality $R(T) \leq N(T) - 1$ holds. Let T be a decision table with $N(T) = m + 1$. We now

prove that $R(T) \leq m$. Since T has at least two rows and rows of T are pairwise different, there exists a column f_i of T, which has both 0 and 1.

Let us consider subtables $T(f_i, 0)$ and $T(f_i, 1)$. It is clear that each of these subtables has at most m rows. Using inductive hypothesis we obtain that for $\delta = 0, 1$ there exists a test B_δ for the table $T(f_i, \delta)$ such that $|B_\delta| \leq N(T(f_i, \delta)) - 1$. We denote $B = \{f_i\} \cup B_0 \cup B_1$. It is clear that B is a test for the table T and $|B| \leq 1 + N(T(f_i, 0)) - 1 + N(T(f_i, 1)) - 1$.

Since $N(T) = N(T(f_i, 0)) + N(T(f_i, 1))$, we have $|B| \leq N(T) - 1$. Therefore $R(T) \leq N(T) - 1$. □

3 Approximate Tests and Reducts

In this section, notions of approximate test and approximate reduct are described, and an upper bound on minimum cardinality of approximate reduct is presented.

Let T be a binary decision table. We denote by $P(T)$ the number of unordered pairs of rows of T with different decisions. We will say that T is a degenerate table if T doesn't have rows or all rows of T are labeled with the same decision. It is clear that T is degenerate if and only if $P(T) = 0$.

Let α be a real number such that $0 \leq \alpha < 1$. An α-test for the table T is a subset of columns $\{f_{i_1}, ..., f_{i_m}\}$ of T such that for any numbers $\delta_1, ..., \delta_m \in \{0, 1\}$ the inequality $P(T(f_{i_1}, \delta_1)...(f_{i_m}, \delta_m)) \leq \alpha P(T)$ holds. Empty set is an α-test for T if and only if T is a degenerate table. An α-reduct for the table T is an α-test T for which each proper subset is not an α-test.

We denote by $R_\alpha(T)$ the minimum cardinality of an α-test for the table T. It is clear that each α-test has an α-reduct as a subset. Therefore $R_\alpha(T)$ is the minimum cardinality of an α-reduct. It is clear also that the set of tests for the table T coincides with the set of 0-tests for T. Therefore $R_0(T) = R(T)$. Let α, β be real numbers such that $0 \leq \alpha \leq \beta < 1$. One can show that each α-test for T is also a β-test for T. Thus, $R_\alpha(T) \geq R_\beta(T)$.

Theorem 2. *Let T be a binary decision table and α be a real number such that $0 \leq \alpha < 1$. Then*

$$R_\alpha(T) \leq (1 - \alpha)N(T) + 1.$$

Proof. We will prove the considered inequality by induction on $N(T)$. If $N(T) = 1$ then $R_\alpha(T) = 0$ and the considered inequality holds. Let for a natural $m \geq 1$ for any decision table T with $N(T) \leq m$ and for any real β, $0 \leq \beta < 1$, the inequality $R_\beta(T) \leq (1 - \beta)N(T) + 1$ holds.

Let T be a decision table with $N(T) = m + 1$ and α be a real number, $0 \leq \alpha < 1$. If T is a degenerate table then $R_\alpha(T) = 0$, and the considered inequality holds. Let us assume now that there exist two rows in T, which are labeled with different decisions. Let these rows be different in a column f_i of the table T. We denote $T_0 = T(f_i, 0)$, $T_1 = T(f_i, 1)$, $N = N(T)$, $N_0 = N(T_0)$ and $N_1 = N(T_1)$. It is clear that $1 \leq N_0 \leq m$ and $1 \leq N_1 \leq m$. We consider three cases.

1. Let $P(T_0) \leq \alpha P(T)$ and $P(T_1) \leq \alpha P(T)$. In this case $\{f_i\}$ is an α-test for the table T, and
$$R_\alpha(T) \leq 1 \leq (1-\alpha)N(T) + 1.$$
2. Let $P(T_0) \leq \alpha P(T)$ and $P(T_1) > \alpha P(T)$ (the case $P(T_1) \leq \alpha P(T)$ and $P(T_0) > \alpha P(T)$ can be considered in the same way). We denote $\beta_1 = \frac{\alpha P(T)}{P(T_1)}$. It is clear that $0 \leq \beta_1 < 1$. Using inductive hypothesis we conclude that there exists a β_1-test B_1 for the table T_1 such that $|B_1| \leq (1-\beta_1)N(T_1)+1$. It is not difficult to show that $B_1 \cup \{f_i\}$ is an α-test for the table T.
Let us prove that $\beta_1 \geq \alpha\frac{N}{N_1}$. To this end, we will show that $\frac{N}{N_1} \leq \frac{P(T)}{P(T_1)}$. It is clear that $P(T) = P(T_0) + P(T_1) + P(T_0, T_1)$ where $P(T_0, T_1)$ is the number of pairs of rows (r', r'') with different decisions such that r' is from T_0 and r'' is from T_1. Thus, $\frac{N}{N_1} = \frac{N_1}{N_1} + \frac{N_0}{N_1} = 1 + \frac{N_0}{N_1}$ and $\frac{P(T)}{P(T_1)} = 1 + \frac{P(T_0)}{P(T_1)} + \frac{P(T_0,T_1)}{P(T_1)}$. We will show that $\frac{N_0}{N_1} \leq \frac{P(T_0,T_1)}{P(T_1)}$. Let $r_1, ...r_{N_0}$ be all rows from T_0. For $i = 1, ..., N_0$ we denote by P_i the number of pairs of rows (r_i, r'') with different decisions, such that r'' is from T_1. Then $\frac{P(T_0,T_1)}{P(T_1)} = \frac{\sum_{i=1}^{N_0} P_i}{P(T_1)}$.
Let us show that $\frac{P_i}{P(T_1)} \geq \frac{1}{N_1}$ for any $i \in \{1, ..., N_0\}$. We consider rows of the table T_1. Let b be the number of rows which have the same decision as r_i. Let a be the number of rows which have other decisions. Then $P_i = a$, $P(T_1) \leq ab + \frac{a(a-1)}{2}$ and $N_1 = a + b$. Since $P(T_1) > \alpha P(T)$, we have T_1 is a non-degenerate table. Therefore, $N_1 \geq 2$ and $a \geq 1$. So, $\frac{P_i}{P(T_1)} \geq \frac{a}{ab+\frac{a(a-1)}{2}} = \frac{1}{b+\frac{a-1}{2}} \geq \frac{1}{b+a}$. Thus, $\frac{P(T_0,T_1)}{P(T_1)} \geq \frac{N_0}{N_1}$, $\frac{P(T)}{P(T_1)} \geq \frac{N}{N_1}$, and $\beta_1 = \frac{\alpha P(T)}{P(T_1)} \geq \frac{\alpha N}{N_1}$.
Therefore,
$$\begin{aligned}|B_1 \cup \{f_1\}| &\leq (1-\beta_1)N_1 + 2 \leq \left(1 - \frac{\alpha N}{N_1}\right) N_1 + 2 \\ &= N_1 - \alpha N + 2 \leq N - \alpha N + 1 = N(1-\alpha) + 1.\end{aligned}$$
We used here evident inequality $N_1 + 1 \leq N$.
3. Let $P(T_0) > \alpha P(T)$ and $P(T_1) > \alpha P(T)$. We denote $\beta_0 = \frac{\alpha P(T)}{P(T_0)}$ and $\beta_1 = \frac{\alpha P(T)}{P(T_1)}$. It is clear that $0 < \beta_0 < 1$ and $0 < \beta_1 < 1$. Using inductive hypothesis we obtain that there exists a β_0-test B_0 for the table T_0 such that $|B_0| \leq (1-\beta_0)N_0 + 1$. Also, there exists a β_1-test B_1 for the table T_1 such that $|B_1| \leq (1-\beta_1)N_1 + 1$. It is not difficult to show that $B_0 \cup B_1 \cup \{f_i\}$ is an α-test for the table T. As for the case 2, one can prove that $\beta_0 \geq \frac{\alpha N}{N_0}$ and $\beta_1 \geq \frac{\alpha N}{N_1}$. Therefore,
$$\begin{aligned}|B_0 \cup B_1 \cup \{f_i\}| &\leq \left(1 - \frac{\alpha N}{N_0}\right) N_0 + 1 + \left(1 - \frac{\alpha N}{N_1}\right) N_1 + 1 + 1 \\ &= N_0 - \alpha N + N_1 - \alpha N + 3 = N - \alpha N + 1 + 2 - \alpha N \\ &= (1-\alpha)N + 1 + 2 - \alpha N.\end{aligned}$$
Let $\alpha N \geq 2$. Then we have $R_\alpha(T) \leq (1-\alpha)N + 1$.
Let now $\alpha N < 2$. Using Theorem 1 we have $R_\alpha(T) \leq R_0(T) \leq N - 1 \leq N - 1 + 2 - \alpha N = (1-\alpha)N + 1$. □

4 Quality of Bounds

In this section, we show that the bound from Theorem 1 is unimprovable in the general case, and the bound from Theorem 2 is almost unimprovable in the general case.

Let n be a natural number. We consider a decision table T_n which contains n columns labeled with conditional attributes $f_1, \ldots, f_n$ and $n+1$ rows. For $i = 1, \ldots, n$, the i-th row has 1 at the intersection with the column f_i. All other positions in the row are filled by 0. This row is labeled with the decision 1. The last $(n+1)$-th row is filled by 0 only and is labeled with the decision 2. One can show that $P(T_n) = n = N(T_n) - 1$.

Let α be a real number such that $0 \leq \alpha < 1$, and $\{f_{i_1}, \ldots, f_{i_m}\}$ be a subset of the set of attributes. It is clear that $P(T_n(f_{i_1}, 0) \ldots (f_{i_m}, 0)) = n - m = P(T_n) - m$. If $\{f_{i_1}, \ldots, f_{i_m}\}$ is an α-test for T_n then

$$P(T_n) - m \leq \alpha P(T_n) = P(T_n) - (1-\alpha)P(T_n)$$

and

$$m \geq (1-\alpha)P(T_n) = (1-\alpha)(N(T_n) - 1) = (1-\alpha)N(T_n) + \alpha - 1.$$

Therefore

$$R_\alpha(T_n) \geq (1-\alpha)N(T_n) + \alpha - 1. \tag{1}$$

Let $\alpha = 0$. By (1),

$$R(T_n) = R_0(T_n) \geq N(T_n) - 1.$$

From Theorem 1 it follows that

$$R(T_n) \leq N(T_n) - 1.$$

Thus the bound from Theorem 1 is unimprovable in the general case.

From Theorem 2 it follows that

$$R_\alpha(T_n) \leq (1-\alpha)N(T_n) + 1.$$

The difference between lower (1) and upper (from Theorem 2) bounds is at most 2. Hence the bound from Theorem 2 is almost unimprovable in the general case.

5 Conclusions

In the paper, upper bounds on minimum cardinality of exact and approximate reducts are considered. We showed that the bound for exact reducts is unimprovable, and the bound for approximate reducts is almost unimprovable in the general case.

References

1. Moshkov, M., Piliszczuk, M., Zielosko, B.: Universal attribute reduction problem. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 417–426. Springer, Heidelberg (2007)
2. Moshkov, M., Piliszczuk, M., Zielosko, B.: Partial Covers, Reducts and Decision Rules in Rough Sets: Theory and Applications. Springer book series Studies in Computational Intelligence, vol. 145. Springer, Heidelberg (2008)
3. Nguyen, H.S., Ślęzak, D.: Approximate reducts and association rules – correspondence and complexity results. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) RSFDGrC 1999. LNCS (LNAI), vol. 1711, pp. 137–145. Springer, Heidelberg (1999)
4. Pawlak, Z.: Rough sets. International J. Comp. Inform. Science 11, 341–356 (1982)
5. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
6. 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)
7. Skowron, A.: Rough sets in KDD. In: Proceedings of the 16th World Computer Congress (IFIP 2000), Beijing, China, pp. 1–14 (2000)
8. 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)
9. Ślęzak, D.: Approximate entropy reducts. Fundamenta Informaticae 53, 365–390 (2002)
10. Ślęzak, D., Wróblewski, J.: Order-based genetic algorithms for the search of approximate entropy reducts. In: Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.) RSFDGrC 2003. LNCS (LNAI), vol. 2639, pp. 308–311. Springer, Heidelberg (2003)
11. Wróblewski, J.: Ensembles of classifiers based on approximate reducts. Fundamenta Informaticae 47, 351–360 (2001)

An Extension of Rough Set Approximation to Flow Graph Based Data Analysis

Doungrat Chitcharoen and Puntip Pattaraintakorn

School of Mathematics, Faculty of Science,
King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand 10520
s0067111@kmitl.ac.th, kppuntip@kmitl.ac.th

Abstract. This paper concerns some aspects of mathematical flow graph based data analysis. In particular, taking a flow graph view on rough sets' categories and measures leads to a new methodology of inductively reasoning form data. This perspective shows interesting relationships and properties among rough set, flow graphs and inverse flow graphs. A possible car dealer application is outlined and discussed. Evidently, our new categories and measures assist and alleviate some limitations in flow graphs to discover new patterns and explanations.

Keywords: Flow graphs, rough sets and decision rules.

1 Introduction

A mathematical flow graph, invented by Pawlak in 2002, is an extension of *rough set theory* [9]. A flow graph represents the information flow from the given data set [10,11,12,13,14]. The branches of a flow graph can be constructed as decision rules, with every decision rule, there are three associated coefficients: *strength*, *certainty* and *coverage* [14]. These coefficients satisfy Bayes' theorem. Inference in flow graphs has polynomial time and flow conservation comes with probabilistic conditional independencies in the problem domain [1]. Flow graphs have led to many interesting applications and extensions such as preference analysis [11], decision tree [13], survival analysis [7], association rule [3], data mining [14], search engines [2], fuzzy set [4,6], entropy measures [8] and granular computing [5]. More studies involving rough sets are discussed and provided in [15].

Flow distribution in flow graphs can be exploited for approximation and reasoning. Based on flow graph contexts, we define fundamental definitions for rough sets: four categories of vagueness, accuracy of approximation, roughness of approximation and dependency degree. In addition, we state formulas to conveniently compute these measures for inverse flow graphs. To illustrate, a possible car dealer preference analysis is provided to support our propositions. New categories and measures assist and alleviate some limitations in flow graphs to discover new patterns and explanations.

This paper is organized as follows. In Section 2, we present the basic concepts of rough sets. In Section 3, we recall preliminary definitions of flow graphs. In Section 4, we present a new bridge between rough sets and flow graphs with an example throughout, followed by a conclusion in the last section.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 418–427, 2010.

2 Rough Set Theory

The following rough sets preliminary is taken from [9]. Rough sets are based on an *information system*. Formally, it is a pair $S = (U, A)$, where U is a nonempty finite set of objects called the *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 called the *domain* of a.

If we partition an information system into two disjoint classes of attributes, called *condition* and *decision attributes*, then the information system will be called a *decision system*, denoted by $S = (U, C, D)$, where $C \cap D = \emptyset$. Any subset B of A determines a binary relation $I(B)$ on U called an *indiscernibility relation*. It is defined as $(x, y) \in I(B)$ if and only if $a(x) = a(y)$ for every $a \in A$, where $a(x)$ denotes the attribute value of element x. Equivalence classes of the relation $I(B)$ are referred to as *B-elementary sets* or *B-elementary granules* denote by $B(X)$, i.e., $B(X)$ describes X in the terms of attribute values from B [11]. Below, we recall key feature definitions of approximations in rough sets.

Definition 1. *[15] Let $S = (U, A)$ be an information system. For $X \subseteq U$, $B \subseteq A$. The B-lower approximations, B-upper approximations and B-boundary region of X are defined as $\underline{B}(X) = \bigcup_{x \in U} \{B(X) | B(X) \subseteq X\}$, $\overline{B}(X) = \bigcup_{x \in U} \{B(X) | B(X) \cap X \neq \emptyset\}$ and $BN_B(X) = \overline{B}(X) - \underline{B}(X)$, respectively.*

If the boundary region of X is the empty set (i.e., $BN_B(x) = \emptyset$), then X is *crisp*. On the contrary, if $BN_B(X) \neq \emptyset$, then X is *rough*. In what follows we recall four basic classes of rough sets, i.e., four categories of vagueness.

Definition 2. *[15] Let $S = (U, A)$ be an information system. For $X \subseteq U$, $B \subseteq A$, the four categories of vagueness are defined as*

- *$\underline{B}(X) \neq \emptyset$ and $\overline{B}(X) \neq U$ iff X is roughly B-definable,*
- *$\underline{B}(X) = \emptyset$ and $\overline{B}(X) \neq U$ iff X is internally B-indefinable,*
- *$\underline{B}(X) \neq \emptyset$ and $\overline{B}(X) = U$ iff X is externally B-definable,*
- *$\underline{B}(X) = \emptyset$ and $\overline{B}(X) = U$ iff X is totally B-indefinable.*

Approximation of a rough set can be characterized numerically by some measurements as follows.

Definition 3. *[15] Let $S = (U, A)$ be an information system. For $X \subseteq U$, $B \subseteq A$, the accuracy of approximation, $\alpha_B(X)$, and roughness of approximation, $\gamma_B(X)$, are defined respectively as $\alpha_B(X) = \frac{card(\underline{B}(X))}{card(\overline{B}(X))}$ and $\gamma_B(X) = 1 - \alpha_B(X) = 1 - \frac{card(\underline{B}(X))}{card(\overline{B}(X))}$, where $card(X)$ denotes the cardinality of X.*

Let us observe that, $0 \leq \alpha_B(X) \leq 1$. If $\alpha_B(X) = 1$, then X is *crisp* with respect to B and otherwise, if $\alpha_B(X) < 1$, then X is *rough* with respect to B.

Definition 4. *Let $S = (U, A)$ be an information system and $F = \{X_1, X_2, \ldots, X_n\}$ be a partition of the universe U. For $B \subseteq A$, F depends on B to a degree $k_B(F) = \frac{\sum_{i=1}^{n} card(\underline{B}(X_i))}{card(U)}$.*

Definitions $2 - 4$ will be stated in the context of flow graphs in Section 4.

3 Flow Graphs

In this section, we recall some concepts of flow graphs which were introduced by Pawlak in [10,11,12,13,14].

A *flow graph* is a *directed, acyclic, finite graph* $G = (\mathcal{N}, \mathcal{B}, \varphi)$, where $\mathcal{N}$ is a set of nodes, $\mathcal{B} \subseteq \mathcal{N} \times \mathcal{N}$ is a set of *directed branches*, φ: $\mathcal{B} \to R^+$ is a *flow function* and R^+ is the set of non-negative real numbers. If $(x, y) \in \mathcal{B}$ then x is an *input* of node y denoted by $I(y)$ and y is an *output* of node x denoted by $O(x)$. The *input* and *output* of a flow graph G are defined by $I(G) = \{x \in \mathcal{N} \mid I(x) = \emptyset\}$ and $O(G) = \{x \in \mathcal{N} \mid O(x) = \emptyset\}$. These inputs and outputs of G are called *external nodes* of G whereas other nodes are called *internal nodes* of G. If $(x, y) \in \mathcal{B}$ then we call (x, y) a *throughflow* from x to y. We will assume in what follows that $\varphi(x, y) \neq 0$ for every $(x, y) \in \mathcal{B}$. With every node x of a flow graph G, we have its associated *inflow* and *outflow* respectively as: $\varphi_+(x) = \sum_{y \in I(x)} \varphi(y, x)$ and $\varphi_-(x) = \sum_{y \in O(x)} \varphi(x, y)$. Similarly, an *inflow* and an *outflow* for the flow graph G are defined as: $\varphi_+(G) = \sum_{x \in I(G)} \varphi_-(x)$ and $\varphi_-(G) = \sum_{x \in O(G)} \varphi_+(x)$. We assume that for any internal node x, $\varphi_-(x) = \varphi_+(x) = \varphi(x)$, where $\varphi(x)$ is a *throughflow* of node x. Similarly then, $\varphi_-(G) = \varphi_+(G) = \varphi(G)$ is a *throughflow* of graph G. As discussed by Pawlak [11], the above equations can be considered as *flow conservation equations* (or *pairwise consistent* [1]).

Normalized Flow Graphs, Paths and Connections

In order to demonstrate interesting relationships between flow graphs and other disciplines (e.g., statistics), we come to the normalized version of flow graphs.

A *normalized flow graph* is a *directed, acyclic, finite graph* $G = (\mathcal{N}, \mathcal{B}, \sigma)$, where $\mathcal{N}$ is a set of *nodes*, $\mathcal{B} \subseteq \mathcal{N} \times \mathcal{N}$ is a set of *directed branches* and σ: $\mathcal{B} \to [0, 1]$ is a *normalized flow function* of (x, y). The *strength* of (x, y) is $\sigma(x, y) = \frac{\varphi(x,y)}{\varphi(G)}$. With every node x of a flow graph G, the associated *normalized inflow* and *outflow* are defined as: $\sigma_+(x) = \frac{\varphi_+(x)}{\varphi(G)} = \sum_{y \in I(x)} \sigma(y, x)$, $\sigma_-(x) = \frac{\varphi_-(x)}{\varphi(G)} = \sum_{y \in O(x)} \sigma(x, y)$. For any internal node x, it holds that $\sigma_+(x) = \sigma_-(x) = \sigma(x)$, where $\sigma(x)$ is a *normalized throughflow* of x. Similarly, *normalized inflow* and *outflow* for the flow graph G are defined as: $\sigma_+(G) = \frac{\varphi_+(G)}{\varphi(G)} = \sum_{x \in I(G)} \sigma_-(x)$, $\sigma_-(G) = \frac{\varphi_-(G)}{\varphi(G)} = \sum_{x \in O(x)} \sigma_+(x)$. It also holds that $\sigma_+(G) = \sigma_-(G) = \sigma(G) = 1$. With every branch (x, y) of a flow graph G, the *certainty* and the *coverage* of (x, y) are defined respectively as: $cer(x, y) = \frac{\sigma(x,y)}{\sigma(x)}$, $cov(x, y) = \frac{\sigma(x,y)}{\sigma(y)}$, where $\sigma(x), \sigma(y) \neq 0$. Properties of these coefficients were studied by Pawlak in [10,11,12,13,14].

Next, if we focus on sequence of nodes in a flow graph, we can find them by using the concept of a directed simple path. A (directed) *path* from x to y $(x \neq y)$ in G, denoted by $[x \ldots y]$, is a sequence of nodes $x_1, \ldots, x_n$ such that $x_1 = x$ and $x_n = y$ and $(x_i, x_{i+1}) \in B$ for every i, $1 \leq i \leq n - 1$. The *certainty, coverage* and *strength of the path* $[x_1 \ldots x_n]$ are defined respectively as:

$cer[x_1 \ldots x_n] = \prod_{i=1}^{n-1} cer(x_i, x_{i+1})$, $cov[x_1 \ldots x_n] = \prod_{i=1}^{n-1} cov(x_i, x_{i+1})$, $\sigma[x \ldots y] = \sigma(x)cer[x...y] = \sigma(y)cov[x \ldots y]$.

The set of all paths from x to y $(x \neq y)$ in G, denoted by $\langle x, y\rangle$, is a *connection* of G determined by nodes x and y. For every connection $\langle x, y\rangle$, the associated *certainty, coverage* and *strength of the connection* $\langle x, y\rangle$ are defined as: $cer\,\langle x, y\rangle = \sum_{[x\ldots y]\in\langle x,y\rangle} cer[x \ldots y]$, $cov\,\langle x, y\rangle = \sum_{[x\ldots y]\in\langle x,y\rangle} cov[x \ldots y]$, $\sigma\,\langle x, y\rangle = \sum_{[x\ldots y]\in\langle x,y\rangle} \sigma[x \ldots y] = \sigma(x)cer\,\langle x, y\rangle = \sigma(y)cov\,\langle x, y\rangle$. If $[x \ldots y]$ is a path such that x and y are the input and output of G, then $[x \ldots y]$ will be referred to as a *complete path*. The set of complete paths from x to y will be called a *complete connection* from x to y in G.

If we substitute every complete connection $\langle x, y\rangle$ in G, where x and y are an input and an output of a graph G with a single branch (x, y) such that $\sigma(x, y) = \sigma\,\langle x, y\rangle$, $cer(x, y) = cer\,\langle x, y\rangle$ and $cov(x, y) = cov\,\langle x, y\rangle$ then we have a new flow graph G' with the property: $\sigma(G) = \sigma(G')$. G' is called a *combined flow graph*.

Starting from a flow graph, if we invert the direction of all branches in G, then the resulting graph G^{-1} will be called the *inverted graph of* G (or the *inverse flow graph of* G) [14]. Essentially, three coefficients of an inverse flow graph can be computed from its flow graph as follows: $\sigma_{G^{-1}}(y, x) = \sigma_G(x, y)$, $cer_{G^{-1}}(y, x) = cov_G(x, y)$ and $cov_{G^{-1}}(y, x) = cer_G(x, y)$.

4 Rough Set Approximations and Flow Graphs

In this section, we provide a bridge between flow graphs and rough approximation. From standard definitions of approximations made by rough sets, we give these definitions in the context of flow graphs below.

Suppose we are given a normalized flow graph $G = (A, \mathcal{B}, \sigma)$, where $A = \{A_{l_1}, A_{l_2}, \ldots, A_{l_n}\}$ is a set of attributes[1], $\mathcal{B}$ is a set of directed branches and σ is a normalized flow function. A set of nodes in a flow graph G corresponding to A_{l_i} is referred to as a *layer* i. For $A = C \cup D$, we have that every layer corresponding to C will be called a *condition layer* whereas every layer corresponding to D will be called a *decision layer*. If an attribute A_{l_i} contains n_{l_i} values, we say that it contains n_{l_i} nodes.

We now consider how to approximate an attribute value $Y \in A_{l_{i+1}}$ from attribute values of A_{l_i} where $A_{l_i} = \left\{X_1, X_2, \ldots, X_{n_{l_i}}\right\}$, to indicate lower approximation, upper approximation and boundary region of Y. In Definition 5, we recall Pawlak's definitions of lower approximation, upper approximation and boundary region for flow graphs.

Definition 5. *[11] Let $G = (A, \mathcal{B}, \sigma)$ be a normalized flow graph, $A_{l_i} = \{X_1, X_2, \ldots, X_{n_{l_i}}\}$, $1 \leq i \leq k-1$, be an attribute in layer i and Y be a node in $A_{l_{i+1}}$. For any branch (X_j, Y), $j \in \{1, \ldots, n_{l_i}\}$, of the flow graph G, the union of all inputs X_j of Y is the upper approximation of Y (denoted $\overline{A_{l_i}}(Y)$), the union of all inputs X_j of Y, such that $cer(X_j, Y) = 1$, is the lower approximation*

[1] In what follows, we regard $\mathcal{N}$ as A for simplicity.

of Y (denoted $\underline{A_{l_i}}(Y)$). Moreover, the union of all inputs X_j of Y, such that $cer(X_j, Y) < 1$, is the boundary region of Y (denoted $A_{l_i}N_{A_{l_i}}(Y)$).

In Definition 6, we state four categories of rough sets mentioned in Definition 2 in terms of flow graph.

Definition 6. *Let $G = (A, \mathcal{B}, \sigma)$ be a flow graph, $A_{l_i} = \left\{X_1, X_2, \ldots, X_{n_{l_i}}\right\}$, $1 \leq i \leq k-1$, be an attribute in layer i and Y be a node in $A_{l_{i+1}}$. For any branch (X_j, Y), $j \in \{1, \ldots, n_{l_i}\}$, of G, we define four categories of vagueness as*

- *$\exists X_j\,[cer(X_j, Y) = 1]$ and $\exists X_j\,[X_j \notin I(Y)]$ iff Y is roughly A_{l_i}-definable,*
- *$\forall X_j\,[cer(X_j, Y) \neq 1]$ and $\exists X_j\,[X_j \notin I(Y)]$ iff Y is internally A_{l_i}-indefinable,*
- *$\exists X_j\,[cer(X_j, Y) = 1]$ and $\forall X_j\,[X_j \in I(Y)]$ iff Y is externally A_{l_i}-definable,*
- *$\forall X_j\,[cer(X_j, Y) \neq 1]$ and $\forall X_j\,[X_j \in I(Y)]$ iff Y is totally A_{l_i}-indefinable.*

From the definition we obtain the following interpretation:

- if Y is roughly A_{l_i}-*definable*, this means that we are able to decide for some elements of U whether they belong to Y or $-Y$[2], using A_{l_i},
- if Y is internally A_{l_i}-*indefinable*, this means that we are able to decide whether some elements of U belong to $-Y$, but we are unable to decide for any element of U, whether it belongs to Y or not, using A_{l_i},
- if Y is externally A_{l_i}-*indefinable*, this means that we are able to decide for some elements of U whether they belong to Y, but we are unable to decide, for any element of U whether it belongs to $-Y$ or not, using A_{l_i},
- if Y is totally A_{l_i}-*indefinable*, we are unable to decide for any element of U whether it belongs to Y or $-Y$, using A_{l_i}.

Property 1. *Let $G = (A, \mathcal{B}, \sigma)$ be a flow graph, $A_{l_i} = \left\{X_1, X_2, \ldots, X_{n_{l_i}}\right\}$, $2 \leq i \leq k$, be an attribute in layer i and W be a node in $A_{l_{i-1}}$. For any branch (X_j, W), $j \in \{1, \ldots, n_{l_i}\}$ in the inverse flow graph of G, the union of all output X_j of W in flow graph G is the upper approximation of W, the union of all outputs X_j of W in a flow graph G, such that $cov(W, X_j) = 1$, is the lower approximation of W. Moreover, the union of all outputs X_i of W, such that $cov(W, X_j) < 1$, is the boundary region of Y.*

Proof. It can be proved in a straightforward way according to definition and property of inverse flow graph and Definition 5. □

Example. Suppose we are given the flow graph for the preference analysis problem depicted in Fig. 1, that describes four disjoint models of cars $X = \{X_1, X_2, X_3, X_4\}$. They are sold to four disjoint groups of customers $Z = \{Z_1, Z_2, Z_3, Z_4\}$ through three dealers $Y = \{Y_1, Y_2, Y_3\}$.

By Definition 5, when we consider customer Z_1: the lower approximation of Z_1 is an empty set, the upper approximation of Z_1 is $Y_1 \cup Y_2$ and the boundary

[2] Where $-Y = U - Y$.

region Z_1 is $Y_1 \cup Y_2$. Hence, by Definition 6, we conclude that Z_1 is internally Y-indefinable. In Fig. 1 (only limited information is available), by using the set of dealers (Y) to approximate the customer group Z_1 together with the flow distribution visualized in layers two and three, our results can be summarized as the following.

- Since no branch connects Y_3 and Z_1, there is no customer Z_1 buys a car from dealer Y_3. As a result if dealer Y_3 plans to run new promotional campaigns, they do not need to pay attention to customer group Z_1 in these campaigns.
- If a customer buys a car through dealer Y_1 or Y_2, then we cannot conclude whether this is a customer in group Z_1 or not. Thus, if dealers Y_1 and Y_2 plan to run promotional campaigns, then they should, at least, target at customer group Z_1 in their campaigns.

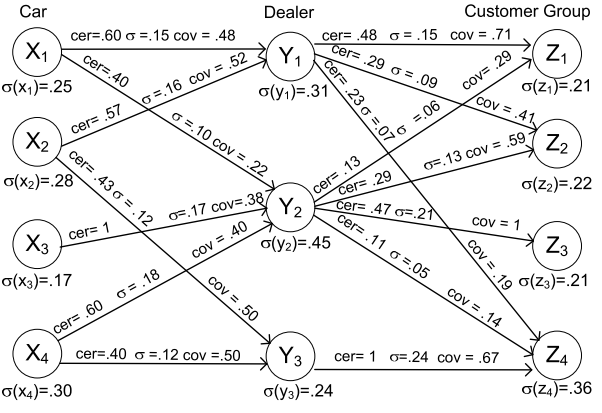


Fig. 1. A normalized flow graph

Similarly, we can approximate all attribute values (node) in the inverse flow graph of G by using Property 1.

However, the flow graph perspective on rough sets' categories in Definition 6 do not provide approximations quantitively. Hence, in Definitions 7 and 8, we define two measures for flow graphs, the accuracy of approximation and the roughness of approximation.

Definition 7. *Let* $G = (A, \mathcal{B}, \sigma)$ *be a flow graph,* $A_{l_i} = \left\{X_1, X_2, \ldots, X_{n_{l_i}}\right\}$, $1 \leq i \leq k-1$, *be an attribute in layer* i *and* Y *be a node in* $A_{l_{i+1}}$. *For any branch* (X_j, Y), $j \in \{1, \ldots, n_{l_i}\}$, *of* G, *the accuracy of approximation,* $\alpha_{A_{l_i}}(Y)$, *is defined as:* $\alpha_{A_{l_i}}(Y) = \frac{card(A_{l_i}(Y))}{card(\overline{A_{l_i}}(Y))}$.

We can use the accuracy of approximation to specify the quality of an approximation. Obviously, $0 \leq \alpha_B(X) \leq 1$. If $\alpha_{A_{l_i}}(Y) = 1$, then Y is crisp with respect to A_{l_i}, and otherwise, if $\alpha_{A_{l_i}}(Y) < 1$, then Y is rough with respect to A_{l_i}.

Definition 8. *Let* $G = (A, \mathcal{B}, \sigma)$ *be a flow graph,* $A_{l_i} = \left\{X_1, X_2, \ldots, X_{n_{l_i}}\right\}$, $1 \leq i \leq k-1$, *be an attribute in layer* i *and* Y *be a node in* $A_{l_{i+1}}$. *For any branch* (X_i, Y), $i \in \{1, \ldots, n_{l_i}\}$, *of* G, *the roughness of approximation,* $\gamma_{A_{l_i}}(Y)$, *is defined as:* $\gamma_{A_{l_i}}(Y) = 1 - \alpha_{A_{l_i}}(Y) = \frac{card(\overline{A_{l_i}}(Y)) - card(\underline{A_{l_i}}(Y))}{card(\overline{A_{l_i}}(Y))}$.

Property 2. *Let* $G = (A, \mathcal{B}, \sigma)$ *be a flow graph,* $A_{l_i} = \left\{X_1, X_2, \ldots, X_{n_{l_i}}\right\}$, $1 \leq i \leq k-1$, *be an attribute in layer* i *and* Y *be a node in* $A_{l_{i+1}}$. *For any branch* (X_j, Y), $j \in \{1, \ldots, n_{l_i}\}$, *of* G, *we have*

(1) $\alpha_{A_{l_i}}(Y) = \frac{\sum_{cer(X_j,Y)=1} \sigma(X_j)}{\sum_{X_j \in I(Y)} \sigma(X_j)}$ *and (2)* $\gamma_{A_{l_i}}(Y) = \frac{\sum_{cer(X_j,Y)<1} \sigma(X_j)}{\sum_{X_j \in I(Y)} \sigma(X_i)}$.

Proof. (1) From Definition 5, we have $card(\underline{A_{l_i}}(Y)) = \sum_{cer(X_j,Y)=1} card(X_j)$ and $card(\overline{A_{l_i}}(Y)) = \sum_{X_j \in I(Y)} card(X_i)$. Since $card(X_j) = \varphi(X_j) = \sigma(X_j)\varphi(G) = \sigma(X_j)\varphi(U)$ and by Definition 7, then $\alpha_B(Y) = \frac{\sum_{cer(X_j,Y)=1} \sigma(X_j)}{\sum_{X_j \in I(Y)} \sigma(X_j)}$.

(2) It can be proved similarly to (1). □

Let us briefly comment on Property 2(1) that the greater the boundary of Y, the lower is the accuracy. If $\alpha_{A_{l_i}}(Y) = 1$, the boundary region of Y is empty.

Property 3. *Let* $G = (A, \mathcal{B}, \sigma)$ *be a flow graph,* $A_{l_i} = \left\{X_1, X_2, \ldots, X_{n_{l_i}}\right\}$, $2 \leq i \leq k$, *be an attribute in layer* i *and* W *be a node in* $A_{l_{i-1}}$. *For any branch* (X_j, W), $j \in \left\{1, \ldots, n_{l_j}\right\}$ *in the inverse flow graph of* G, *we have*

(1) $\alpha_{A_{l_j}}(W) = \frac{\sum_{cov(W,X_j)=1} \sigma(X_j)}{\sum_{X_j \in O(W)} \sigma(X_j)}$ *and (2)* $\gamma_{A_{l_j}}(W) = \frac{\sum_{cov(W,X_j)<1} \sigma(X_j)}{\sum_{X_j \in O(W)} \sigma(X_j)}$.

Proof. (1) From Property 1, we have $card(\underline{A_{l_j}}(W)) = \sum_{cov(X_j,W)=1} card(X_j)$ and $card(\overline{A_{l_j}}(Y)) = \sum_{X_j \in O(W)} card(X_j)$. Since $card(X_j) = \varphi(X_j) = \sigma(X_j)\varphi(G) = \sigma(X_j)\varphi(U)$ and by Definition 7, then $\alpha_{A_{l_j}}(W) = \frac{\sum_{cer(X_j,W)=1} \sigma(X_j)}{\sum_{X_j \in O(W)} \sigma(X_j)}$.

(2) It can be proved similarly to (1). □

Example (Cont.) Consider the branches between dealer and customer group in Fig. 1. We can read from our flow graph that 24% of all customers buy cars through dealer Y_3 ($\sigma(Y_3) = 0.24$) and all of them are in customer group Z_3 ($cer(Y_3, Z_4) = 1$). There is only one branch (Y_3, Z_4) with $cer(Y_3, Z_4) = 1$[3]. Thus, by Property 2(1), we have $\alpha_Y(Z_1) = \alpha_Y(Z_2) = \alpha_Y(Z_3) = 0$ and $\alpha_Y(Z_4) = \frac{\sum_{cer(Y_i,Z_4)=1} \sigma(Y_i)}{\sum_{Y_i \in I(Z_4)} \sigma(Y_i)} = \frac{\sigma(Y_3)}{\sigma(Y_1)+\sigma(Y_2)+\sigma(Y_3)} = 0.24$.

These results imply that we should not make decisions involving customer groups Z_1, Z_2 and Z_3 solely by using dealers due to high imprecision. Nevertheless, we can partly check that it will be customer group Z_4 with low accuracy by

[3] By employing the approach presented in our previous study [3], we can extract some interesting association rules. If the model of car X_2 (or X_4) is bought through dealer Y_3 then the customer group is Z_4 with support 0.12 and confidence 1.

using dealers. Similarly, if we consider the roughness of approximation between dealer and customer group, then by Property 2(2), we have $\gamma_Y(Z_1) = \gamma_Y(Z_2) = \gamma_Y(Z_3) = 1$ and $\gamma_Y(Z_4) = 0.76$. We can draw a conclusion in a similar manner as we did for the roughness measure.

Please note that we can calculate the accuracy and the roughness of approximation between attributes in the inverse flow graph by using Property 3. Another important topic in data analysis is dependency between attributes. We introduce dependency degree between any two attributes in Definition 9.

Definition 9. *Let $G = (A, \mathcal{B}, \sigma)$ be a flow graph, $A_{l_i} = \left\{X_1, X_2, \ldots, X_{n_{l_i}}\right\}$ and $A_{l_{i+1}} = \left\{Y_1, Y_2, \ldots, Y_{n_{l_{i+1}}}\right\}$, $1 \leq i \leq k$, be any two adjacent layers. $A_{l_{i+1}}$ depends on A_{l_i} to a degree $k_{A_{l_i}}(A_{l_{i+1}}) = \frac{\sum_{l=1}^{n_{l_{i+1}}} card(\underline{A_{l_i}}(Y_l))}{card(U)}$.*

If $k_{A_{l_i}}(A_{l_{i+1}}) = 1$, we say that $A_{l_{i+1}}$ depends totally on A_{l_i}, and if $k_{A_{l_i}}(A_{l_{i+1}}) < 1$, we say that $A_{l_{i+1}}$ depends partially in a degree $k_{A_{l_i}}(A_{l_{i+1}})$ on A_{l_i}. It is worth pointing out that our dependency measure is different to the one given by Pawlak [14]. The former gives dependency degree between two adjacent attributes (layers) while the latter gives dependency degree between two nodes connected by directed branch.

Property 4. *Let $G = (A, \mathcal{B}, \sigma)$ be a flow graph, $A_{l_i} = \left\{X_1, X_2, \ldots, X_{n_{l_i}}\right\}$ and $A_{l_{i+1}} = \{X_1, X_2, \ldots, \quad X_{n_{l_{i+1}}}\}$, $1 \leq i \leq k$, be any two adjacent layers. $A_{l_{i+1}}$ depends on A_{l_i} to a degree $k_{A_{l_i}}(A_{l_{i+1}}) = \sum_{cer(X_i, X_j)=1} \sigma(X_i)$.*

Proof. From Definition 5, $\sum_{j=1}^{n_{l_{i+1}}} card(\underline{A_{l_i}}(X_j)) = \sum_{j=1}^{n_{l_{i+1}}} \sum_{cer(X_i,Y_j)=1} card(X_i)$. Since $X_n \cap X_m = \emptyset$, $1 \leq n \neq m \leq n_{l_i}$, then $\underline{A_{l_i}}(X_n) \cap \underline{A_{l_i}}(X_m) = \emptyset$. Thus $\sum_{j=1}^{n} card(\underline{A_{l_i}}(X_j)) = \sum_{cer(X_i,Y_j)=1} card(X_i)$. Since $\varphi(X_i) = \sigma(X_i)\varphi(G) = \sigma(X_i)\varphi(U)$ and by Definition 9, we can write $\gamma_B(D) = \sum_{cer(X_i,X_j)=1} \sigma(X_i)$. □

Property 5. *Let $G = (A, \mathcal{B}, \sigma)$ be a flow graph, $A_{l_j} = \left\{X_1, X_2, \ldots, X_{n_{l_j}}\right\}$ and $A_{l_{j-1}} = \{X_1, X_2, \ldots \ , X_{n_{l_{j-1}}}\}$, $1 \leq j \leq k+1$, be any two adjacent layers in the inverse flow graph of G. $A_{l_{j-1}}$ depends on A_{l_j} to a degree $k_{A_{l_j}}(A_{l_{j-1}}) = \sum_{cov(X_i,X_j)=1} \sigma(X_i)$.*

Proof. It can be proved similarly as Property 4. □

Example (Cont.) Consider model of car and dealer in the flow graph G in Fig. 1. By Property 4, dealer depends on model of car to a degree $\gamma_X(Y) = \sum_{cer(X_i,Y_j)=1} \sigma(X_i) = \sigma(X_3) = 0.17$. On the other hand, if we consider customer and dealer in the inverse flow graph of G, then by Property 5, we obtain that dealer depends on customer group to a degree $\gamma_Z(Y) = \sum_{cov(Y_i,Z_j)=1} \sigma(Z_i) = \sigma(Z_3) = 0.21$. These results give a conclusion that dealers depend on customer groups more than models of cars.

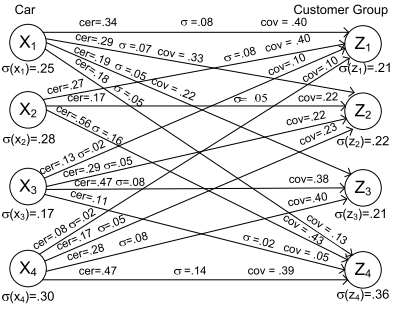


Fig. 2. A combined flow graph

In what follows, we aim to approximate a specific attribute value by some attribute values such that they are not in adjacent layers. We can use the concept of a *connection* to do this. More specifically, if we aim to approximate an attribute value in an output layer by attribute values in an input layer, then we will use the concept of *complete connection*.

Example (Cont.) For model of car and customer group in Fig. 1, we give a combined flow graph in Fig. 2. By Definition 5, for Z_4, the lower approximation of Z_4 is an empty set, the upper approximation and the boundary region of Z_4 are $X_1 \cup X_2 \cup X_3 \cup X_4$. Hence, by Definition 6, Z_4 is totally X-indefinable.

By Property 2, we have the accuracy and the roughness approximation of customer Z_4 by model of car as: $\alpha_X(Z_4) = 0$ and $\gamma_X(Z_4) = 1$. Additionally, we can use Property 4 to compute the dependency between model of car and customer group, and the result is 0. From these results due to the imprecision and dependency, we should not make decisions involving customer group Z_4 by using only model of car. As before, we can approximate and measure them for the inverse flow graph in the same way. Comparing the obtained accuracy and roughness measures, we can draw a conclusion that from this population *dealer* is a better indicator for analyzing customer group Z_4 than *model of car*.

5 Conclusion

In this paper, we introduce definitions and properties of rough set approximations, accuracy and roughness of approximation which are defined in terms of a flow graph. They can be useful when the initial data is in the form of flow graph and contains some limitations. We illustrate a car dealer preference analysis to support our propositions.

Acknowledgments

This paper was supported by the grant MRG5180071 from the Thailand Research Fund and the Commission on Higher Education and King Mongkut's Institute of Technology Ladkrabang. Thanks also to G.M. Zaverucha.

References

1. Butz, C.J., Yen, W., Yang, B.: The Computational Complexity of Inference Using Rough Set Flow Graphs. In: Ślęzak, D., Wang, G., Szczuka, M.S., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 335–344. Springer, Heidelberg (2005)
2. Czyzewski, A., Kostek, B.: Musical Metadata Retrieval with Flow Graphs. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 691–698. Springer, Heidelberg (2004)
3. Chitcharone, D., Pattaraintakorn, P.: Knowledge Discovery by Rough Sets Mathematical Flow Graphs and its Extension. In: Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, Innsbruck, Austria, pp. 340–345 (2008)
4. Chitcharone, D., Pattaraintakorn, P.: Towards Theories of Fuzzy Set and Rough Set to Flow Graphs. In: The 2008 IEEE World Congress on Computational Intelligence, pp. 1675–1682. IEEE Press, Hong Kong (2008)
5. Liu, H., Sun, J., Zhang, H.: Interpretation of Extended Pawlak Flow Graphs Using Granular Computing. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets VIII. LNCS, vol. 5084, pp. 93–115. Springer, Heidelberg (2008)
6. Matusiewicz, Z., Pancerz, K.: Rough Set Flow Graphs and Max $-*$ Fuzzy Relation Equations in State Prediction Problems. In: Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) RSCTC 2008. LNCS (LNAI), vol. 5306, pp. 359–368. Springer, Heidelberg (2008)
7. Pattaraintakorn, P., Cercone, N., Naruedomkul, K.: Rule Learning: Ordinal Prediction Based on Rough Set and Soft-Computing. Appl. Math. Lett. 19(12), 1300–1307 (2006)
8. Pattaraintakorn, P.: Entropy Measures of Flow Graphs with Applications to Decision Trees. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS, vol. 5589, pp. 618–625. Springer, Heidelberg (2009)
9. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
10. Pawlak, Z.: Rough Sets, Decision Algorithms and Bayes' Theorem. European J. of Oper. Res. 136, 181–189 (2002)
11. Pawlak, Z.: Rough Set and Flow Graphs. In: Ślęzak, D., Wang, G., Szczuka, M.S., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 1–11. Springer, Heidelberg (2005)
12. Pawlak, Z.: Inference Rules and Decision Rules. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 102–108. Springer, Heidelberg (2004)
13. Pawlak, Z.: Decision Trees and Flow Graphs. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 1–11. Springer, Heidelberg (2006)
14. Pawlak, Z.: Flow Graphs and Data Mining. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 1–36. Springer, Heidelberg (2005)
15. Pawlak, Z., Skowron, A.: Rudiments of Rough Sets. Inform. Sciences 177, 3–20 (2007)

Credibility Coefficients Based on SVM

Roman Podraza and Bartosz Janeczek

Institute of Computer Science
Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
R.Podraza@ii.pw.edu.pl

Abstract. ARES System was a data analysis tool supporting Rough Set theory. It has been expanded to cover other approaches like Emerging Patterns and Support Vector Machine. A special feature of ARES System is ability to identify exceptional objects within information systems by using credibility coefficients. The credibility coefficient is a measure, which attempts to weigh up a degree of typicality of each object in respect to the rest of information system. The paper presents an idea of credibility coefficients based on SVM approach. The new coefficients are compared with the others ones available in the ARES System.

Keywords: credibility coefficients, information system, classification, outliers, SVM.

1 Introduction

Credibility of data is a necessary condition to have confidence to results or drawn conclusions. In data analysis a credibility of data has to be taken into consideration to identify outliers - the data with uncertain reliability. The outliers can be removed from the data set, corrected or left untouched. Recognition of the doubtful data can be done only if the domain of data is known and understood. To identify outliers automatically each object has to get its credibility coefficient in the context of the whole information system [1] [2]. In general no interpretation of data is possible. It is assumed that the majority of data set is correct and only a minor part of it can be suspected to be improper. This approach to credibility analysis is universal and can be applied to any information system regardless of its domain.

The main purpose of defining the credibility coefficients is to find out exceptions to rules because very often they can be more interesting than the rules themselves. For instance, in medicine it would be invaluable to automatically detect such cases, where a treatment gives results beyond the expected sphere defined by some rules. Then potentially the most difficult and interesting cases are identified and extracted from maybe overwhelming set of routine ones.

Credibility coefficients are a part of capabilities of ARES System [2], which is a common platform for different data analysis approaches. Its initial functionality was based on Rough Set (RS) theory [3], then it has been expanded by Emerging Patterns (EP) approach [4] and Support Vector Machine (SVM) methodology [5] [6] [7].

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 428–437, 2010.

In the next section an overview of ARES System is sketched. A section describing concept of credibility coefficients is followed by three proposals of credibility coefficients based on SVM. Then some practical results are outlined and finally conclusions complete the paper.

2 ARES System

The system has been designed to give an interactive access to process of data analysis involving different approaches. Hierarchically organized items in a directory browser enable observing and/or comparing different phases of data analysis.

ARES System processes information systems or decision tables with single decision attribute. Editing functionalities enable cutting off information system by removing objects (rows) and/or attributes (columns). There are available capabilities of data discretization by a number of methods.

The initial version of ARES System [1] comprised modules for performing the following tasks from Rough Set domain such as discovering approximations of decision classes, determining discernibility matrices, finding relative reducts, discovering frequents sets and mining decision rules. The domain of Rough Set theory in ARES System has been supplemented by module for discovering discriminant of information system by algorithms LEM1, LEM2 and AQ [8].

ARES System has incorporated the KTDA system [9] based on Emerging Patterns (EP) approach. Two different algorithms of discovering EPs are supported– using maximal frequent itemsets proposed in [4] or using decision tree. The former one reflects the classical approach and requires stating minimal growth rate and minimal support in the target class, while the latter one uses Fisher's Exact Test used to discover only such EPs which are statistically significant. EPs enable data classification for which CAEP (Classification by Aggregating Emerging Patterns) algorithm is applied.

Support Vector Machines is yet another methodology being integrated to ARES System to evaluate credibility coefficients, which are presented in detail in the chapters following a general description of credibility analysis approach.

3 Credibility Coefficients

Credibility coefficients define relative measures of credibility of all objects from information system. The domain of credibility coefficients is interval [0; 1], but their values assessed by different methods are incomparable. Credibility coefficients should be used to introduce a ranking of objects from information system.

A number of methods of evaluation of credibility coefficients are available [1] [2], but they can produce distinctive results. It is difficult to draw a common conclusion from applying several credibility approaches because their aggregations have no meaning (like an average value for set of physical values with various units of measure). Still there are a lot of questions regarding meaning, applications and efficiency of credibility coefficients.

In general the credibility analysis should find out typical schemas of data and dependencies between them or statistical relations appearing for analyzed data set (information system). To automatically identify a particular object as an outlier, its

inferior regularity in respect to other objects should be exposed. So we try to identify a non-typicality, presuming that if an object is less typical, so it does not precisely conform to knowledge discovered in the data set. The object not supporting relations observed frequently enough should receive lower evaluation of typicality.

Currently in ARES System a number of algorithms for calculations of credibility coefficients have been implemented. Heuristics of the algorithms were based on the following concepts:

- Approximation of Rough Set classes,
- Statistics of attribute values,
- Hybrid one combining the previous two,
- Frequent Sets,
- Extracted Rules (Rough Set approach),
- Voting Classifier (CAEP - for Emerging Patterns),
- Support Vector Machines.

Some experiments with credibility coefficients presented their ability to identify corrupted data "injected" into original data sets; however it is difficult to prove superiority of particular approaches over others. The effectiveness of different credibility coefficients vary depending on their applications (e.g. identifying falsified data, incrementing measures of quality indicators or discovering new and/or "better" rules by removing the most improper data) and the benchmarks (information systems).

Credibility coefficients based on SVM methodology can be evaluated with support of the following algorithms:

- *C*-SVM (classification),
- *v*-SVM (classification),
- ε-SVR (regression),
- *v*-SVR (regression),
- SVM clustering algorithm.

A kernel function for the SVM algorithms can be chosen out of four: linear, polynomial, RBF (Radial Basis Function) and sigmoidal. For algorithm *C*-SVM with RBF used as kernel function there is an option to automatically adjust the parameters by performing series of tests with cross validations.

4 Credibility Coefficients Based of SVC

We proposed credibility coefficient, which is based on Support Vector Classifier. The following formula

$$f_d(\mathbf{x}) = \mathbf{w}_d^T \mathbf{x} + b_d \tag{1}$$

denotes function evaluating adherence of object $\mathbf{x}$ to category d, where
$d \in \boldsymbol{V}_{dec}$ is a value of decision attribute from its domain V_{dec},
$\mathbf{x} \in \boldsymbol{R}^p$ is an object from information system with p attributes,
$\mathbf{w} \in R^p$ is normal vector to p-dimensional hyperplane,
$b_d \in R$

We have a sequence of classification tasks performed for each category *d*. Formula (1) is a general case and its dual form using Lagrange function is used for calculation of credibility coefficients.

To obtain the required range of credibility coefficients ([0; 1]), we apply the following algorithm. In the first step we calculate a domination vector according to the formula

$$p_k(i) = Score(i,d) - Score(i,k) \tag{2}$$

where $p_k(i)$ is *k*-th element of domination vector for object $i \in I$, *Score (i, d)* is the measure of adherence to category *d* (the object belongs to this category) and *Score (i, k)* is the measure of adherence to category $k \neq d$ (the domination vector has its dimension smaller by 1 than number of all categories).

Then we calculate an auxiliary coefficient using elements of domination vector

$$t(i) = \begin{cases} \min\limits_k p_k(i) & if \ \forall\limits_k p_k(i) \geq 0 \\ -\sqrt{\sum\limits_k (p_k(i))^2} & if \ \exists\limits_k p_k(i) < 0 \end{cases} \tag{3}$$

This coefficient is positive if the measure of adherence is greatest for the category, to which objects belongs and the value is equal to minimal domination over any other category. The coefficient is equal to 0 if there is another category in which measure of adherence is the same as for its own category. The negative value means that there is at least one category, in which the measure of adherence is greater than for its own category. Each such category affects the value of this auxiliary coefficient. Finally the values of these auxiliary coefficients are mapped into predefined range ([0; 1]) of all credibility coefficients. It is done by applying the formula

$$cred_{SVC}(i) = \begin{cases} \dfrac{t(i)}{2t_{\max}} + \dfrac{1}{2} & if \ t(i) > 0 \\ \dfrac{1}{2} & if \ t(i) = 0 \\ -\dfrac{t(i)}{2t_{\min}} + \dfrac{1}{2} & if \ t(i) < 0 \end{cases} \tag{4}$$

where t_{max} and t_{min} are maximum and minimum of auxiliary coefficients, respectively.

This way of evaluating credibility coefficients has an important drawback. The values of credibility coefficients for objects, which have intuitively high evaluation of adherence to the given category, may vary significantly, because they are strongly correlated with distance to the hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$. For a two-dimensional case, a single point with significantly greater distance from the separating line could significantly lower the values of credibility coefficients of other points properly classified, but lying closer to the line.

To avoid this problem we decided to use algorithms evaluating probability of adherence to a particular category. They are based, as previously presented method on values of $f_d(\mathbf{x})$ from formula (1), but their interpretation is different.

To assess probability of adherence of an object to a particular category firstly we consider probability of adherence (of the object) to one out of two categories.

$$p_{mn}(i) = p(y = d_m \mid y = d_m \vee y = d_n, i), \quad d_m, d_n \in V_D \tag{5}$$

denotes probability for object $i \in I$ (information system) to adhere to category d_m, taking into account only categories d_m and d_n. This probability may be assessed by

$$p_{nm}(i) \approx \frac{1}{1 + e^{-f_{mn}(\mathbf{x})}} \tag{6}$$

where f_{mn} denotes hyperplane separating objects from categories d_m and d_n, $\mathbf{x}$ is vector of attributes of object $i \in I$.

After Platt modification [10] introducing coefficients A and B we have

$$p_{nm}(i) \approx \frac{1}{1 + e^{Af_{mn}(\mathbf{x})+B}} \tag{7}$$

To adjust values of coefficients A and B, we are looking for a solution with the greatest likelihood by minimization of approximation error for all objects $i \in I$. To achieve it a new variable is introduced:

$$t_i = t_{nm}(i) = \frac{y_i + 1}{2} = \begin{cases} 0 & for \ y_i = d_n = -1 \\ 1 & for \ y_i = d_m = 1 \end{cases} \tag{8}$$

where y_i denotes category of object i: 1 is set for category d_m and -1 is set for category d_n. In a similar manner, to simplify denotation, by $p_i = p_{mn}(i)$ we represent probability estimated by the algorithm. Finally values of A and B are calculated by minimizing measure of Kullback-Leibler distance describing differences in probability distributions – required one and estimated by the algorithm (t and p)

$$\min_{A,B} - \sum_i (t_i \log(p_i) + (1 - t_i) \log(1 - p_i)) \tag{9}$$

When values of A and B are set and all values of $p_{mn}(i)$ are calculated it is possible to calculate probability of adherence of object $i \in I$ to category d_m [11]. The following relationship resulting from Bayes theorem is used for it.

$$\begin{aligned} & p(y = d_m \mid y = d_m \vee y = d_n, i) \cdot p(y = d_n \mid i) = \\ & = p(y = d_n \mid y = d_m \vee y = d_n, i) \cdot p(y = d_m \mid i) \\ & or \ otherwise \\ & p_{mn}(i) \cdot p_n(i) = p_{nm}(i) \cdot p_m(i) \end{aligned} \tag{10}$$

$$\begin{aligned} & \min_{p_m, p_n} \frac{1}{2} \sum_m \sum_{n:n \neq m} (p_{nm}(i) \cdot p_m(i) - p_{mn}(i) \cdot p_n(i))^2 \\ & where \\ & m\sum p_m = 1 \ and \ \bigvee_m p_m \geq 0 \end{aligned} \tag{11}$$

To keep equity **(10)** (or to be as close as possible) the optimizing task **(11)** is solved.

The algorithm is implemented in library *libsvm* [12]. Of course, values of credibility coefficients are dependent on parameters of Support Vector Classifier and in particular on chosen kernel function and its parameters.

5 Credibility Coefficients Based of SVR

The next version of credibility coefficient is based on Support Vector Regression. Regression function $f(\mathbf{x})$ estimates value of decision y of object i with accuracy to error z

$$y_i = f(\mathbf{x}_i) + z_i \tag{12}$$

where z_i are independent random variables with the same probability distribution. This distribution is dependent on values of error functions ξ_i and ξ^*_i, defined as

$$\begin{aligned} \xi_i &= (y_i - f(\mathbf{x}_i)) - \varepsilon_i \quad for\ \ y_i > f(\mathbf{x}_i) + \varepsilon_i \\ \xi_i^* &= (f(\mathbf{x}_i) - y_i) - \varepsilon_i \quad for\ \ y_i < f(\mathbf{x}_i) - \varepsilon_i \end{aligned} \tag{13}$$

where ε_i is the error margin.

In [13] a model of the probability distribution exploiting Laplace distribution with probability density $p(z)$ was proposed

$$p(z) = \frac{1}{2\sigma} e^{-\frac{|z|}{\sigma}} \tag{14}$$

where σ is a scale parameter evaluated with values of error functions

$$\sigma = \frac{\sum_{i=1}^{l} (\xi_i + \xi_i^*)}{l} \tag{15}$$

where $i \in \{1, \ldots, l\}$ denotes a number (index) of successive object from the information system. Knowing the real category y_i and value of regression function $f(\mathbf{x}_i)$ it is possible to set $z_i = y_i - f(\mathbf{x}_i)$. The object without error should have credibility equal to 1 and for greater absolute value of z the credibility coefficient is smaller.

$$cred_{SVR}(i) = 1 - \int_{-|z|}^{|z|} p(z)dz = 1 - \int_{-|z|}^{|z|} \frac{1}{2\sigma} e^{-\frac{|z|}{\sigma}} dz \tag{16}$$

Similarly to values of credibility coefficients based on Support Vector Classifier the calculated values are correlated to kernel functions and its parameters.

In an example presenting Support Vector Regression approach (Table 1) there were generated 12 objects with conditional attribute x and decision attribute y. Algorithm ν-SVR (with kernel function RBF, $\nu = 0.5$) was used to convey credibility analysis and as a result σ was set to 0.0656.

Table 1. Credibility coefficients based on SVR

i	*x*	*y*	*f(x)*	*z*	*cred(i)*
1	0.036	0.182	0.228	-0.046	49.6 %
2	0.080	0.274	0.276	-0.002	97.0 %
3	0.110	0.484	0.304	0.180	6.4 %
4	0.172	0.352	0.346	0.006	91.2 %
5	0.268	0.248	0.379	-0.131	13.6 %
6	0.308	0.376	0.384	-0.008	88.5 %
7	0.448	0.386	0.384	0.002	97.0 %
8	0.576	0.394	0.399	-0.005	92.7 %
9	0.676	0.458	0.449	0.009	87.2 %
10	0.792	0.562	0.568	-0.006	91.3 %
11	0.874	0.702	0.695	0.007	89.9 %
12	0.938	0.864	0.858	0.006	91.2 %

6 Credibility Coefficients Based on Clustering

The last method of credibility analysis based on SVM methodology employs algorithm of clustering. So far there is no probability approach to assess exactness of clustering and this is the main difference between this approach and the previous two. Clustering algorithm returns only binary information, whether a particular objects belongs to a cluster or not. Credibility coefficients have to be evaluated for all objects without any prior knowledge about them and this assumption excludes any training of the algorithm on a subset of the credible data, which is quite typical for the clustering.

We propose to perform N attempts of clustering with different parameters and credibility coefficient for object i can be worked out as

$$cred_{SVMC}(i) = \frac{\sum_{n=1}^{N} clust_n(i)}{N} \tag{17}$$

where $clust_n(i)$ denotes the result of clustering in n-th attempt (the result is 1 if the object belongs to the cluster or 0 otherwise).

$$clust(i) = \begin{cases} 1 & if \;\; f(\mathbf{x}) > 0 \\ 0 & if \;\; f(\mathbf{x}) \le 0 \end{cases} \tag{18}$$

The other approach takes M attempts of clustering with the same for a subset of objects with proportion $(M\text{-}1)/M$ of all objects, and each object is clustered M-1 times.

$$cred_{SVMC}(i) = \frac{\sum_{m=1}^{M} clust_m(i)}{M-1} \tag{19}$$

In general all three proposed methods of evaluation of credibility coefficients produce similar results – the values of credibility coefficients are different, but all methods identify the same objects as the least credible.

7 Experiment

The experiment was carried out for classic data set iris, containing descriptions of 150 iris blossoms with 4 conditional attributes (*sepallength sepalwidth, petallength, petalwidth*) and belonging to 3 categories (*setosa*, *versicolor*, *virginica*) [14]. In the first phase the data set was enlarged by 6 new improper objects (2 for each category) with attribute values significantly different than original objects. Then next 15 randomly chosen original objects had their original category replaced by another one and were added again to the data set. And for such prepared data different credibility coefficients were calculated. In the second phase all attributes of objects from information system were discretized (to get approximately 20 discrete values for domain of each attribute) and then processed as in the phase one. The following credibility coefficients were evaluated:

- $cred_{RS}$ – based on rough set class approximation,
- $cred_{FS}$ – based on frequent sets (with parameter: "Minimal Support" = 10%),
- $cred_{RB}$ – based on decision rules (with parameters: "Minimal Support" = 5%, "Minimal Confidence" = 10%),
- $cred_{EP}$ – based on emerging patterns applying decision tree (with parameters: "Split Significance Level" = 10%, "EP Significance Level" = 5%),
- $cred_{SVC}$ – based on classifier *C*-SVC (with kernel function RBF with parameters set automatically),
- $cred_{SVR}$ – based on regression analysis ε-SVC (with kernel function RBF, $C = 10$, $\varepsilon = 0.1$, $\gamma = 0$),
- $cred_{SVMC}$ – based on SVM clustering (with kernel function RBF, $C = 10$, $v = 0.05$, $\gamma = 0$).

The results of the experiment is presented in Table 2 and Table 4, where average values of credibility coefficients were grouped for original objects (correct ones – group M_1), improper objects (these produced artificially before calculations – group M_0) and for all objects (group M_{10}). Table 2 and Table 4 present the results for original and discretized values of objects' attributes, respectively.

The values of credibility coefficients should have a meaningful interpretation. A threshold should be somehow set to interpret data as sufficiently credible or improper. This arbitrary decision may be based on a purpose of credibility analysis. We may be interested in identifying a predefined number or a predefined ratio of the worst cases.

Table 2. Average values of credibility coefficients for the extended set *iris*

	$cred_{RS}$	$cred_{FS}$	$cred_{RB}$	$cred_{EP}$	$cred_{SVC}$	$cred_{SVR}$	$cred_{SVMC}$
M_1	94 %	94 %	90 %	81 %	81 %	62 %	97 %
M_0	54 %	86 %	61 %	30 %	27 %	17 %	76 %
M_{10}	89 %	93 %	87 %	75 %	75 %	57 %	94 %

We may try to identify all doubtful cases (maybe with a small margin of proper ones). Anyway for the experiment the thresholds were defined as a half of the average value of credibility coefficient in group M_{10}. This value was necessary to evaluate coefficients of classification quality like accuracy (*Acc.*) and precision (*Prec.*) from confusion matrices. These values and other indicators of classification quality like Somer's D statistic and area under curve (*Auc*) for ROC (Receiver Operating Characteristics) curve for credibility analysis results from Table 2 are presented in Table 3.

Table 3. Coefficients of classification quality for the extended set *iris*

	RS	***FS***	***RB***	***EP2***	***SVC***	***SVR***	***SVMC***
Acc.	88 %	88 %	89 %	93 %	96 %	86 %	88 %
Prec.	88 %	88 %	92 %	95 %	96 %	96 %	90 %
Somer's D	0.679	0.202	0.442	0.796	0.886	0.762	0.205
Auc	0.839	0.601	0.721	0.898	0.943	0.881	0.602

Table 4. Average values of credibility coefficients for the discretized extended set *iris*

	$cred_{RS}$	$cred_{FS}$	$cred_{RB}$	$cred_{EP}$	$cred_{SVC}$	$cred_{SVR}$	$cred_{SVMC}$
M_1	94 %	77 %	89 %	83 %	77 %	81 %	31 %
M_0	54 %	55 %	51 %	22 %	25 %	33 %	66 %
M_{10}	89 %	74 %	84 %	75 %	71 %	75 %	35 %

Table 5. Coefficients of classification quality for the discretized extended set *iris*

	RS	***FS***	***RB***	***EP2***	***SVC***	***SVR***	***SVMC***
Acc.	89 %	89 %	89 %	95 %	96 %	94 %	31 %
Prec.	96 %	89 %	93 %	97 %	97 %	95 %	77 %
Somer's D	0.685	0.712	0.546	0.959	0.818	0.695	0.360
Auc	0.842	0.856	0.773	0.979	0.909	0.848	0.680

Discretization of data set *iris* improved discriminating capabilities of credibility coefficients based on frequent sets. The results of algorithms based on SVM became slightly worse except the method based on SVM clustering, which anyway yielded inferior results in both phases compared to other SVM counterparts.

8 Conclusions

The paper presents concepts of credibility coefficients based on SVM algorithms. They were implemented in ARES System completing the other ones based on the Rough Set theory and Emerging Patterns.

Credibility coefficients based on SVM were applied to specially prepared information system along with some other credibility coefficients. Their usefulness was assessed by some quality indicators. The new credibility coefficients appeared to be very effective in comparison to the others, maybe with exception of the algorithm

based on SVM clustering. In some situations only credibility coefficients based on SVC and SVR were able to produce reasonable results – e.g. for a small number of objects with continuous attributes. In particular credibility coefficients employing *C*-SVM algorithm with automatic parameter settings presented itself as the best one. This approach can be dominating in a universal usage. There is anyway a price for it. Processing of automatic parameter settings is many times longer than execution of the SVM algorithm. The values have to be chosen from a predefined domain and the constant step of scanning the domain may be not precise. These drawbacks set our sights to further research. A model of adaptive determining values of the parameters along with a progress of evaluation of the coefficients seems to be very interesting and promising. The other field requiring some more effort is methodology based on SVM clustering where the results were obviously worse than other SVM approaches. Still the concept has to be verified in further research and practical experiments.

References

[1] Podraza, R., Dominik, A.: Problem of Data Reliability in Decision Tables. Int. J. of Information Technology and Intelligent Computing (IT&IC) 1(1), 103–112 (2006)

[2] Podraza, R., Walkiewicz, M., Dominik, A.: Credibility Coefficients in ARES Rough Set Exploration System. In: Ślęzak, D., Yao, J., Peters, J.F., Ziarko, W.P., Hu, X. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3642, pp. 29–38. Springer, Heidelberg (2005)

[3] Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)

[4] Dong, G., Li, J.: Efficient Mining of Emerging Patterns: Discovering Trends and Differences. In: Proc. of the SIGKDD 5th ACM Int. Conf. on Knowledge Discovery and Data Mining, San Diego, USA, pp. 43–52 (1999)

[5] Schölkopf, B., Smola, A.: Learning with Kernels. MIT Press, Cambridge (2002)

[6] Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)

[7] Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)

[8] Grzymała-Busse, J.W.: Rule Induction. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 277–295. Springer Sci. + Business Media Inc. (2005)

[9] Podraza, R., Tomaszewski, K.: Ordinal credibility coefficient - a new approach in the data credibility analysis. In: An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.) RSFDGrC 2007. LNCS (LNAI), vol. 4482, pp. 190–198. Springer, Heidelberg (2007)

[10] Platt, J.: Probabilities for SV Machines. In: Smola, A., Bartlett, P., Schölkopf, B., Schuurmans, D. (eds.) Advances in Large Margin Classifiers, MIT Press, Cambridge (2000), `http://research.microsoft.com/~jplatt/SVMprob.ps.gz`

[11] Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. Journal of Machine Learning Research 5 (2004), `http://www.csie.ntu.edu.tw/~cjlin/papers/svmprob/svmprob.pdf`

[12] Chang, C., Lin, C.: LIBSVM: a library for support vector machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

[13] Lin, C.-J., Weng, R.C.: Simple probabilistic predictions for support vector regression, Technical Report, Department of Computer Science, National Taiwan University (2004), `http://www.csie.ntu.edu.tw/~cjlin/papers/svrprob.pdf`

[14] UCI Machine Learning Repository, School of Information and Computer Science, University of California, USA, `http://www.ics.uci.edu/~mlearn/MLRepository`

On Algorithm for Building of Optimal α-Decision Trees

Abdulaziz Alkhalid, Igor Chikalov, and Mikhail Moshkov*

Mathematical and Computer Sciences & Engineering Division
King Abdullah University of Science and Technology
Thuwal 23955-6900, Saudi Arabia
{abdulaziz.alkhalid,igor.chikalov,mikhail.moshkov}@kaust.edu.sa

Abstract. The paper describes an algorithm that constructs approximate decision trees (α-decision trees), which are optimal relatively to one of the following complexity measures: depth, total path length or number of nodes. The algorithm uses dynamic programming and extends methods described in [4] to constructing approximate decision trees. Adjustable approximation rate allows controlling algorithm complexity. The algorithm is applied to build optimal α-decision trees for two data sets from UCI Machine Learning Repository [1].

Keywords: Decision tree, dynamic programming, algorithm complexity.

1 Introduction

Decision trees are widely used for representing knowledge and as algorithms in test theory [3], rough set theory [7,8], machine learning and data mining [2]. These applications pay attention to model complexity to make it understandable and to prevent model overfitting to training data. There are several complexity measures: depth and total path length of decision tree nominally characterize work time, while number of nodes characterizes space required to store the model. For many applications several complexity measures are relevant. For example, successful model interpretation requires number of nodes to be limited and the major cases to be described by a reasonably short path in the tree.

For many cases the problem of building optimal decision tree is known to be NP-hard. However, there are special types for problems for which there exists a polynomial algorithm. In [4] an algorithm is presented that finds a set of optimal decision trees and allows for sequential optimization relatively to different complexity measures. The algorithm uses dynamic programming methods in order to be computationally effective. The paper [4] provides necessary conditions for the number of induced subproblems to be limited by a polynomial on decision table size that guarantees polynomial complexity of the algorithm.

* The research has been partially supported by KAUST-Stanford AEA project "Predicting the stability of hydrogen bonds in protein conformations using decision-tree learning methods".

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 438–445, 2010.

In this paper, we study possibilities of applying a similar algorithm to an arbitrary problem. The exact solution cannot be found except for small decision tables as the number of subproblems grows exponentially. The results of [5] imply that under reasonable assumptions there are no good approximate algorithms of polynomial complexity for exact decision trees. To overcome these limitations we introduce an uncertainty measure that is the number of unordered pairs of rows with different decisions in the table. Then we consider α-decision trees that do not solve the problem exactly but localize each row in a subtable with uncertainty at most α. The parameter α controls computational complexity and makes the algorithm applicable to solving complex problems.

The rest of the paper is organized as follows. Section 2 gives notions of decision table and irredundant α-decision tree. Section 3 gives a way of representing a set of trees in a form of directed acyclic graph. Section 4 introduces notion of complexity measure and describes a procedure of finding α-decision trees that are optimal relatively to different complexity measures. Section 5 contains experimental results that show dependence of algorithm complexity on α.

2 Basic Notions

Consider a *decision table* T depicted in Figure 1.

f_1	...	f_m	d
δ_{11}	...	δ_{1m}	c_1
	...		...
δ_{N1}	...	δ_{Nm}	c_N

Fig. 1. Decision table

Here $f_1, \ldots, f_m$ are names of columns (conditional attributes); $c_1, \ldots, c_N$ are nonnegative integers which can be interpreted as decisions (values of the decision attribute d); δ_{ij} are nonnegative integers which are interpreted as values of conditional attributes (we assume that the rows $(\delta_{11}, \ldots, \delta_{1m}), \ldots, (\delta_{N1}, \ldots, \delta_{Nm})$ are pairwise different). Let $f_{i_1}, \ldots, f_{i_t} \in \{f_1, \ldots, f_m\}$ and $a_1, \ldots, a_t$ be nonnegative integers. Denote by $T(f_{i_1}, a_1) \ldots (f_{i_t}, a_t)$ the subtable of the table T, which consists of such and only such rows of T that on the intersection with columns $f_{i_1}, \ldots, f_{i_t}$ have numbers $a_1, \ldots, a_t$ respectively. Such nonempty tables (including the table T) will be called *separable subtables* of the table T. For a subtable Θ of the table T we will denote by $R(\Theta)$ the number of unordered pairs of rows that are labeled with different decisions. Later we will interpret the value $R(\Theta)$ as *uncertainty* of the table Θ.

Let Θ be a nonempty subtable of T. A minimum decision value which is attached to the maximum number of rows in Θ will be called *the most common decision* for Θ.

A *decision tree* Γ *over* the table T is a finite directed tree with root in which each terminal node is labeled with a decision. Each nonterminal node is labeled

with a conditional attribute, and for each nonterminal node the outgoing edges are labeled with pairwise different nonnegative integers.

Let v be an arbitrary node of the considered decision tree Γ. Let us define a subtable $T(v)$ of the table T. If v is the root then $T(v) = T$. Let v be not the root, and in the path from the root to v, nodes be labeled with attributes $f_{i_1}, \ldots, f_{i_t}$, and edges be labeled with numbers $a_1, \ldots, a_t$ respectively. Then $T(v) = T(f_{i_1}, a_1), \ldots, (f_{i_t}, a_t)$.

Let α be a nonnegative real number. We will say that Γ is an *α-decision tree for T* if for each row r of the table T there exists a terminal node v of the tree such that r belongs to $T(v)$, v is labeled with the most common decision for $T(v)$ and $R(T(v)) \leq \alpha$.

Denote by $E(T)$ the set of attributes (columns of the table T), each of which contains different numbers. For $f_i \in E(T)$ let $E(T, f_i)$ be the set of numbers from the column f_i.

Among α-decision trees for the table T we distinguish *irredundant* α-decision trees. Let v be a node of an irredundant α-decision tree Γ. If $R(T(v)) \leq \alpha$ then v is a terminal node labeled with the most common decision for $T(v)$. Let $R(T(v)) > \alpha$. Then the node v is labeled with an attribute $f_i \in E(T(v))$. If $E(T(v), f_i) = \{a_1, \ldots, a_t\}$ then t edges leave node v, and these edges are labeled with $a_1, \ldots, a_t$ respectively. We denote by $D_\alpha(T)$ the set of all irredundant α-decision trees for the table T.

3 Representation of the Set of Irredundant α-Decision Trees

Consider an algorithm for construction of a graph $\Delta_\alpha(T)$, which represents in some sense the set $D_\alpha(T)$. Nodes of this graph are some separable subtables of the table T. During each step we process one node and mark it with symbol *. We start with the graph that consists of one node T and finish when all nodes of the graph are processed.

Let the algorithm have already performed p steps. Let us describe the step $(p + 1)$. If all nodes are processed then the work of the algorithm is finished, and the resulted graph is $\Delta_\alpha(T)$. Otherwise, choose a node (table) Θ that has not been processed yet. Let b be the most common decision for Θ. If $R(\Theta) \leq \alpha$, label the considered node with b, mark it with symbol * and proceed to the step $(p + 2)$. Let $R(\Theta) > \alpha$. For each $f_i \in E(\Theta)$ draw a bundle of edges from the node Θ. Let $E(\Theta, f_i) = \{a_1, \ldots, a_t\}$. Then draw t edges from Θ and label these edges with pairs $(f_i, a_1), \ldots, (f_i, a_t)$ respectively. These edges enter to nodes $\Theta(f_i, a_1), \ldots, \Theta(f_i, a_t)$. If some of nodes $\Theta(f_i, a_1), \ldots, \Theta(f_i, a_t)$ do not present in the graph then add these nodes to the graph. Mark the node Θ with symbol * and proceed to the step $(p + 2)$.

Now for each node of the graph $\Delta_\alpha(T)$ we describe the set of α-decision trees corresponding to it. It is clear that $\Delta_\alpha(T)$ is a directed acyclic graph. A node of such graph will be called *terminal* if there are no edges leaving this node. We will move from terminal nodes, which are labeled with numbers, to the node T. Let Θ

Fig. 2. Trivial α-decision tree

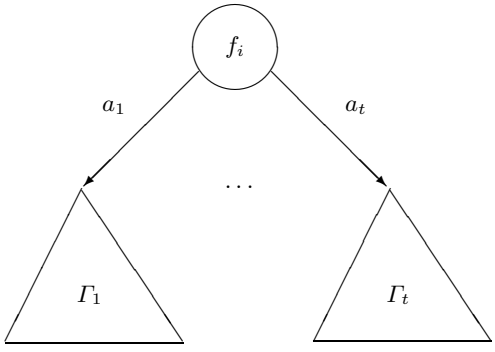


Fig. 3. Aggregated α-decision tree

be a node, which is labeled with a number b. Then the only trivial α-decision tree depicted in Figure 2 corresponds to the considered node. Let Θ be a node (table), for which $R(\Theta) > \alpha$. Let $f_i \in E(\Theta)$ and $E(\Theta, f_i) = \{a_1, \ldots, a_t\}$. Let $\Gamma_1, \ldots, \Gamma_t$ be α-decision trees from sets corresponding to the nodes $\Theta(f_i, a_1), \ldots, \Theta(f_i, a_t)$. Then the α-decision tree depicted in Figure 3 belongs to the set of α-decision trees, which corresponds to the node Θ. All such α-decision trees belong to the considered set. This set does not contain any other α-decision trees.

The following proposition shows that the graph $\Delta_\alpha(T)$ represents all irredundant α-decision trees for the table T.

Proposition 1. *Let T be a decision table and Θ a node in the graph $\Delta_\alpha(T)$. Then the set of α-decision trees corresponding to Θ coincides with the set $D_\alpha(\Theta)$ of all irredundant α-decision trees for the table Θ.*

4 Selecting Optimal α-Decision Trees

In this section, we introduce some notions and give the procedure of finding a set of optimal α-decision trees.

4.1 Proper Subgraphs of Graph $\Delta_\alpha(T)$

Let us introduce the notion of *proper* subgraph of the graph $\Delta_\alpha(T)$. For each node of the graph $\Delta_\alpha(T)$, which is not terminal, we can remove any but not all bundles that leave the node. Further we remove all nodes such that there are no directed paths to the considered node from the node T. Denote the obtained subgraph by G. Such subgraphs will be called proper subgraphs of the graph $\Delta_\alpha(T)$. It is clear that all terminal nodes of G are terminal nodes of the graph $\Delta_\alpha(T)$. As it was described earlier, we can associate a set of α-decision trees to each node Θ of G. It is clear that all these α-decision trees belong to the set $D_\alpha(\Theta)$. We denote this set of α-decision trees by $D_{\alpha,G}(\Theta)$.

4.2 Complexity Measures

We will consider complexity measures which are given in the following way: values of considered complexity measure ψ, which are nonnegative numbers, are defined by induction on pairs (T, Γ), where T is a decision table and Γ is an α-decision tree for T. Let Γ be an α-decision tree represented in Figure 2. Then $\psi(T, \Gamma) = \psi^0$ where ψ^0 is a nonnegative number. Let Γ be an α-decision tree depicted in Figure 3. Then

$$\psi(T, \Gamma) = F(N(T), \psi(T(f_i, a_1), \Gamma_1), \ldots, \psi(T(f_i, a_t), \Gamma_t)).$$

Here $N(T)$ is the number of rows in the table T, and $F(n, \psi_1, \psi_2, \ldots)$ is an operator which transforms the considered tuple of nonnegative numbers into a nonnegative number. Note that the number of variables $\psi_1, \psi_2, \ldots$ is not bounded from above.

The considered complexity measure will be called *monotone* if for any natural i, t, $1 \leq i \leq t-1$, and any nonnegative numbers $a, c_1, \ldots, c_t, d_1, \ldots, d_t$ the inequality $F(a, c_1, \ldots, c_t) \geq \max\{c_1, \ldots, c_t\}$ holds, the equality $F(a, c_1, \ldots, c_i, c_{i+1}, \ldots, c_t) = F(a, c_1, \ldots, c_{i+1}, c_i, \ldots, c_t)$ holds, the inequality $F(a, c_1, \ldots, c_{t-1}) \leq F(a, c_1, \ldots, c_t)$ holds if $t \geq 2$, and from inequalities $c_1 \leq d_1, \ldots, c_t \leq d_t$ the inequality $F(a, c_1, \ldots, c_t) \leq F(a, d_1, \ldots, d_t)$ follows.

The considered complexity measure will be called *strongly monotone* if it is monotone and for any natural t and any nonnegative numbers $a, c_1, \ldots, c_t, d_1, \ldots, d_t$ from inequalities $a > 0$, $c_1 \leq d_1, \ldots, c_t \leq d_t$ and inequality $c_i < d_i$, which is true for some $i \in \{1, \ldots, t\}$, the inequality $F(a, c_1, \ldots, c_t) < F(a, d_1, \ldots, d_t)$ follows.

Now we take a closer view of some complexity measures.

Number of nodes: $\psi(T, \Gamma)$ is the number of nodes in α-decision tree Γ. For this complexity measure $\psi^0 = 1$ and $F(n, \psi_1, \psi_2, \ldots, \psi_t) = 1 + \sum_{i=1}^{t} \psi_i$. This measure is strongly monotone.

Depth: $\psi(T, \Gamma)$ is the maximal length of a path from the root to a terminal node of Γ. For this complexity measure $\psi^0 = 0$ and $F(n, \psi_1, \psi_2, \ldots, \psi_t) = 1 + \max\{\psi_1, \ldots, \psi_t\}$. This measure is monotone.

Total path length: for an arbitrary row $\bar{\delta}$ of the table T we denote by $l(\bar{\delta})$ the length of the path from the root to a terminal node of Γ which accepts $\bar{\delta}$. Then $\psi(T, \Gamma) = \sum_{\bar{\delta}} l(\bar{\delta})$, where we take the sum on all rows $\bar{\delta}$ of the table T. For this complexity measure $\psi^0 = 0$ and $F(n, \psi_1, \psi_2, \ldots, \psi_t) = n + \sum_{i=1}^{t} \psi_i$. This measure is strongly monotone.

Note that the average depth of Γ is equal to the total path length divided by $N(T)$.

Proposition 2. *Let T be an α-decision table and ψ a monotone complexity measure. Then there exists an irredundant α-decision tree for T that is optimal relatively to complexity measure ψ among all α-decision trees for T.*

4.3 Procedure of Optimization

Let G be a proper subgraph of the graph $\Delta_\alpha(T)$, and ψ be a complexity measure. Below we describe a procedure, which transforms the graph G into a proper subgraph G_ψ of G. We begin from terminal nodes and move to the node T. We attach a number to each node, and possible remove some bundles of edges, which start in the considered node. We attach the number ψ^0 to each terminal node. Consider a node Θ, which is not terminal, and a bundle of edges, which starts in this node. Let edges be labeled with pairs $(f_i, a_1), \ldots, (f_i, a_t)$, and edges enter to nodes $\Theta(f_i, a_1), \ldots, \Theta(f_i, a_t)$, to which numbers $\psi_1, \ldots, \psi_t$ are attached already. Then we attach to the considered bundle the number $F(N(\Theta), \psi_1, \ldots, \psi_t)$.

Among numbers attached to bundles starting in Θ we choose the minimal number p and attach it to the node Θ. We remove all bundles starting in Θ to which numbers are attached that are greater than p. When all nodes will be treated we obtain a graph. We remove from this graph all nodes such that there is no a directed path from the node T to the considered node. Denote this graph by G_ψ. As it was done previously for any node Θ of G_ψ we denote by $D_{\alpha,G_\psi}(\Theta)$ the set of α-decision trees associated with Θ.

Let T be a decision table and ψ a monotone complexity measure. Let G be a proper subgraph of $\Delta_\alpha(T)$ and Θ an arbitrary node in G. We will denote by $D^{opt}_{\alpha,\psi,G}(\Theta)$ the subset of $D_{\alpha,G}(\Theta)$ containing all α-decision trees having minimal complexity relatively to ψ, i.e. $D^{opt}_{\alpha,\psi,G} = \{\hat{\Gamma} \in D_{\alpha,G}(\Theta), \psi(\Theta, \hat{\Gamma}) = \min_{\Gamma \in D_{\alpha,G}(\Theta)} \psi(\Theta, \Gamma)\}$.

The following theorems describe important properties of the set $D_{\alpha,G_\psi}(\Theta)$ for the cases of monotone and strongly monotone complexity measure ψ.

Theorem 1. *Let T be a decision table and ψ a monotone complexity measure. Let G be a proper subgraph of $\Delta_\alpha(T)$ and Θ an arbitrary node in the graph G. Then $D_{\alpha,G_\psi}(\Theta) \subseteq D^{opt}_{\alpha,\psi,G}(\Theta)$.*

Theorem 2. *Let T be a decision table and ψ a strongly monotone complexity measure. Let G be a proper subgraph of $\Delta_\alpha(T)$ and Θ be an arbitrary node in the graph G. Then $D_{\alpha,G_\psi}(\Theta) = D^{opt}_{\alpha,\psi,G}(\Theta)$.*

5 Managing Algorithm Complexity

The main obstacle for designing efficient algorithm based on the graph optimization procedure is large computational complexity. In [4] a class of restricted information systems was considered. A restricted information system describes an infinite family of decision tables for which algorithm complexity is bounded from above by a polynomial on the table description length. In general case, the number of separable subtables grows exponentially, that makes the procedure of building and optimizing graph computationally intractable. However complexity can be managed by increasing parameter α. We did not have theoretical estimates for dependence of the number of subtables on α, but experiments show it drops rather quickly.

To illustrate the dependence we took two data sets from UCI Machine Learning Repository [1]. Training part of Poker Hand data table contains 25010 rows and 10 columns (attributes). SPECT data set (both training and test part) contains 216 rows after removing duplicating rows and 22 attributes. For both data sets we chose several values of α and built α-decision trees optimal relatively to depth, average depth and number of nodes. Tables 1 and 2 contain experimental results for Poker Hand and SPECT data sets respectively. Each table contains the following columns:

- *sf*: uncertainty scale factor (we assume α to be initial uncertainty of the decision table scaled to this paramter);
- *nodes*: number of nonterminal nodes in the graph $\Delta_\alpha(T)$;
- *time*: working time of algorithm that builds graph $\Delta_\alpha(T)$ in seconds;
- *optimal* and *greedy*: groups of parameters that describe characteristics of optimal trees and trees built by a greedy algorithm [6];
- *depth*: minimal depth of α-decision tree;
- *avg depth*: minimal total path length in decision tree divided by $N(T)$;
- *# nodes*: minimal number of nodes in α-decision tree.

The greedy algorithm is supposed to minimize depth of the tree so the results of the considered and greedy algorithm matches for the depth except for several cases. The difference in average depth (total path length) and in number of nodes is expectable. A good point about the considered algorithm is that it builds

Table 1. Exeperimental results for Poker Hand data set

sf	nodes	time	optimal			greedy		
			depth	avg depth	# nodes	depth	avg depth	# nodes
0	1426236	177	5	4.08	18831	5	4.15	22989
10^{-8}	1112633	124	5	3.99	15766	5	4.03	20071
10^{-7}	293952	27	4	3.73	6658	4	3.82	15966
10^{-6}	79279	7	3	3	2269	3	3	2381
10^{-5}	15395	2	3	3	733	3	3	2381
10^{-4}	4926	< 1	2	2	183	2	2	183
10^{-3}	246	< 1	2	2	57	2	2	183
10^{-2}	21	< 1	1	1	14	1	1	14
10^{-1}	1	< 1	1	1	5	1	1	14

Table 2. Exeperimental results for SPECT data set

sf	nodes	time	optimal			greedy		
			depth	avg depth	# nodes	depth	avg depth	# nodes
0	1089352	54	8	4.38	65	9	5.05	115
10^{-3}	1010598	39	8	4.07	39	8	4.58	29
10^{-2}	396159	13	5	3.16	19	5	3.36	21
10^{-1}	8330	< 1	2	1.81	7	2	2	7

a set of optimal trees and allows sequential optimization relative to different complexity measures.

6 Conclusions

The paper is devoted to consideration of an algorithm for building optimal approximate decision trees. Possibilities of tradeoff between approximation rate and complexity are illustrated by experiments with two data sets from UCI ML Repository. Further studies will be connected with extension of this tool to decision tables which contain continuous attributes.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
2. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth & Brooks (1984)
3. Chegis, I.A., Yablonskii, S.V.: Logical methods of electric circuit control. Trudy MIAN SSSR 51, 270–360 (1958) (in Russian)
4. Chikalov, I., Moshkov, M., Zelentsova, M.: On optimization of decision trees. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets IV. LNCS, vol. 3700, pp. 18–36. Springer, Heidelberg (2005)
5. 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)
6. Moshkov, M.: Greedy algorithm of decision tree construction for real data tables. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Świniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 161–168. Springer, Heidelberg (2004)
7. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
8. 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)

Layered Approximation Approach to Knowledge Elicitation in Machine Learning

Tuan Trung Nguyen

Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warsaw, Poland
nttrung@pjwstk.edu.pl

Abstract. Domain knowledge elicitation constitutes a crucial task in designing effective machine learning algorithm, and is often indispensable in problem domains that display a high degree of internal complexity such as knowledge discovery and data mining, the recognition of structured objects, human behavior prediction, or multi-agent cooperation. We show how to facilitate this difficult and sometimes tedious task with a hierarchical concept learning scheme, designed to cope with the inherent vagueness and complexity of knowledge therein used. We present how our approach, based on Rough Mereology and Approximate Reasoning frameworks, correlate to other well established approaches to machine learning.

1 Introduction

From a general point of view, a concept learning task is to distinguish exemplars from non-exemplars of a given abstract concept using their available feature information. The standard paradigm to this problem, usually known as supervised learning, assumes that a certain amount of knowledge is available for the task in the form of a training set which contains training samples $u_i = (a_i, d_i), i = 1, \ldots, n$ where a_i denotes the available samples' feature values, and $d_i \in \{1, -1\}$ marks whether u_i is an exemplar of the concept being learned or not. Within the rough set approach to artificial intelligence, concept learning is equivalent to the task of concept approximation using attribute values provided by an information system [11].

Under a more technical point of view, the principal task of concept learning is to reconstruct an investigated decision function f that associates input data u_i with their outputs d_i, by way of assuming a hypothesis about the function f in the form of another function f_h, selected from a known *hypothesis space* H. Ideally, f_h should return the same output values as f for the same inputs. In practice, this is rarely attainable, and we try to approach that agreement in output values as close as possible.

1.1 The Need to Divide and Conquer

Machine learning problems are good examples of inverse problems which, unfortunately, are generally *ill-posed*. That means the solution function f might

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 446–455, 2010.

not exist, might not be unique, and most importantly, might not be stable – an arbitrarily small deviations in data may cause large deviations in solutions. The problem further aggravates in domains where sample data display highly complex structural features, such as in optical character recognition, face recognition, or image analysis. Mapping from inputs to outputs when dealing with such complex objects is usually intractable, which necessitates the use of incremental, step-by-step approaches. Instead of an all-out effort to search for the target hypothesis, we attempt to attain consecutive, simpler, more manageable subgoals that would gradually lead to the desired results.

It is therefore not surprising that many studies in the theory of learning [20], [5] pointed out that certain aspects of machine learning algorithms can be tackled with using classic methods from functional optimization and inverse problem theory. For example, Tikhonov regularization, a standard technique for ill-posed problems, can be profitably applied to supervised learning. The minimized empirical error method, together with regularization, gave rise to the Structural Risk Minimization (SRM) technique [20]. Both techniques can be viewed as attempts to optimize the hypothesis search process by reducing the dimensionality of the search spaces and the description length.

A similar trend to the incremental approach to learning can also be observed in the Theory of Cognition, where knowledge chunking techniques are used in cognitive architecture such as SOAR to optimize search spaces and elicit search control knowledge of subgoals' learning procedures [6]. Chunking can be viewed as a mean to breakdown a complex learning task into more manageable and reusable subtasks.

One central to this divide-and-conquer approach however is to determine how to effectively and efficiently break down a given complex task into intermediate subtasks. We shall show that the layered approximation methodology can help in addressing this problem.

1.2 The Role of Domain Knowledge

A second important issue in machine learning the use of domain knowledge. It is widely acknowledged that learning algorithms would perform better when equipped with certain knowledge on the domain of interest. [2] Domain (or background) knowledge can serve as additional search control tools. Usually fast and efficient greedy searches have limits in the patterns they can discover, while complex and more elaborated, more exhaustive strategies typically display high computational costs. The trade-off between the two groups might be greatly refined with domain knowledge in order to steer the search process to more promising areas more quickly or to fine tune the construction of components patterns that would be difficult to find greedily.

Domain knowledge may help in setting certain *a priori* assumptions to the learning process. In this case it is often referred to as *learning bias* and can also greatly influence the design of learning algorithms. For instance, one might confidently assume that humans write characters or numerals as a sequence of strokes, and this prior knowledge might be decisive in building a learning model

that will try to extract strokes from a character's image and to classify the image using so identified strokes. [2]. Another prior information, that character identities are generally invariant in regard to sample's rotation or scaling, is vastly beneficial to a character recognition system, e.g. in selecting distance functions that are known to be invariant to such transformations. Once incorporated into a learning system, this kind of domain knowledge usually remains unchanged.

Another, more dynamic form of domain knowledge comes from external domain (usually human) experts that would interact with a learning system and provide it with their own evaluation of what the system is doing. It is worthy to note that while humans sometimes may not be able to explicitly explain how they perform certain tasks, they often find it easy to correct things that "went wrong" on specific examples. Incorporating this knowledge into the learning process is an effective way to improve its overall performance. This approach to learning is commonly referred to as learning from instruction [9].

Domain knowledge of this type usually needs to be "assimilated", i.e., converted to such a form that can be readily used by the learning system. This is mainly because the domain knowledge usually comes from human sources who tend to reason in quasi-natural languages, usually rich in ambiguities, vagueness and imprecision, while computing systems need to perform operations using more precise and strictly defined languages. Moreover, the expert most typically employs an ontology of concepts and relations that are completely foreign to the learner system. The expert knowledge therefore needs to be made suitable for execution in the target system. This process is sometimes referred to as operationalization, or knowledge refinement [9]. We will demonstrate that this task can be dealt with effectively using the layered approximation methodology.

1.3 Vagueness of Concepts Being Learned

It is very often that a concept is not readily expressible in an universe of samples U and can only be approximated by description languages available to the learner. Due to the limited expression power of these languages, we have to accept the vagueness and imprecision of the induced descriptions. Vagueness and uncertainty also occur in the knowledge elicitation process from external human experts. The expert's ontology and declarative languages are often based on natural language constructs, where ambiguity and imprecision are abundant.

This issue can be addressed by a number of techniques in artificial intelligence such as multi-valued logics, fuzzy sets, and most notably, by using rough set theory, which provides an excellent theoretical framework, as well as effective tools to deal with vagueness and uncertainty in reasoning about data.

In our hierarchical learning approach, we assume that the decomposition scheme will be provided by an external human expert in an interactive process. Knowledge acquired from human expert will serve as guidance to break the original learning model A into simpler, more manageable sub-models A_i, organized in a lattice-like hierarchy. They would correspond to subsequent levels of abstractions in the hierarchy of perception and reasoning of the human expert.

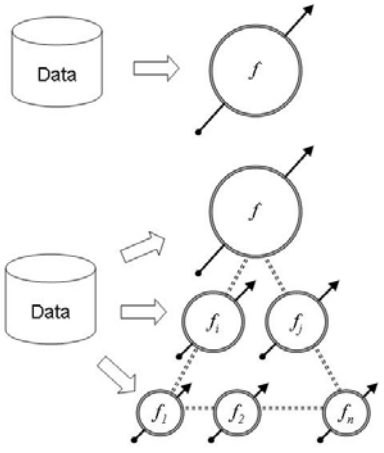


Fig. 1. All-out vs. hierarchical learning

2 Knowledge Elicitation

The knowledge on training samples that comes from a 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.

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.

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 [15], and are designed to deal with the notion of inclusion to a degree.

An example of such concept inclusion measures would be:

$$Match(p, C) = \frac{|\{u \in T : Found(p, u) \wedge Fit(C, u)\}|}{|\{u \in T : Fit(C, u)\}|}$$

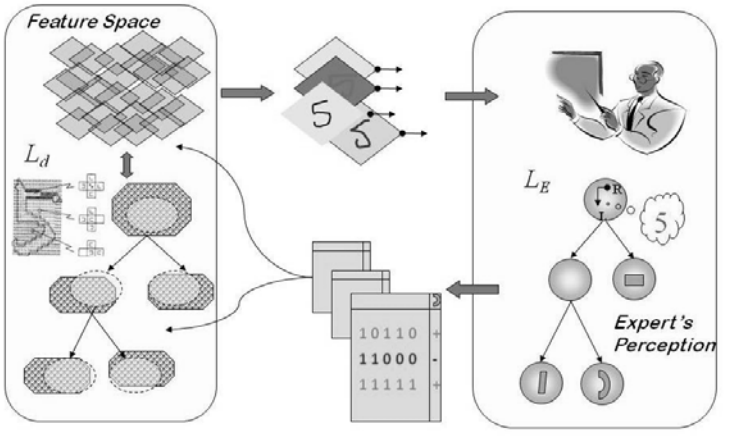


Fig. 2. Layered approximation in knowledge elicitation

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.

We essentially seek to convert 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 can be obtained by fine-tuning various pertaining coefficients. Adjustment of these coefficients based on feedback from actual data may help optimize the approximation quality.

For an example, let us 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 '$Weak$'.

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 (see Table 2). Domestic patterns satisfying the defined quality requirement can be quickly found, taking into account

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*

that sample tables submitted to experts are usually not very large. Since this is essentially a rather simple supervised learning task that involves feature selection, many strategies can be employed. In [14], genetic algorithms equipped with greedy heuristics are reported successful for a similar problem. Neural networks also prove suitable for effective implementation.

Similarly, we can approximate the expert's perception on relations between parts of a sample (see Table 3). The corresponding low-level features may be expressed by, for instance, $S_y < B_y$, which tells whether the median center of the stroke is placed closer to the upper edge of the image than the median center of the belly. (see Table 4)

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)

The expert's perception *"A '6' is something that has a 'vertical stroke' 'above' a 'belly open to the west'"* is eventually approximated by a classifier in the form of a rule:

if S(#BL_SL > 23) **AND** B(#NESW $> 12\%$) **AND** $S_y < B_y$ **then** CL='6',

where S and B are designations of pixel collections, #BL_SL and #NESW are numbers of pixels with appropriate topological features, and $S_y < B_y$ concerns the centers of gravity of the two collections.

For more detailed descriptions on successful implementations of the layered concept approximation scheme, see [11] or [13]

3 Layered Approximation and Other Learning Techniques

In this section, we discuss the common aspects and motivations of the Layered Approximation method and other better known techniques in supervised learning.

3.1 Statistical Learning and Structural Risk Minimization

As mentioned in [1], the general outset of a supervised learning task very often results in an ill-posed problem. In particular, as stated in [20], the problem of estimating the desired class density function f from a large set $\mathcal{F}$ of possible candidate solutions for an supervised learning task is ill-posed. One way to alleviate this problem is to employ the so-called Structural Risk Minimization (SRM) technique. The technique, in short, is based on a theorem on the risk bounds, which essentially states that

$$R(\alpha) \leq R_{emp}(\alpha) + CI(\alpha)$$

which means the risk functional $R(\alpha)$, expressing how far we are from the desired solution for a parameter α from a general parameter set S, is bounded by the sum of the empirical risk $R_{emp}(\alpha)$ and a confidence interval $CI(\alpha)$ containing the Vapnik-Chervonenkiss dimension of the function space S.

Instead of optimizing α over an arbitrary set of possible parameters S, we use the bounds to find a set S^* for which the risk bound is minimal, and then perform the search for the solution α^* within S^*. For more details, see [20].

The SRM technique essentially seeks to optimize the complexity of the hypothesis space using statistical results and methods that can be traced back to regularization theory and the minimal description length principle.

The hierarchical learning approach, which reduces the complexity of the original learning problem by decomposing it into simpler ones, tries to optimize the corresponding search spaces on subsequent levels of the learning hierarchy, and is similar in function to the SRM technique, with the distinction that the breakdown scheme is provided by an external human expert, and the learning process is incremental instead of a straightforward regularization-based optimization.

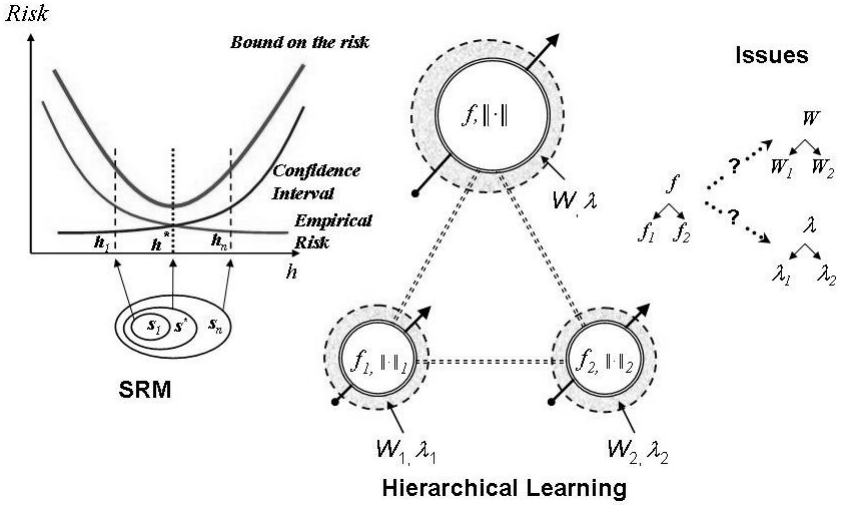


Fig. 3. SRM vs. Layered Approximation

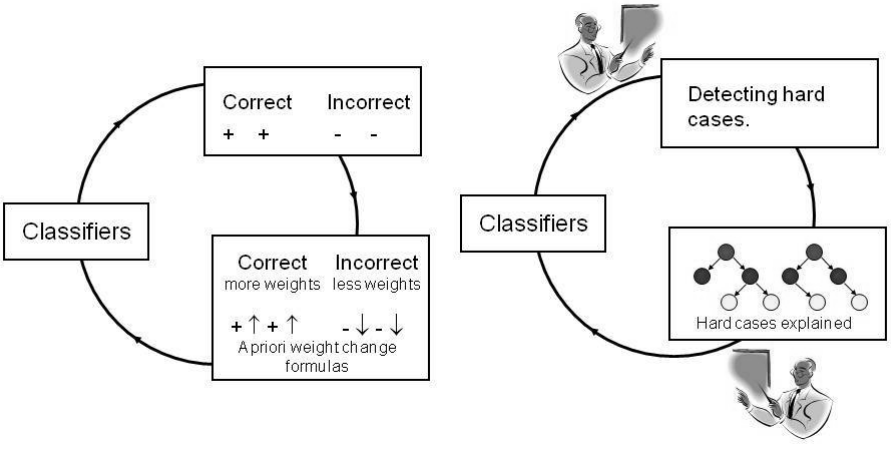


Fig. 4. Boosting vs. Layered Approximation

3.2 Boosting Algorithms

Boosting [4] is a meta learning algorithm that aims to improve the performance of a classifier by its iterative training on the same training set with adjusted weights on the successfully learned samples and the confusing ones. The main idea is to steer the classifier toward the difficult, hard-to-learn samples at the expense of the successfully learned ones. The boosting method is theoretically proved to yield strong classifier even for weak starting classifier [4]. Boosting also makes excellent platform to incorporate prior and domain knowledge [16].

The boosting learning scheme parallels the layered approximation method most notably in the incremental, iterative scheme that steer the attention of the classifier toward difficult, hard-to-learn cases. The difference is layered approximation relies on an external expert in hard case detection and treatment, whereas boosting employs fixed weight adjustment algorithms.

3.3 Learning from Instruction

The layered approximation process, in many aspects, takes after the Learning from Instruction paradigm, where a learning system acquires new information and skills from an external instructor [9]. They share many common issues, such as the knowledge operationalization or the search space optimization problems. However, most Learning from Instruction existing methods employ declarative implementation tools such as logic programming, whereas layered approximation prefers direct procedural supervised learning at subsequent approximation levels.

4 Conclusion

We describe the fundamentals of the layered approximation approach to the task of eliciting domain knowledge from a human expert in machine learning. The designated objective was attained through a hierarchical, step-by-step concept assimilation and approximation process. We show that this approach, based on rough mereology and approximate reasoning, is closely related and complementary to flagship methods in the theory of learning. A reference to successful implementation of the described methods is also provided.

References

1. Domingos, P.: Toward knowledge-rich data mining. Data Minning and Knowledge Discovery 15(1), 21–28 (2007)
2. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley-Interscience Publication, Hoboken (2000)
3. Fensel, D.: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer, New York (2003)
4. Freund, Y., Schapire, R.E.: A short introduction to boosting. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, pp. 1401–1406. Morgan Kaufmann, San Francisco (1999)
5. Kurkova, V.: Learning from data as an inverse problem. In: 16th Symposium on Computational Statistic (COMPSTAT 2004), pp. 1377–1384. Physica-Verlag, Heidelberg (2004)
6. Laird, J.E., Rosenbloom, P.S., Newell, A.: Chunking in soar: The anatomy of a general learning mechanism. Mach. Learn. 1(1), 11–46
7. Langley, P., Laird, J.E.: Cognitive architectures: Research issues and challenges, Technical Report (2002)
8. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
9. Mostow, D.J.: Machine transformation of advice into a heuristic search procedure. In: Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (eds.) Machine Learning: An Artificial Intelligence Approach, pp. 367–403. Springer, Heidelberg (1984)
10. Newell, A.: Unified theories of cognition. Harvard University Press, Cambridge (1994)
11. Nguyen, S.H., Bazan, J., Skowron, A., Nguyen, H.S.: Layered learning for concept synthesis. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Świniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 187–208. Springer, Heidelberg (2004)
12. Nguyen, T.T.: Domain knowledge assimilation by learning complex concepts. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 617–626. Springer, Heidelberg (2006)
13. Nguyen, T.T.: Domain knowledge assimilation by learning complex concepts. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 617–626. Springer, Heidelberg (2006)
14. 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 (ICPR 2002), pp. 568–571 (2002)

15. Polkowski, L., Skowron, A.: Rough mereology: A new paradigm for approximate reasoning. Journal of Approximate Reasoning 15(4), 333–365 (1996)
16. Schapire, R.E., Rochery, M., Rahim, M.G., Gupta, N.: Incorporating prior knowledge into boosting. In: ICML 2002: Proceedings of the Nineteenth International Conference on Machine Learning, pp. 538–545. Morgan Kaufmann Publishers Inc., San Francisco (2002)
17. Shavlik, J.W., Towell, G.G.: An approach to combining explanation-based and neural learning algorithms. Connection Science 1, 231–254 (1989)
18. Skowron, A.: Rough sets in perception-based computing. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 21–29. Springer, Heidelberg (2005)
19. Skowron, A., Polkowski, L.: Rough mereological foundations for design, analysis, synthesis, and control in distributed systems. Information Sciences 104(1-2), 129–156 (1998)
20. Vapnik, V.N.: The Nature of Statistical Learning Theory (Information Science and Statistics). Springer, Heidelberg (1999)
21. Zadeh, L.A.: From imprecise to granular probabilities. Fuzzy Sets and Systems 154(3), 370–374 (2005)

Configuration Management of Mobile Agents Based on SNMP

Michał Komorowski

Institute of Computer Science,
Warsaw University of Technology,
Nowowiejska 15/19, Warsaw 00-665, Poland
M.Komorowski@ii.pw.edu.pl

Abstract. Mobile agents are small programs that can transport their code, data and execution context from one machine to another and be capable of continuing execution in the new environment. This technology has a lot of advantages and promising applications, unfortunately there is a noticeable absence of deployed solutions. There are few reasons of this situation but one of the most important is the lack of tools that can be used for the configuration management of mobile agents. This process focuses on the monitoring and controlling configuration items and is essential for other processes like incident management or availability management. In this paper a new, flexible and universal solution for the configuration management of mobile agents is proposed. This solution is based on well known and widely used management standard - *SNMP* (Simple Network Management Protocol).

Keywords: Configuration management, Mobile agent, Mobile agent system, Multiagent systems, *SNMP*.

1 Introduction

The Internet is the most popular and widespread medium in the whole history of the humanity. Year by year many new solutions that take advantage of it are proposed, among them mobile agents (*MA*). Mobile agents are small programs that can migrate from one environment to another. They are autonomous, can communicate and cooperate with each other and learn. This technology has a lot of advantages and many examples of promising applications [6], [12], [18] can be found in the literature. Unfortunately there are not many examples of deployed systems that are based on mobile agents. There are a few reasons of this situation but one of the most important is the lack of tools that can support the configuration management of mobile agents. Although there are many publications [2], [3], [8], [9] on managing information infrastructure with *MA*, the problem how to manage mobile agents is neglected.

The configuration management is a very wide term. According to [5] it is identification, monitoring and reporting the state of chosen elements during the life of a system. Configuration management provides the basic information about

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 456–465, 2010.

how the system works. Without this information other management processes cannot function properly. For example the availability management needs the detailed configuration of the hardware/software used in the company in order to predict how the failure of the particular element affects others. One of the most popular technology supporting this process is *SNMP* (Simple Network Management Protocol).

In this paper a new, flexible and universal solution for the configuration management of mobile agents is proposed. In comparison to other solutions it takes advantages of well known management standard - *SNMP* and can be used with different mobile agent systems. This solution is designed in such a way that, if necessary, a support for other management technologies can be easily implemented. In order to evaluate the described solution the partial implementation of it was developed and put under the strict functional and efficiency tests.

At the beginning of this paper, in the section 2, mobile agents are described. Then the process of the configuration management of mobile agents is characterised in the section 3. The *SNMP* is briefly discussed in the section 4. In the section 5 existing solutions are described. The description of the proposed solution can be found in the section 6. Finally the implementation is described in section 7 and the experiments in the section 8. The last section 9 contains the summary.

2 Mobile Agents

Mobile agents are special kind of software agents. According to [19] a software agent is a component "which is capable of acting exactingly in order to accomplish tasks on behalf of its user" and it should have at least two features from the set of three: being autonomous, being able to communicate and cooperate with other agents and learn. Mobile agents have all these features. They can work without control of a human (autonomy), send messages to each other (ability to communicate) and change behaviour based on the observation of the environment (ability to learn). Except this set of features mobile agents have one more - mobility.

Mobility should be understood as ability to stop execution on the machine A and move code, data and execution context to the machine B and there continue processing. What is important the target machine does not have to know anything about a migrating agent, especially about the code of the agent. The software that allows mobile agents to migrate is known as *MAS* (Mobile agent system). The host is a computer that has this kind of software installed. The running sample of *MAS* software is in turn known as instance of *MAS*. The term mobile agent system has one more meaning. It can be also understood as a set of hosts with running instances of *MAS* software.

At present the majority of *MAS* is written in *Java*. As a result the migration of execution context (content of registers, stack, etc.) is not possible because programmers do not have access to this information in the *Java Virtual Machine*. Some examples of mobile agent systems are: *SeMoA* [20], *Jade* [10] or *Aglets* [1].

Thanks to the mobility *MA* can perform calculation near the data storage. Local calculations are usually faster because there is no need to transfer data through the network. Moreover mobile agents have access to the newest data and thanks to it better and more suitable decisions can be taken. The local calculations have also this advantage that data are more secure because they do not leave local environment. It is true that *MA* can be intercepted during migration but usually they do not carry a lot of data so it is not so dangerous. The another advantage of local calculations is lower network load because data are not sent through the network. Mobile agents are generally small programs so they should not affect the bandwidth.

MA are also very flexible. It is always possible to create a new type of mobile agent and to use it in already working system. The actualisation process of existing types of agents is also easy. Simplifying, everything that should be done is the replacement of agents' code in the repository. It is also worth mentioning that *MA* can improve system immunity to networks failures. If the *MA* is well designed, it will be able to wait on some host until network communication is restored.

As it was mentioned majority of *MAS* is created in *Java*. Thanks to it and to the popularity of this technology *MA* can work in heterogeneous environments that can consist of personal computers, PDA's, machines with Windows or Linux operating system installed.

3 Configuration Management of Mobile Agents

The process of the configuration management of mobile agents is similar to management processes in the other technologies. Every process of the configuration management focuses on configuration elements (*CE*). It can be a hardware element like router, switch, personal computer, etc. or software like some kind of application or database engine. In the context of *MAS* every instance of *MAS* or every *MA* is a configuration element. Configuration of a *CE* is simply a description of its state and properties (configuration parameters). The description of a mobile agent can contain information about its life cycle, its location or parameters specific for the particular agent type. The life cycle describes all possible states of mobile agents (active, waiting, migrating, etc.) and transitions between these states. The process of the configuration management of mobile agents consists of three sub-processes:

– **Configuration management of individual mobile agents**. This subprocess focuses on the individual mobile agents and their configuration. For example, a mobile agent designed for performing some kind of distributed calculations can use parameter defining the set of hosts, on which the calculations can be performed. When the administrator notices that it takes too much time to finish the calculations, he or she can extent the set of the machines available for the calculations.

- **Configuration management of hosts**. This sub-process focuses on the environment in which mobile agents are working. Some examples of parameters important for this activity are: amount of resources available to agents, number of agents that can be working in the system at the same time or types of agents that are allowed to visit individual hosts. For instance, consider the situation in which some hosts in the network have efficiency problems. The administrator of the system can limit amount of resources available to agents on these hosts to check if the problem is caused by them. At the same time *MA* can continue their tasks.
- **Information collection**. This sub-process is responsible for collecting information about mobile agents activity: number of working mobile agents, amount of used resources or history of hosts visited by the particular *MA*. This sub-process is essential for the diagnosis of the state of the system. For example information collection is necessary to check if some *MA* uses too much of the resources and should me preempted.

4 SNMP

SNMP (Simple Network Management Protocol) is a standard that describes how to manage software and hardware connected to some network. Although *SNMP* is an old technology (first version of *SNMP* was proposed in 1988), it is still widely used. For example it is possible to manage Windows operating system, Oracle databases or CISCO products with it.

Network management system based on *SNMP* consists of a few kinds of elements. Management station is a computer with proper software installed that is responsible for monitoring and changing the configuration of routers, databases, instances of *MAS* or individual mobile agents. To achieve this goal a management station communicates with *SNMP* agents in protocol defined by the standard. *SNMP* agents are programs that understand this protocol and can communicate with configuration elements. In the context of mobile agents *SNMP* agent can be the part of the mobile agent system provided as one of the services.

SNMP also defines *SMI* (Structure of Management Information) notation which is used to describe the configuration of *CE*. Once the description is created it can be shared by many manufacturers. This description, known as the *MIB* (Management Information Base), is only the declaration of configuration parameters (*MIB* objects) and should not be mistaken with the real values of configuration parameters (with instances of *MIB* objects). The instances of *MIB* objects can be stored in many ways for example in the relational database.

The more detailed description of *SNMP* is beyond the scope of this paper and can be found in the publication [17].

5 Related Works

In this section previous works on the topic of the configuration management of mobile agents are described. Some tools integrated into mobile agent systems are also mentioned.

The *Jade* [10] mobile agent system allows monitoring of the activity of mobile agents, messages sent and received by them and many other parameters. Unfortunately these tools are specific for *Jade* and cannot be used with other systems. The *Aglets* [1] provides user only with very basic tools as possibility to stop or remove *MA* from the system. *Voyager Edge* [21] does not have any tools of this kind. The *SeMoA* [20] has the application for monitoring geographic location of agents but it is more a gadget that a useful tool.

The authors of [14] propose a new life cycle model of mobile agents and used it in a monitoring system. The solution seems to be interesting however it focuses on only one aspect of the configuration management of mobile agents - information collection. In [4] authors propose solution that allows finding instance of *MAS* for particular *MA* based on available resources. Unfortunately at the same time they introduced completely new way of describing configuration of *MAS* instead of choosing *SNMP* or other standard. The authors of [13] proposed a new mobile agent system *JAMES* than can be managed with *SNMP* but they did not prepare their solution for integration with other systems. Similar idea can be found in [15].

To summarise existing solutions are not satisfactory. They are specific for particular mobile agent systems, they do not use universal technologies and cannot be integrated with many *MAS*.

6 Mobile Agents Configuration Management System

6.1 Requirements

In this subsection based on the prior analysis of the existing solutions, requirements for a new tool are formulated. Later the term *MACMS* (Mobile Agents Configuration Management System) will be used.

Systems based on *MA* can change very dynamically and it is always possible to create a new type of *MA* that will be described by a different set of configuration options. In consequence *MACMS* must be flexible and allow administrators to add, modify or remove configuration parameters easily.

MACMS should be also universal enough to be used in many different types of *MAS*. Of course some modification of existing mobile agent systems can be necessary in order to be integrated with *MACMS*. If this requirement is fulfilled, it will be possible to manage many kinds of mobile agents with one tool at the same time.

Finally, *MACMS* should be based on widely known and accepted technology e.g. like *SNMP*, so that the solution would be easier to integrate with existing configuration management systems. However it is possible that in future chosen technology will be replaced by another so the *MACMS* must be easily modified and extended.

6.2 Architecture

The central part of the system is called the core of *MACMS*. It has a few functions. Firstly, it is responsible for communication with *MAS* and mobile agents.

The communication can be implemented in many different ways depending on the situation. If the code of *MAS* is available and can be modified the core can be integrated into *MAS* as one of the services. Otherwise the core can work as the separate process but in this case a *IPC*(Inter process communication) mechanism must be used. Both approaches have advantages and disadvantages. The first one seems to be easier but on the another hand the crash in the core of *MACMS* can affect the mobile agent system.

The core is also responsible for storing and retrieving values of configuration parameters in/from the data store. The access to the data store (a relational data base, a XML file, a flat file) is synchronised in order to preserve the consistency of the data.

The next important function is managing the definitions of configuration parameters. These definitions are read every time the system is initialised. The definitions can be created, removed or changed on the request from the administrative application. This application provides administrator with easy to use graphical user interface. Thanks to it administrator can easily and quickly customise definitions of configuration parameters used by many instances of *MAS*.

Finally the core of *MACMS* provides services known as access points or interfaces that are used to monitor the configuration parameters and change them accordingly. The *SNMP* interface use the *SNMP* protocol to communicate with the external environment.

Another example of the access point is the one that uses *HTTP* protocol. This interface allows only for reading values of configuration parameters but everything that is necessary to make the query is a web browser. Depending on future requirements many other access points can be easily implemented.

Thanks to the proper design every part of the *MACMS* can be replaced by other implementation without much effort. Firstly, every function of the system is accessible through well an defined interface. Secondly, concrete implementation of these interfaces are produced in one central place known as a factory. For example in order to replace the storage for values of configuration parameters new implementation of *IDataStoreProvider* interface should be provided and the factory *DataStoreProvidersFactory* should be adapted accordingly. It is always possible to switch between different implementations of the same interface by modifying configuration file.

7 Implementation

In order to evaluate the proposed solution a partial implementation of it was developed. In comparison to the full implementation it is not possible to manage configuration of the hosts, monitoring information are not collected and support for *SNMP* is limited (the table data structure and some protocol commands are not supported). The implementation uses *Aglets* [1] mobile agent system. This *MAS* was created in *IBM* laboratories but at present it is an open source project. Thanks to it, it was possible to integrate the core of *MACMS* into system physically as one of the *MAS* service. Prior to the integration the code of *Aglets*

was reviewed and warnings or uses of the deprecated *API* were removed. The compilation of the system was also migrated from *Java* 1.4 to *Java* 6.

In order to provide backward compatibility for existing mobile agents, during the integration the old *Aglets API* remained unchanged. Especially the old base class for all mobile agents (*Aglet*) was not modified but the new base class (*CMSAglet*) was prepared for those who want to use functionality of *MACMS*.

The *JAgentX* [11] technology was used in order to provide *SNMP* support. It is n *Java* implementation of the *AgentX* (Agent Extensibility) protocol that is described in RFC 2741. *AgentX* is an extension of *SNMP* that allows to dynamically change (add/remove/modify) sets of parameters supported by *SNMP* agents. It was necessary to use one of *SNMP* extensions because in the basic version *SNMP* is generally a static solution - the range of parameters (*MIB* objects) handled by *SNMP* agent is fixed. *JAgentX* was chosen because it is the newest product.

As a storage for values of configuration parameters the relational data base was chosen. The implementation uses the *HSQLDB* data base [7].

8 Experiments

The implementation was validated against a set of test cases. Every test case consists of dozens of steps and descriptions of the expected results. Some examples of test scenarios are:

1. Reading the value of a configuration parameter with the *SNMP* interface.
2. Modification of a configuration parameter by mobile agents.
3. Creation of a new configuration parameter by the administrator of the system.

The functional tests were conducted in the environment consisted of three hosts: two personal computers and one laptop. The personal computers were communicating via the cable while the laptop were using the wireless connection. Mobile agents used in the tests were migrating between hosts, modifying configuration parameters like: number of visits on the host or number of creations and reading parameters to know how to behave, for instance the text of message for the user. At the same time the management station was communicating with the *SNMP* interface to monitor the state of system and to change the behaviour of agents. As the management station, the professional network monitoring software [16] was used. The *HTTP* interface was also tested.

After functional tests had been successfully finished the efficiency tests of the solution were conducted on the machine with AMD Athlon 64 3500+ processor and 1 GB of RAM memory. Firstly, it was checked how much time it takes to read or write a configuration parameter depending on the number of mobile agents requesting the access to the parameter at the same time. It was difficult to synchronise the activity of hundreds of mobile agents so the following approach was used. The single test consisted of many attempts of every individual agent to read or write the parameter. On the beginning of every attempt the result of the

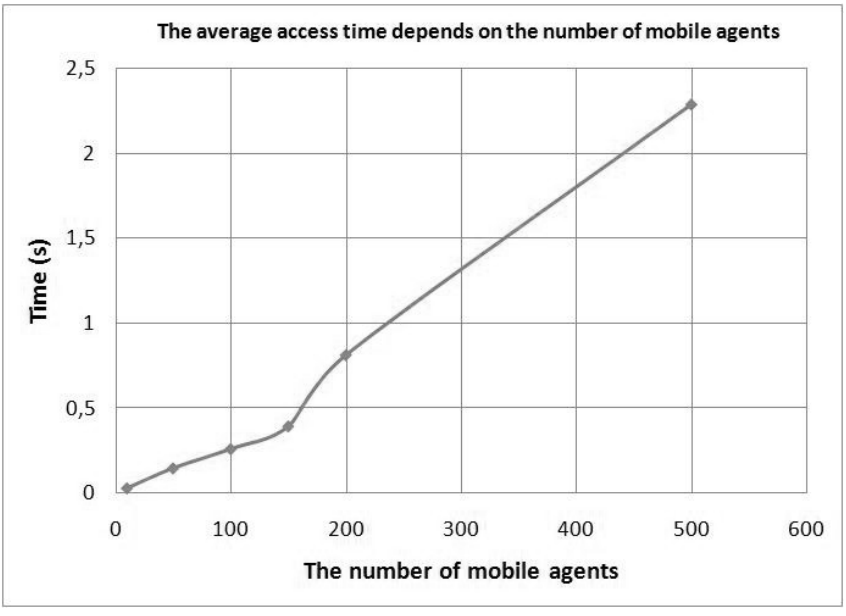


Fig. 1. The average access time depends on the number of mobile agents

System.currentTimeMillis method was saved. Then on the end of the operation the difference between current and saved time was calculated. These results were finally used to calculate the average access time. What is worth mentioning the configuration of every test was also read from the *MACMS*.

The result of the test are shown on the Figure 1. The access time can exceed even a few seconds. It should be also noticed that if a mobile agent needs to read or write a few parameters one by one the total time of the operation can reach dozens of seconds or even more. It is unacceptable in majority of applications. However, the author estimates that the number of *MA* working at the same time on any host should not be greater than dozen or so. In these conditions the access time is minor and can be neglected.

In the second test it was measured how the number of configuration parameters handled by the *MACMS* affects the start time of the instance of the *MAS*. This test was also based on the use of the *System.currentTimeMillis* method. The results are shown on the Figure 2. As it can be observed, the average start time increased considerably with the number of configuration parameters. It is not good, however, usually the instance of the *MAS* is not restarted very often, because the administrative tasks like the deployment of a new type of *MA* can be performed without system restart. After more detailed analysis of the logs it was also discovered that majority of the start time falls to the initialisation of the *SNMP* access point. It is a tip which component of the system should be optimised in the future.

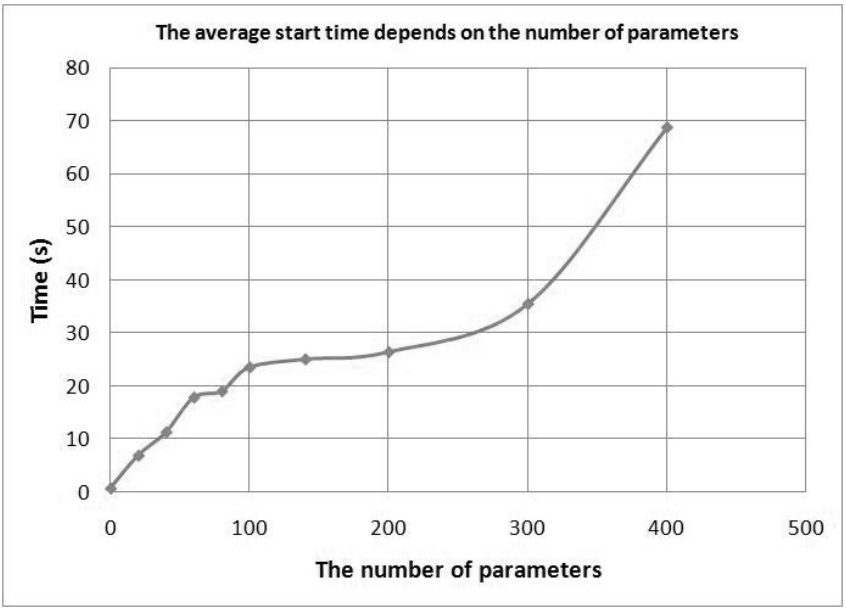


Fig. 2. The average start time depends on the number of parameters

To summarise the developed solution can affect negatively the efficiency of the mobile agent system but under normal conditions insignificantly and the work of mobile agents should not be disturbed. The normal conditions can be characterised as the situation in which there is no need to restart instances of the *MAS*, mobile agents are handling requests properly as they come and there are no network failures.

9 Summary

In this paper the problem of the configuration management of mobile agents was analysed and the new tool supporting this process was described. The proposed solution can be integrated with many different *MAS* and it is based on the *SNMP* but other technologies can be also used. The partial implementation of the propounded tool was developed and tested.

Nonetheless the problem of the management of the *MA* is not closed. Firstly, the security mechanism for controlling which agents can read/modify particular configuration parameters should be proposed. Secondly, the methodology describing how to manage the configuration of the *MA* is necessary. Among others the basic set of configuration parameters, the definition of *MIB* for mobile agents should be proposed. These parameters should be universal enough to be shared by many types of mobile agents and many different mobile agent systems.

Last but not the least, the data mining of the results of the monitoring seems to be very important matter. The analysis of the collected data can be helpful in detection of failures of mobile agents or efficiency problems.

References

1. Aglets, http://aglets.sourceforge.net, http://www.trl.ibm.com/aglets/
2. Lazar, S., Kodeswaran, S., Varadharaj, R., Sidhuu, D.: ATM network connection management using mobile agents. In: 25th Annual IEEE Conf. on Local Computer Networks, pp. 620–621. IEEE Computer Society, Los Alamitos (2000)
3. Kona, M., Xu, C.-Z.: A Framework for Network Management Using Mobile Agents. In: 16th Int'l. Parallel and Distributed Processing Symp., pp. 227–234. IEEE Computer Society, Los Alamitos (2002)
4. Giampapa, J.A., Juarez-Espinosa, O.H., Sycara, K.P.: Configuration management for multi-agent systems. In: 5th Int'l. Conf. on Autonomous Agents, pp. 230–231. ACM Press, Montreal (2001)
5. Hass, A.M.J.: Configuration Management Principles and Practice. Addison-Wesley, Reading (2003)
6. Zapf, M., Herrmann, K., Geihs, K., Wolfang, J.: Decentralized SNMP Management with Mobile Agents. In: 6th IFIP/IEEE IM Conf. on Network Management, Boston, pp. 623–635 (1999)
7. HSQLDB, http://hsqldb.org/
8. Pagurek, B., Wang, Y., White, T.: Integration of mobile agents with SNMP: Why and how. In: IEEE/IFIP Network Operations and Management Symp., Honolulu, pp. 609–622 (2000)
9. Kelash, H.M., Faheem, H.M., Amoon, M.: It takes a multiagent system to manage distributed systems. IEEE Potentials 26(2), 39–45 (2007)
10. Jade, http://www.jade.tilab.com
11. J. AgentX, http://eden.dei.uc.pt/agentx/
12. Cardoso, A.R., Celestino Junior, J., Celestino, R.A.R.: Management of Heterogeneous ATM Networks Based on Integration of Mobile Agents with Legacy Systems. In: Network Operations and Management Symp., vol. 1, pp. 879–882. IEEE/IFIP (2002)
13. Lus, P.S.: Mobile Agent Infrastructures: A Solution for Management or a problem to Manage? (2001), http://citeseer.ist.psu.edu/459261.html
14. Gong-ping, Y., Guang-zhou, Z.: Mobile Agent Life State Management. In: IMACS Multiconference on Computational Engineering in Systems Applications, vol. 1, pp. 448–451 (2006)
15. Rivalta, P.C.: Mobile Agent Management, Carleton University (2000)
16. OpManager, http://www.manageengine.com/products/opmanager
17. Stallings, W.: SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. Addison-Wesley Professional, Reading (1999)
18. Lange, D.B., Oshima, M.: Seven Good Reasons for Mobile Agents. Communications of The ACM 42(3), 88–89 (1999)
19. Nwana, H.S., Heath, M.: Software Agents: An Overview (1996), http://citeseer.ist.psu.edu/nwana96software.html
20. SeMoA, http://semoa.sourceforge.net
21. Voyager Edge, http://www.recursionsw.com/Products/voyager.html/

Adaptive Immunity-Based Multiagent Systems (AIBMAS) Inspired by the Idiotypic Network

Chung-Ming Ou[1] and C.R. Ou[2]

[1] Department of Information Management, Kainan University, Luchu 338, Taiwan
cou077@mail.knu.edu.tw
[2] Department of Electrical Engineering, Hsiuping Institute of Technology, Taichung 412, Taiwan
crou@mail.hit.edu.tw

Abstract. An adaptive immune-inspired multiagent system (AIBMAS) is proposed. The intelligence behind such system is based on the idiotypic immune network. Tunable activation Threshold (TAT) proposes that agents adapt their activation thresholds. Immune algorithm based on the immune network theory and memory mechanism is derived.

1 Introduction

The biological immune system inspires the immunity-based multiagent system (IBMAS) which can be an information framework of intra-agent processing and interagent information flows. From engineering viewpoints, concepts of immune systems are more important in its applications rather than biological explanations. This paradigm, namely artificial immune system (AIS), inspires researches of multiagent system as well.

Multiagent systems (MAS) have some features in common with AIS and provide scope for applying immune system methodologies. The main goal of the human immune system is to protect the internal components of the human body by eliminating the foreign elements such as the fungi, virus and bacteria. These processes include recognition, learning, communication, adaption, memory and control. MAS are based on behavior management of several independent agents. AIS may be applied to MAS to attain the computational intelligence of agents. It is suggested that some action generator be applied to MAS. According to Ishida [1], genetic coding for an agent could be used in a similar manner to that of genetic algorithm (GA), or other method in evolutionary computation (EC). However, a major challenge for AIS is to explain how each system adjusts responses to the environment when some antigens are recognized.

The AIS has been developed according to negative selection algorithm and clonal selection algorithm which are based on the classical self-nonself (SNS) theory; nonselfs are entities which are not part of human organisms [2]. Immune algorithm may basically apply to any system where the environment is unpredictable.

An adaptive system can be realized in immunity-based system by providing agents with further autonomy of reproduction with mutation. The term "agent"

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 466–475, 2010.

roughly corresponds to the immune cells such as B-cells and T-cells with autonomy and cooperation. The agent equipped with some special sensor and actuator will carry out actions corresponding to some special signals. We focus on taking theoretical immune property such as the tunable activation threshold (TAT) model [3] to the agent's considerations. TAT asserts that the responsiveness of individual lymphocyte to antigens and other signals can be tuned and updated. Controllability of MAS can be reached via tuning parameters of agent-based TAT model. This theory proposes that lymphocytes adapt their activation thresholds based on recent interactions with their environment [3] [4] [5].

Jerne [6] proposed the immune network theory by investigation immune systems as complex adaptive systems. Many researches have been proceeded according to this theory [7]. In this approach, immune network must first self-organize itself so that it will not respond to the self. This is the concept of self-maintenance. On the other hand, the immune system has indeed a short-term memory in the sense that it can respond more efficiently and rapidly to an antigen in the second invasion.

The purpose of this paper is to develop an adaptive immunity-based multiagent system (AIBMAS). The immune algorithm is inspired by AISIMAN [8] with enhanced memory mechanism and tunable activation threshold. The arrangement of this paper is as follows. In section 2, knowledge related to the AIBMAS is introduced. In section 3, AIBMAS architecture and corresponding immune algorithm are discussed; some control strategies for AIBMAS dynamics are proposed. In section 4, AIBMAS model evaluations are given.

2 Background

2.1 Immune Systems and Immune-Based Systems

The immune system consists of the antibodies and lymphocytes, which include T-cells and B-cells. The human immune system uses a large number of highly specific B- and T-cells to recognize antigens. Only B-cells secrete antibodies. Clonal selection theory explains the details of antibody secretion specific to an antigen where T-cells help regulating. The binding between antigen and specific lymphocytes trigger proliferation from immature lymphocytes to mature one and the secretion of antibodies. The immune system must interact not only with the nonself from the outer world, but also the self from the internal world.

An immunity-based system (IMBS) involves a self-maintenance system. According to [1], IMBS has the following three properties:

1. a self-maintenance system with monitoring both the nonself and the self
2. a distributed system with autonomous components capable of mutual evaluation
3. an adaptive system with diversity and selection.

2.2 Immunity-Based Multiagent Systems (IBMAS) with Adaptiveness

Agent is an entity that has the ability of consciousness, solving problem, self-learning and adapting to the environment. To have the agents learn, we may utilize the immune system whose response attributes of specificity, diversity, memory and self/non-self recognition are needed. Adaptiveness is a challenge and also an important feature for multiagent system to interact with the environment. Three major stages for IBMAS inspired by the clonal selection theory are diversity generation, self-maintenance and memory of nonself. The last two properties define the adaptiveness of the IBMAS. These steps are carried out by agents distributed over the MAS.

Diversity Generation. (Continuous) diversity generation leads to the "adaptation" of IBMAS. Diverse agents with distinct specificity of the receptor and the effector are generated by way of mutations.

Self-Maintenance. Agents are adjusted to be insensitive to known patterns (self) during the developmental phase. Negative selection theory is a central of this phase.

Memory of Nonself. Agents are adjusted to be more sensitive to unknown patterns (nonself) during the working phase.

2.3 Immune Network Theory

According to immune network theory, the interaction between various species of antibodies plays an important role in immune regulation; moreover, the immune system is composed of a superposition of a number of smaller network systems.

Idiotypic Immune Network. Many idiotypic immune network models focus on the interactions between antibodies and antigens. Jerne [6] proposed the idiotypic network theory in which cells co-stimulate each other in a way that mimics the presence of the antigen. .

An epitope of antigen A_g is recognized by the antibody molecule Ab_1 and by the receptor molecule on the lymphocyte of LU_1 (Fig. 1). The antibody Ab_1 and the receptor of LU_1 have the idiotope which is recognized by antibody Ab_2 and the receptor on the lymphocyte of LU_2. On the other hand, the antibody Ab_1 and the receptor on the lymphocyte of LU_1 also recognize idiotopes on antibody Ab_n. Ab_n constitutes an *internal image* of the antigen A_g. Network is formed by interactions between lymphocyte interactions. The epitope of antibody molecule is called idiotope

2.4 Activation Threshold

A lymphocytes can adjust its antigen response to the "context" in which the antigen is encountered. According to [3] [4] [5], context represents the physiological milieu and various quantitative and qualitative aspects of antigen presentations.

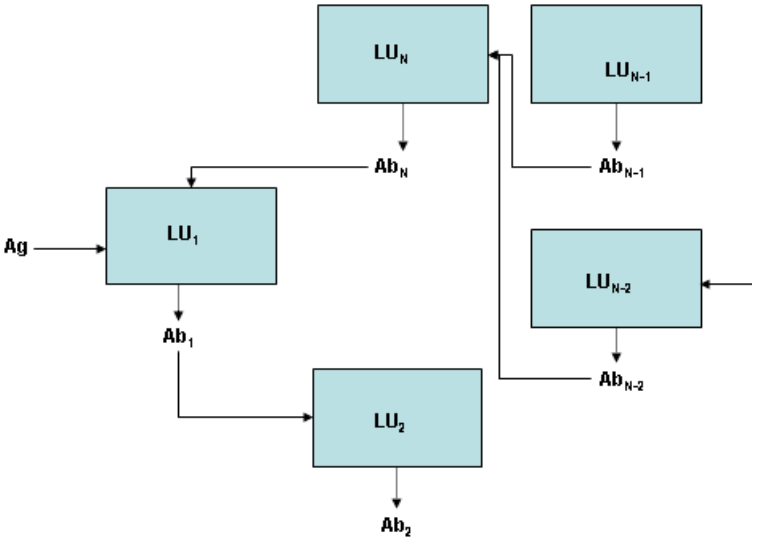


Fig. 1. Schematic diagram of the idiotypic network

Activation is a threshold phenomenon and the threshold is tuned by the stimulatory experience of the cell. It is believed that such adaptiveness of lymphocyte activation is characterized by specificity and memory [9].

While designing adaptive IBMAS, agents should be activated to respond to some foreign agents, called nonself agents (NAG). Such activation behavior is dynamic, namely, the activation threshold for each agent should be updated according to current and past events. Excitation index (of an agent) is defined by some time-dependent, weighted average of this agent's past excitation levels. Upon each excitation event, the agent undergoes a perturbation defined as the difference between the current excitation level and the excitation index. This index is the activity level which represents a cumulative interagent memory of the recent excitation events experienced by the agent. The concept of excitability implies that the existence of the short-term agent memory. Excitation index minus some constant is equal to the activation threshold.

2.5 Immune Memory and Learning Mechanism

Immune memory is one of the hallmarks of the immune system. One popular model is called the long-lived memory cell theory [10]. Some lymphocytes, both B- and T-cells, that have a close match to an antigenic source differentiate into 'memory cells'. These memory cells are then highly responsive to the original antigen. This theory assumes that memory cells live a very long time, thus preserving immunity for many years.

On the other hand, long-lived memory cell theory cannot explain the equilibrium states of immune systems while there is no antigen. In particular B-cells can perform suppression and activation functions without antigens. Idiotypic network theory describes this phenomenon [11]; moreover, it also explains the effective short-term memory of B-cells by generating internal images of detected antigens. Accordingly, immune memory leads to the learning mechanism of immune systems. Once the foreign antigen is removed, the immune system will restore some information of such antigen by this internal image mechanism. The effect of immune memory can contribute to the second immune response.

3 Adaptive Multiagent-Based Framework Inspired by Immune Network

Sathyanath and Sahin [8] proposed an artificial immune system based intelligent multiagent model (AISIMAM). Some disadvantages of AISIMAM model is related to its adaptiveness. In this section, we establish an adaptive IBMAS (AIBMAS) majorally characterized by immune network described in the previous section. Tolerance and memory are two major processes of adaptiveness to the self and nonself, respectively. Now we define the adaptiveness of an IBMAS.

Definition 1. *A multiagent system is adaptive, if it satisfies the property of self-maintenance and memory.*

3.1 AIBMAS Model

There is a set of agents, called lymphocyte agents (LA), in the initial stage of AIBMAS model. An agent which is not a LA is called a foreign agent; there are two types of foreign agents, namely, self agent (SAG) and nonself agent (NAG). LAs' missions are to detect any susceptible NAG (Fig. 2).

3.2 Dynamical Behaviors of AIBMAS

So far, AIBMAS is very similar to AISIMAM in MAS architecture. However, the adaptiveness of the latter needs to be improved. This goal can be reached by deploying ABTAT mechanism , CBR memory mechanism and immune algorithm based on idiotypic network within AIBMAS.

Agent-based Tunable Activation Threshold (ABTAT). AIBMAS improves the adaptiveness of AISIMAM by adopting several immune mechanisms. The first improvement is the activation threshold of agents. We propose that an adaptive agent can distinguish a "perturbation" from continuous stimulation and readjust its level of activation. One major assumption is that the activation threshold is not fixed for a given agent while subject to dynamic environment. The activation, which is defined as level of excitation (plus some constant), is dynamic. Such agent-based "tunable" activation threshold model (ABTAT) would allow agents to participate in its definition based one their own experiences.

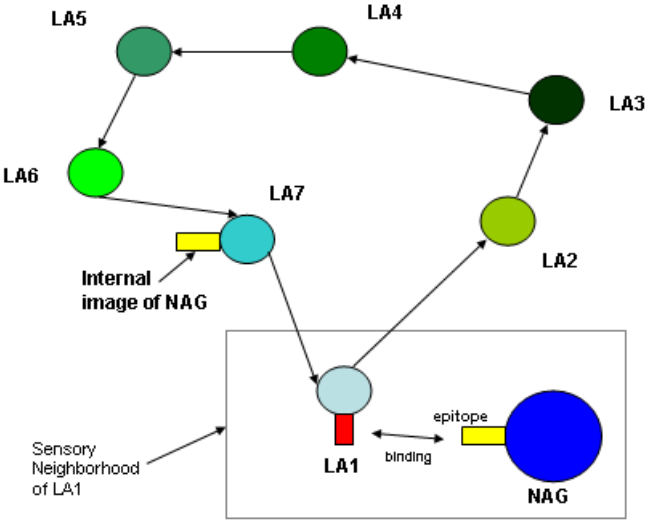


Fig. 2. Architecture of AIBMAS based on Idiotypic Network

Case-Based Reasoning on Dynamic Memory. Case-based reasoning (CBR) can be considered in parallel with the AIBMAS; particularly, it can fit Jerne's network paradigm (see Fig. 4.5. of [1]). Repertoire of LAs connected by idiotypic network is exactly a case knowledge database. The memory is dynamic since each antigenic activation will generate an internal image to some LAs according to idiotypic network.

3.3 Immune Algorithm in AIBMAS

Now the AIBMAS is functioning according some immune algorithm, which satisfies the adaptiveness of MAS. There are four phases for this AIBMAS algorithm, namely, initial phase, self-tolerance establishment, activation process and learning & memory.

Parameter Definitions. We define the Lymphocyte agent by LA_i, where $i = 1, 2, \ldots, N$. For each LA_i, there exists an $n-$dimensional information vector $B^i = [b_1, b_2, \ldots b_n]$. A foreign agent FA_j, there exists an $m-$dimenisonal information vector $A^j = [a_1, a_2, \ldots a_m]$. Define T_{ai} the (SNS) activation threshold of LA_i.

Functionalities of Immune Algorithm Phases

1. **Initial Phase.** Initialize all parameters of immune algorithm.
2. **Self-Tolerance Establishment.** For foreign agent, which is regarded as self agent (SAG) by LA according to some identifier. This identifier is connected to CBR database, or by associative memory mechanism of multiagent systems.

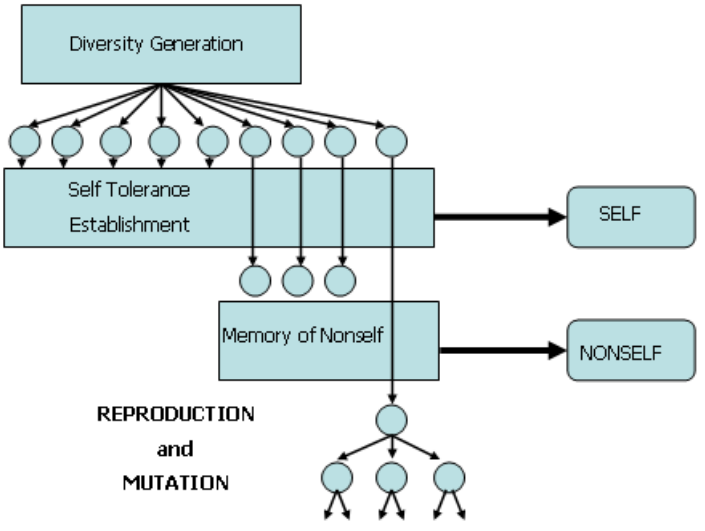


Fig. 3. Immune Algorithm for an agent-based Framework

3. Activation Phase. New actions are generated for SAGs; choose mature actions, then activate them.
4. Self Maintenance. SAG is cloned with mature actions, then added to the family of LA.
4. Learning and Memory. For foreign agent which is regarded as an NAG, the related information is transmitted to CBR.

Now we describe the immune algorithm in more details.

Initial Phase. The main operation in the initial phase is the diversity generation. For each LA_i, define LA^i_{nbd} as its sensory neighborhood.

- For each FA_j in LA^i_{nbd}, calculate $M_{ij} = f(B^i, A^j)$, where B^i is the message string of LA_i, and A^j is the message string of FA_j. The matching function f is defined as follows. $f(B^i, A^j) = 0$, if FA_j is not in the LA^i_{nbd}, otherwise, equal to 1.
- If $M_{ij} = 1$, then the information about the FA_j is transmitted to the other LAs according to the idiotypic immune network mechanism.

Assume that $FA_j \in LA^i_{nbd}$, where $j = 1, 2, \ldots, e$.

Phase of Self-Tolerance Establishment. Identify these FA_j detected by LA_i in initial phase using the identifier function *iden* defined as follows.

$$I_j = iden(FA_j) \tag{1}$$

If $d(B_i, A_j) > K_{match}$, then FA_i is regarded as a NAG; otherwise, a SAG. d is the Euclidean distance, and K_{match} is the activation threshold of LA_i.

Activation Phase. For an NAG, using action generation function *ActGen* to generate possible k new actions according to identifier I_j as follows.

$$U_l^j = ActGen(I_j), l = 1, 2, \ldots k. \tag{2}$$

Find the affinity for all possible vectors U_i^j by the affinity function

$$Af_l^j = affin(U_l^j), l = 1, 2, \ldots k. \tag{3}$$

Choose mature actions whose affinity is greater than the (action) threshold value T_a. The mature action set Y is thus defined as follows.

$$Y = \{U_l^j | Af_l^j > T_a\}, l = 1, 2, \ldots p. \tag{4}$$

Rearrange the index l in necessary. The activation of the mature actions within the time t_b (called binding time) is given by

$$U_i^j = g(Y, t_b) * [u(t) - u(t - t_b)] \tag{5}$$

u is the unit step function. If there is an activation, $g = 0$, otherwise, $g \neq 0$. T_a can be tuned so that $p = 1$, this is the case for the best action.

Self-Maintenance. If FA_j is a SAG and $U_i^j \neq 0$, it is cloned with mature action set Y to generate q number of this SAG, say S_z, where $z = N+1, N+2, \ldots, N+q$. Define $LA_z = S_z$.

Learning and Memory of Nonself

Learning. If FA_j is an NAG, then its information vector A_j is sent to the CBR database. This database will update the record of FA_j according to (SNS) activation threshold K_{mat} by the following equation:

$$K_{mat}(t + 1) = k_{mat}(t) + \alpha E(t)(E(t) - k_{mat}(t)), t = 1, 2, \ldots, t_b \tag{6}$$

where E is the excitation level of LA_i, and t_b is the binding time between LA_i and FA_j.

Memory. LA_i will generate an internal image of FA_j restored at other LA_r. The information vector of LA_r is updated by adding A_j.

3.4 Controllability of AIBMAS

One purpose of controlling the LA's activation threshold (equivalently excitation level) is to avoid the abnormal behaviors such as false negative. We notice the effect of parameter α of (6) with lower value producing an excitation index curve that tunes at a lower rate. This parameter determines how quickly the excitation index tunes, the smaller it is, the smaller the increment to the excitation index and the slower it tunes to the value of the excitation. α is therefore controlling the memory effect of the activation threshold; the lower the α value the longer term memory of past excitations.

4 Model Evaluations

In this section, we evaluate the AIBMAS by comparing it with AISIMAM. For Table 1, advantages of AIBMAS are listed with respect to AISIMAM.

Table 1. Comparisons between Immunity-based Multiagent Systems

Algorithm	AIBMAS	AISIMAM
Diversity Generation	Yes	Yes
Self Tolerance	Yes	No
Learning	Yes	No
SNS Act. Threshold	Yes	Yes
Self Maintenance	Yes	Yes
Short-term Memory	Yes	Yes
Long-term Memory	Yes	No

These two algorithms deploy functionalities such as diversity generation and short-term memory (of NAGs). However, AIBMAS equips with long-term memory mechanism by generating internal image restored in LAs. AISIMAM depends on memory cells, which are short-lived, to restore antigenic information.

Table 2 is a comparison of functionalities between AIBMAS and AISIMAM.

Table 2. Functionalities of Immunity-based Multiagent Systems

Algorithm	AIBMAS	AISIMAM
Self-Tolerance	Action generator	Action generator
Memory	internal image, TAT	memory cells
behavior management	action generators	action generator
Network Type	P2P	P2P
Agents proliferation number	small	large

According to this table, AIBMAS proliferate fewer SAGs than AISIMAM. From agent management viewpoint, the former has advantage over the latter.

5 Conclusions

We propose an adaptive immunity-based multiagent system (AIBMAS). The intelligence behind such system is based on the learning and memory mechanism of immune network systems. Lymphocyte agents detect foreign agents and decide whether they are SAGs or NGAs according to the immune algorithm. The latter can generate dynamic activation threshold according to agent environment. One challenge is the information transferring mechanism by idiotypic immune network. Another interesting issue is the associativity of the agent learning and memory provided by idiotypic immune network. Whether an agent can perform associative memory is worth of future research.

References

1. Ishida, Y.: Immunity-Based System, A Design Perspective. Springer, Heidelberg (2004)
2. Andrews, P., Timmis, J.: Adaptable lymphocytes for artificial immune systems. In: Bentley, P.J., Lee, D., Jung, S. (eds.) ICARIS 2008. LNCS, vol. 5132, pp. 376–386. Springer, Heidelberg (2008)
3. Grossman, Z.: Cellular Tolerance as a Dynamic State of the Adaptable Lymphocyte. Immunological Reviews (133), 45–73 (1993)
4. Grossman, C., Gruy, F., Haudin, C.-S., El Hentati, F., Guy, B., Lambert, C.: Mathematical Modeling of T-Cell Activation Kinetic. Journal of Computational Biology 15(1), 105–128 (1992), doi:10.1089/cmb.2007.0125
5. Grossman, Z., Paul, P.: Adaptive cellular interactions in the immune system: the tunable activation threshold and the significance of subthreshold responses. Proc. Natl. Acad. Sci. USA 89, 10365–10369 (1992)
6. Jerne, N.: Ann. Immunol. (Inst. Pasteur) 125C, 373 (1974)
7. Hoffman, G.W.: A theory of regulation and self-nonself discrimination in an immune network. Eor. J. Immunol. 5, 638–647 (1975)
8. Sathyanath, S., Sahin, F.: AISIMAM-An Artificial Immune System Based Intelligent Multi Agent Model and its Application to a Mine Detection Problem. In: ICARIS 2002 (2002)
9. Chan, C., George, A., Stark, J.: T Cell Sensitivity and Specificity - Kinetic Proofreading Revisited. Discrete and Continuous Dynamical Systems Series B 3(3), 343–360 (2003)
10. Robbins, M., Garrett, S.: Evaluating Theories of Immunological Memory Using Large-Scale Simulations. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 193–206. Springer, Heidelberg (2005)
11. Parisi, G.: A Simple Model for the Immune Network. Proceedings of the National Academy of Sciences 87, 429–433 (1990)

Distributed Default Logic for Context-Aware Computing in Multi-Agent Systems*

Dominik Ryżko and Henryk Rybiński

Warsaw University of Technology, Institute of Computer Science, Nowowiejska 15/19
00-665 Warsaw, Poland
{d.ryzko,hrb}@ii.pw.edu.pl

Abstract. The paper describes how Distributed Default Logic (DDL) can be used as a formalism for context-aware computing in a Multi-Agent System. It is shown that the original notation does not require any changes. The DDL reasoning engine has been adapted to handle situations like unavailability of sensors. New semantics of Distributed Default Rules in the application to reasoning with context information is also described.

Keywords: multi-agent systems, default logic, context-aware computing.

1 Introduction

With the rapidly growing number of computing devices and mobility of their users, the importance of context-aware computing is growing very fast. We want various services to be available any time and in any place we visit. This poses formidable challenges on the software that needs to adapt to the changing context of the user.

Context awareness as a term has originated from ubiquitous computing, which was sought to deal with linking changes in the environment with computer systems, which are otherwise static. The term context can be defined as any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves [6].

We use context all the time when taking daily decisions. Let us take an everyday example of choosing the right clothing before going out of home. We can use simple commonsense rules like "If it is summer and no indications of bad weather, we can wear a t-shirt". The rule will work fine in most locations we visit. To be able to apply it we need sensors, which will give us additional context information to trace any exceptional situations, which should prohibit us from going out unprepared. There are several potential context sources – thermometer, barometer, our eyes, internet weather service, tv weather forecast or even coin tossing. Depending on our experience some of them might be preferred over others, while depending on our location only a

* The research was supported by the Polish National Budget Funds 2009-2011 for science under the grant N N516 3757 36.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 476–483, 2010.

limited number of them might be available at a particular time. In this paper we will show how to perform such reasoning in a Multi-Agent System.

Multi-Agent Systems are especially well suited to model distributed, ubiquitous environments with high need for adaptability driven by context-awareness. By introducing a concept of intelligent, autonomous and proactive agents, the system is able act flexibly and effectively. Autonomy of its parts allows for differentiation of computation based on different context values. Proactiveness, on the other hand, can be used to find new context sources and monitor the ones that are already available.

DDL is an extension of Default Logic as defined by Reiter [8], designed especially to support distributed reasoning in a Multi-Agent System [9]. Its main feature is embedding of environmental information into rules, to speed up the reasoning process. The rules contain links to other agents, who possess relevant information. The distributed algorithm answers queries by computing a preferred extension of the global theory. We show how this formalism can be used for reasoning about context.

2 Previous work

There are several approaches to context-aware computing in a distributed environment. Lots of effort has been done towards definition of context. Apart from the definition mentioned in the Introduction, others can be found [11], [12].

It has been argued that for efficient implementation of context-aware computing, a middleware software is needed [7]. In this paper we will not consider how context is measured and delivered to the agents. Rather, we concentrate on application of a dedicated formalism, which allows for efficient computation of context information.

In [3] a context-based agent architecture has been proposed. An ontology-based representation for context elements is introduced. Since the system has been designed for handheld devices, an external context reasoning layer is introduced. Chen [4] proposes a broker-centric agent architecture called Context Broker Architecture (COBRA) in order to reduce the cost of building context-aware systems.

Default Logic or its variations has already been used in MAS. Apart from our earlier work other approaches can be found in [10], [2]. The first approach proposes social default theories. It can be used for representing various social attitudes of agents, including cooperative planning and negotiations. The second paper describes how default reasoning agents can be applied with reasoning by case capabilities.

3 Distributed Default Logic (DDL)

This chapter presents definitions related to the Distributed Default Logic, which was designed to facilitate distributed commonsense reasoning in a Multi-Agent System.

3.1 Definitions

We will start with brief reminder of Reiter's original approach. A default theory Δ is described as a pair (D, W), where D represents a set of default rules while W is a set first-order formulas.

Definition 1: A default rule d is a rule of the form

$$\frac{\alpha(x) \mid \beta_1(x),\ldots,\beta_m(x)}{\omega(x)}$$

where $\alpha(x)$, $\beta_1(x)$, ..., $\beta_2(x)$, $\beta_m(x)$ are all classical logical formulae. $\alpha(x)$ is called prerequisite - p(d); $\beta_1(x)$, ..., $\beta_m(x)$ is justification - j(d) and $\omega(x)$ is consequent - c(d). If p(d) is known and j(d) is consistent with W then c(d) may be inferred.

Definition 2: Let $\Delta = (D, W)$ be a default theory. For any set of formulae S**,** let $\Gamma(S)$ be the smallest set satisfying the following conditions:

(i) $W \subseteq \Gamma(S)$

(ii) $Th(\Gamma(S)) = \Gamma(S)$

(iii) If $(\alpha : M\ \beta_1, ..., \beta_n/\omega) \in D$, $\alpha \in \Gamma(S)$ and $\neg\beta_1, ..., \neg\beta_n \notin S$, then $\omega \in \Gamma(\mathbf{S})$

A set Ext is an *extension* of default theory Δ iff $\Gamma(Ext) = Ext$.
Now we are ready to present DDL.

Definition 3: A Multi-Agent System (MAS) is a collection of agents operating in the same environment.:

$$MAS = \{A_1, A_2, ..., A_n\}$$

Definition 4: *Extended default template T* is the rule of the form

$$\frac{\alpha(x) : \beta(x)_1^{L1},\ldots,\beta(x)_m^{Lm}}{\omega(x)}$$

where $L_k \subseteq MAS$ and α, β and ω have the same meaning as in standard defaults.

The reason for indexing justifications is to allow agents to keep information about agents as metadata within the rules, which can help them to justify their assumptions. When applying the rule, only single agent per justification will be chosen, which leads to creation of *extended default rule*. This process, called *materialization,* will lead to no more then one agent signature attached to each of the justifications.

Definition 5: *Extended default rule* is a following rule

$$\frac{\alpha(x) : \beta(x)_1^{l1},\ldots,\beta(x)_m^{lm}}{\omega(x)}$$

where $l_k \in L_k$ and α, β and ω have the same meaning as in standard default rules.

Definition 6: We say that an extended rule r is a materialization of T iff

(i) $p(r) = p(T)$

(ii) $c(r) = c(T)$

(iii) $\forall \beta_m^{Lm}(x) \in j(T)\ \exists\ \beta_m^{lm}(x) \in j(r) : l_m \in L_m$

where p(r), c(r), j(r) are prerequisite, consequence and justification of the rule r respectively.

We can now define the Distributed Default Theory in the following way

Definition 7: *Distributed Default Theory (DDT)* is a set $\{\Delta_1, \Delta_2, \ldots \Delta_n\}$, where index *n* enumerates nodes of the system and $\Delta_i = (D_i, W_i)$ is a default theory stored at node $i \in$ <1, n>. Each of the sets D_i contains Extended Default Templates.

By D*istributed Default Logic* (DDL) we will understand the D*istributed Default Theory* together with the process for D*istributed Default Reasoning* (DDR).

The rule described in the Introduction, can be written in DDL in the form of the following distributed default template:

$$\frac{summer : \neg cold^{termometer, internet, tv} \wedge \neg rain^{internet, tv, barometer} \wedge \neg wind^{internet, tv, eyes}}{wear\,t-shirt}$$

If we have no internet or tv, the materialization of the above template in the form of DDR will look like this:

$$\frac{summer : \neg cold^{termometer} \wedge \neg rain^{barometer} \wedge \neg wind^{eyes}}{wear\,t-shirt}\ .$$

3.2 Reasoning with DDT

In DDT only one extension of the given theory is generated. This is done by prioritizing defaults based on a confidence function. For each agent the following confidence function will be maintained

$$C_n: MAS \rightarrow [0;1]$$

with additional restriction that for *i-th* agent, $A_i \in MAS$

$$C_i(A_i) = 1$$

In the case of sensors, an accuracy measure can be used to estimate how reliable a particular source of context is.

Definition 8: *Priority* P_d of DDR $d = \alpha : \beta_1^{A1}, \ldots, \beta_n^{An} / \gamma$ is

$$P_d = \prod_{i=1\ldots n} C(A_i)$$

The confidence function is just one of the properties of other entities that can be maintained by an agent. Depending on the purpose of the system we can measure time of response, price etc. We will define the following property matrix which, together with the confidence function, will provide a guidance for choosing the other agents (sensors) for cooperation.

Definition 9: For *n* agents with *m* properties we define a *property matrix*

$$\begin{bmatrix} v_{1,1} & v_{1,2} & \dots & v_{1,n} \\ v_{2,1} & v_{2,2} & \dots & v_{2,n} \\ \dots & \dots & \dots & \dots \\ v_{m,1} & v_{m,2} & \dots & v_{m,n} \end{bmatrix}$$

where $v_{i,j}$ is a value of property i for agent j. Therefore, each column represents one agent, while each row refers to a single property.

We might as well have included the confidence function as one of the properties in the property matrix. However as it is shown in the paper, this property is so distinct in the way it is used and calculated that we decided to keep it separately to underline its significance.

Agent properties can be used to exclude some of the agents from the list in DDT, based on some additional constraints (e.g. cost, time etc.).

We will now partly follow the terminology introduced by [1] for partitioning of FOL and propositional theories as well as by [5] who introduced stratified default theories.

Let *A* be a Distributed Default Theory divided into n partitions

$$A = \cup_{i=1..n} A_i$$

$\{A_i\}_{n \le i}$ will be called a *partitioning* of default theory *A*. By $L(A_i)$ we will denote the signature of a partition (the set of non-logical symbols). $\angle(A_i)$ will be the set of formulae built with $L(A_i)$ called the language of A_i.

Each of the partitions will contain a portion of global default theory

$$(D, W) = \cup_{i=1..n} (D_i, W_i)$$

Each partitioning can be represented by a labeled and directed graph G = (V, E, l), which we will call the *intersection graph*. In this graph, each node corresponds to the individual partition A_i, so V={1..n}. Two nodes *i, j* are linked by the oriented edge leading from *i* to *j* if

$$\exists d_i \in D_i,\ d_j \in D_j,\ p \in \angle(D,W) : p \in c(d_i) \wedge p \in p(d_j)$$

or

$$\exists d_j \in D_j,\ p \in \angle(D,W) : p \in W_i \wedge p \in p(d_j)$$

The edges are labeled with the set of symbols that the associated partitions A_i and A_j share ($l(i, j) = L(A_i) \cap L(A_j)$). *l(i, j)* will be called *communication language* between partitions A_i and A_j.

Definition 10. The intersection graph G satisfying the following conditions will be called *well-ordered iff*:

(iv) G is a tree

(v) There exists such a numbering of nodes, that all edges lead from a lower to a higher value of this numbering

(vi) all references leading from justification of default templates lead to a node with a higher number

(vii) there are no conflicts such that there are two nodes with the same symbols in conclusions of their rules

The numbering (ii) of a well-ordered intersection graph is called *well-ordered numbering.*

In [9] we show that a DDT with well-ordered intersection graph is stratified according to well-ordered numbering and that a distributed reasoning algorithm can generate extension efficiently processing one strata at a time.

4 Context-Aware Reasoning with DDL

DDL with little modifications can be easily adapted for context-aware computing. Firs of all we need to give context semantics to the DDR. We want agents to be able to reference both internal and external sources of context. Therefore, we will use agent references stored in DDT as references to different sensors. This does not require any changes to the DDL. A sensor possessed by a reasoning entity can be modeled as an additional simple agent with knowledge base limited to the current sensor reading.

What we need to add are means for reasoning in case of unavailability of some of the sensors. DDL initially assumes all entities can participate in the reasoning process. In real life it is often the case that we cannot access some sensor reading. Sometimes we have to manage without our preferred source of context. The problem is that context-aware computing requires by definition high efficiency. In case of a missing agent (sensor) we cannot afford to recalculate all the rule priorities and start all over the reasoning process.

Therefore we propose the following approach. If a sensor is missing, the next preferred sensor which can provide the same information is used. If the needed context value cannot be accessed at all, the next preferred rule is applied. Such approach is semi-optimal according to the calculated confidence measures, but it is fast and is in line with our commonsense behavior. In the case of our dressing example, if we cannot get a thermometer reading we will try to find out the temperature in some other way. Usually, only after we use all available source of information and fail, we will try to apply some other knowledge for choosing the right clothing.

The reasoning process must take all of the above into account. Below a modified version of distributed default reasoning algorithm is presented:

```
Algorithm 1: Generate extension
Input: (D_n, W_n)
Output: Ext (D_n, W_n)
Begin
  DR = Materialize (D_n) //generate rules from templates
  Loop//main loop
   Rules = DR
   Loop//rule loop
    If (Rules empty)
      Exit rule loop
    End if
    R = GetRule (Rules) //Get rule with highest priority
                        //and remove from Rules
    If (p(R) in W_n and j(R) is consistent with W_n)
      For (J from j(R)) loop //justification loop
     //send query to the agent referenced in
justification
        Do
          agent = get_next_agent(j(R))
          If (no more agents) exit rule loop
          Send (agent, J)
        While (not available agent)
      End loop
      If (no justifications inconsistent)
        Add (conclusions, W_n)
        Add (conclusions, Result)
        Exit rule loop
      End if
    End if
   End loop//end of rule loop
   If (nothing new in Result) exit
  End loop//end of main loop
  Send (Result, parent agent)
End
```

5 Conclusions and Future Work

We have shown how a DDL formalism initially designed for distributed, common-sense reasoning in a multi-agent system, can be adapted for context-aware computing. References to other agents can be used to reference sensors and other context sources. In this way the context information is captured naturally within the formalism.

The reasoning algorithm has been slightly modified in order to take into account situation where some sensors become unavailable. This allows to continue the reasoning without the need of reorganizing the whole system, even if some communication problems between its components occur.

The approach presented in the paper extends the original formalism without limiting any of the capabilities it was designed to possess. It is possible to combine links to knowledge stored by other agents with the links to sensor information. This means, that within one distributed reasoning process context of different agents can be taken into account.

References

1. Amir, E., McIlraith, S.: Partition-based logical reasoning for first-order and propositional theories. AI 162, 49–88 (2005)
2. Besnard, P., Gregoire, E.: About agents that reason by case. In: IEEE International Conference on Information Reuse & Integration, IRI 2009, pp. 405–410 (2009)
3. Bucur, O., Beaune, P., Boissier, O.: Representing context in an agent architecture for context-based decision making. In: Proceedings of the Workshop on Context Representation and Reasoning (CRR 2005), Paris, France (2005)
4. Chen, H., Tolia, S.: Steps towards creating a context-aware agent system. Technical report, Hewlett Packard Labs (2001)
5. Cholewinski, P.: Reasoning with stratified default theories. In: Proc. of 3rd Int'l. Conf. on Logical Programming and Nonmonotonic Reasoning. Cambridge University Press, Cambridge (1995)
6. Dey, A.: Understanding and Using Context. Personal Ubiquitous Computing 5(1), 4–7 (2001)
7. Ranganathan, A., Campbell, R.H.: A middleware for context-aware agents in ubiquitous computing environments. In: Endler, M., Schmidt, D.C. (eds.) Middleware 2003. LNCS, vol. 2672, pp. 143–161. Springer, Heidelberg (2003)
8. Reiter, R.: A logic for default reasoning. Artificial Intelligence 13, 81–132
9. Ryzko, D., Rybinski, H.: Distributed Default Logic for Multi-agent System. In: Proceedings IAT 2006 (2006)
10. Sakama, C.: Social Default Theories. In: Erdem, E., Lin, F., Schaub, T. (eds.) LPNMR 2009. LNCS, vol. 5753, pp. 470–476. Springer, Heidelberg (2009)
11. Schilit, B., Adams, N., Want, R.: Context-aware computing applications. In: Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, California, pp. 85–90. IEEE Computer Society Press, Los Alamitos (1994)
12. Schmidt, A., Aidoo, K.A., Takaluoma, A., Tuomela, U., Van Laerhoven, K., Van de Velde, W.: Advanced interaction in context. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707, pp. 89–101. Springer, Heidelberg (1999)

A Novel Approach to Default Reasoning for MAS

Przemysław Więch and Henryk Rybiński

Warsaw University of Technology, Institute of Computer Science
Nowowiejska 15/19 00-665 Warsaw, Poland
{pwiech,hrb}@ii.pw.edu.pl

Abstract. In a multi-agent system the sought information can often be found across various knowledge bases, which means that making early assumptions can lead to hasty conclusions. In the paper we present a formalism for distributed default reasoning to be performed by a group of agents that share knowledge in the form of a distributed default theory. The formalism is based on default transformations, which can be used to derive answers to queries in the form of defaults. Such new defaults can then be treated as intermediate results in the reasoning process. It is shown that passing messages containing transformed defaults is more informative than strict statements and enables avoiding early conclusions. Moreover, the extended reasoning features are embedded in the description logic framework.

Keywords: multi-agent system, default logic, description logic, distributed reasoning, Distributed Description Logic.

1 Introduction

Many real world applications require knowledge, which is distributed and is located aside of the entity assigned to solve the given problem. Such entities must be able to cooperate in order to reach solutions of the problems presented to them. This is actually the approach of multi-agent systems (in the sequel MAS), which provide tools for modelling the situations by means of a set of collaborating autonomous, intelligent and proactive agents. Examples of such applications of MAS in the area of the energy markets are shown in [10].

Knowledge sharing in a distributed environment is essential. Recently, the area of Agent-Mediated Knowledge Management has emerged, which considers Knowledge Management in a multi-agent setting. A shift of interest can be observed from traditional knowledge management to the cooperation of distributed and often heterogeneous sources of knowledge [9].

In the Semantic Web knowledge is distributed throughout the Web and it can only be seen as a network of agents, each having its own knowledge base and reasoning facilities. The entities can have specialized knowledge, which can be shared and reused by agents that need to collect remote information in order to perform a reasoning task. Distributed reasoning in a peer-to-peer setting is shown in [1], where a message passing algorithm is introduced to exchange knowledge

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 484–493, 2010.

between peers. Each peer runs an inference procedure on local knowledge to answer queries from neighbouring peers. Here, in contrast to other approaches, the global theory defined as a sum of all local knowledge is unknown.

In [15,14] a multi-agent system is proposed for knowledge sharing in an environment of agents equipped with default reasoning abilities. The Distributed Default Logic framework (DDL) is composed of agents having their knowledge in the form of default logic theories, and able to communicate with each other in order to resolve the locally unknown facts.

In a distributed default logic system, exchanging information between agents in the form of facts may cause loss of valuable information about the assumptions made during the process of reasoning. Thus, we argue that it is beneficial to enable the agents to exchange information in the form of defaults as these contain additional information about default justifications. In the paper we present the formalism of default transformations together with an algorithm for deriving defaults as inference results from a default theory. We show that this approach can be integrated with description logic in a multi-agent system.

2 Related Work

Logic is often used as the basis for knowledge representation in multi-agent systems. In [11] Kowalski and Sadri describe an extension of logic programming to provide rationality and reactiveness in the multi-agent setting.

In a distributed environment the knowledge is scattered among the agents. The field of theory partitioning studies the methods of dividing a logical theory in order to increase the efficiency of reasoning. Amir and McIlraith [2] introduce forward and backward reasoning algorithms for a partitioned first-order logic theory. Here, message passing is used to transfer knowledge between partitions.

Distributed reasoning with defaults has been introduced in [15,14]. The formalism named *Distributed Default Logic* extends Reiter's Default Logic [12] by defining a *distributed default theory*. The approach adopts the stratification of default theories, which has been considered by Cholewinski [7].

Distributed reasoning is essential for the domain of the Semantic Web as the knowledge is inherently distributed among many sources. The Semantic Web bases its knowledge representation on Description Logics (DLs) [3]. On the grounds of the DL formalisms several approaches to mapping distributed knowledge bases have been investigated [6,8,5]. Our work extends the notions of *Distributed Description Logic* by introducing defaults to the knowledge representation formalism and to the inference procedure.

3 Basic Concepts

3.1 Default Logic

Default logic has been introduced by Reiter [12] as an approach to commonsense reasoning. It can be used to deal with the inability to fully describe the world

and to provide more concise representations of knowledge due to the form of specifying exceptions to defaults. A **default** is in the form $\frac{\alpha:\beta}{\gamma}$ where α, β and γ are well-formed formulae. α is the **prerequisite**, β is the **justification** and γ is the **consequent**. The default can be applied and the consequent inferred if the prerequisite can be proven and the justification is consistent with the current knowledge. For a default d, let $Pre(d)$, $Jus(d)$ and $Con(d)$ denote the formulae occurring in the prerequisite, justification and consequent, respectively, of the default d. We say that a default is **closed** if it contains no free variables.

Given a set W of first-order logic formulae creating a world description, and a set D of defaults, we define a **default theory** as a pair $\Delta = \langle D, W \rangle$. The default theory is **closed** if it contains only closed defaults.

The inferences of default logic are defined by means of extensions. Extensions can be obtained by applying a non-deterministic iterative process to a default theory. In each step a default is used to add the consequent to the resulting set of formulae. An extension is defined by the fixed point of this process. Let E be a set of closed formulae, and $\langle D, W \rangle$ be a closed default theory. By $Th(E)$ we denote the deductive closure of a set of formulae E. Let $E_0 = W$ and

$$E_{i+1} = E_i \cup \left\{ \gamma \mid \frac{\alpha : \beta}{\gamma} \in D, \alpha \in Th(E_i) \text{ and } \beta \notin Th(E) \right\}$$

$Th(E)$ is an **extension** of $\langle D, W \rangle$ iff $Th(E) = \bigcup_{i=0}^{\infty} Th(E_i)$

There can be many extensions of a default theory depending on the order the defaults are applied. By $ext(\Delta)$ we denote the set of all extensions of the default theory Δ.

Definition 1. [12] *The set of* **generating defaults** *for an extension E of theory $\Delta = \langle D, W \rangle$ is the set*

$$GD(E, \Delta) = \{\alpha : \beta / \gamma \in D | \alpha \in Th(E) \text{ and } \neg\beta \notin Th(E)\}$$

3.2 Embedding Defaults into Description Logics

Description logics (DLs) [3] are a family of knowledge representation formalisms. Knowledge in DLs is represented by defining concepts from a selected domain, which comprise a terminology, and using these concepts for classifying objects and describing their properties.

A description logics knowledge base consists of only universal statements, which do not allow exceptions. This allows the reasoning system to unambiguously assign individuals to concepts. However, this method does not provide means for commonsense reasoning, where some assumptions can eventually be shown to be false. The application of the results achieved in default logic can provide a method for commonsense reasoning without losing important features of description logics. Reiter's default logic uses first-order logic as the base language and since description logics are decidable subclasses of FOL, they can be extended with the notion of defaults using the original semantics. Baader and Hollunder [4] show how defaults can be embedded into description logics.

Definition 2. *A default is in the form* $\frac{A:B}{C}$ *where A, B and C are concept expressions. This default is equivalent to the default in which concepts are expressed as unary predicates* $\frac{A(x):B(x)}{C(x)}$

The default expresses that it can be inferred that x is an instance of the concept C if x is an instance of A and it is consistent to assume that x is an instance of B.

Embedding defaults in description logics is not as straightforward as it may seem. The problem is with treatment of open defaults by Skolemization. A terminological knowledge base with an even smaller set of constructors than OWL-DL is undecidable, unless we consider only closed defaults. This means that defaults can only be applied to named individuals which already exist in the knowledge base.

A normal default in the form $\frac{A:B}{C}$ can be seen as a weaker form of subsumption, such that permits exceptions. The default $\frac{A:\top}{B}$ is also weaker than the axiom $A \sqsubseteq B$ because it, being a rule, does not imply its contrapositive $\neg B \sqsubseteq \neg A$.

Because of problems caused by Skolemization which are described in [4], restricted semantics has to be applied. Defaults are only applied to explicitly referenced individuals forming a finite set of closed defaults.

4 Distributed Reasoning with Defaults

4.1 Motivation

One of the main reasoning tasks in description logics is subsumption checking. The query of whether a concept B subsumes a concept A is stated in description logic language as $A \sqsubseteq B$, which corresponds to the statement $A(x) \rightarrow B(x)$ in first-order logic. Such queries are useful when reasoning in a distributed environment such as a multi-agent system. For instance, the distributed reasoning algorithm presented in [16] exchanges such queries between the peers in a Distributed Description Logic system. In this setting the answer to a query can only be *true* or *false*.

Taking into account the integration of defaults into distributed reasoning, giving such a definite answer to a subsumption query does not always fully express the knowledge contained in a remote knowledge base and can cause loss of valuable information about the assumptions made during reasoning.

Example 1. Let us consider an example knowledge base which can be a part of a distributed system.

$$d_1 : \frac{\mathsf{Bird} : \mathsf{Flies}}{\mathsf{Flies}}$$

$\mathsf{Stork} \sqsubseteq \mathsf{Bird}$	$\mathsf{Stork}(\mathsf{SAM})$
$\mathsf{Goose} \sqsubseteq \mathsf{Bird}$	$\mathsf{Stork}(\mathsf{TIM})$
$\mathsf{Penguin} \sqsubseteq \mathsf{Bird}$	$\neg\mathsf{Flies}(\mathsf{TIM})$
$\mathsf{Penguin} \sqsubseteq \neg\mathsf{Flies}$	$\mathsf{Penguin}(\mathsf{PAT})$

Let us consider the following queries to this knowledge base:

Case 1: $Q1:\ \mathsf{Flies(TIM)}$ $A1:\ false$

The answer to this query is *false* because there is a straightforward fact in the knowledge base stating that TIM does not fly.

Case 2: $Q2:\ \mathsf{Flies(SAM)}$ $A2:\ true$

A positive answer is returned by applying the default.

Case 3: $Q3:\ \mathsf{Stork} \sqsubseteq \mathsf{Flies}$ $A3:\ false$

If this query is interpreted as "Do all storks fly?", the answer should be negative, as TIM is an example of a stork that does not fly.

Case 4: $Q4:\ \mathsf{Goose} \sqsubseteq \mathsf{Flies}$ $A4:\ undefined$

For this case the answer cannot be unambiguously stated. On the one hand, the knowledge base cannot provide a negative example of a goose that does not fly, so a negative answer cannot be given. On the other hand the open world assumption does not permit giving a positive answer because there may exist geese that cannot fly. The default can only be applied in its closed form with a concrete individual and cannot be applied without instantiating its variable.

From this example we can see that the answers to queries $Q2 - Q4$ lose information which exists in the form of the default. In a distributed system, where many knowledge bases can contain the sought information, making early assumptions can lead to too hasty conclusions. What could be expected in these cases are answers with the following meanings:

$A2'$. *It is assumed that* SAM *flies unless it is proved otherwise.*

$A3'$. *Typically, storks fly unless it is proved otherwise.*

$A4'$. *Typically, geese fly unless it is proved otherwise.*

These statements can be expressed as the following defaults:

$$A2':\ \frac{:\ \mathsf{Flies(SAM)}}{\mathsf{Flies(SAM)}} \qquad A3':\ \frac{\mathsf{Stork} : \mathsf{Flies}}{\mathsf{Flies}} \qquad A4':\ \frac{\mathsf{Goose} : \mathsf{Flies}}{\mathsf{Flies}}$$

These defaults, if provided as answers, give more information from the original knowledge base than usual answers. The rules form a concise intermediate result and can be triggered to achieve the final answer. In a distributed environment the triggering of these rules will occur on the side of the asking agent. Its own knowledge gathered from other sources can be useful for providing justifications for defaults or rejecting defaults based on provided exceptions to defaults.

As shown in [13], combining defaults and implication, although semantically correct, can lead to conclusions that are not intended. For example, the statements *Typically adults are married* and *18-year-olds are adults* leads to a default *Typically 18-year-olds are married*. However, such situations can be solved by adding additional defaults or adding justifications to existing ones.

In a distributed environment the fact that only closed defaults have to be used in order to reason is very limiting. This would imply that two agents sharing knowledge have to have a common set of individuals and one agent would not be able to ask another agent about a general relationship $A(x) \rightarrow B(x)$ without instantiating the variable x.

In the next section we will discuss what transformations can be applied to defaults while an agent prepares answers in the form of defaults.

4.2 Transforming Defaults

Exchanging knowledge between agents in an efficient way requires an agent to answer a question as precisely as possible. When agent A asks agent B whether it believes that formula ϕ is true, it expects a short answer whether ϕ is true or not. However, when using defaults in the reasoning process the answering agent might use defaults while finding the answer to a question. This leads to making possibly wrong assumptions. Giving a strict answer of $true$ or $false$ would make the asking agent interpret the answer as "Agent B **believes** that ϕ is true (false)" while agent B only **assumes** that ϕ is true (false).

In order to deal with such situations the answer to an agent's query should also carry the information about assumptions made during the reasoning process. In default logic, assumptions are expressed through the use of justifications in defaults. By tracing the justifications of defaults that would be triggered when trying to prove a formula, additional information can be collected and further used in answers to queries. For a query in the form $a \rightarrow b$ we will allow an answer in the form $\frac{a:b \wedge j}{b}$, where j is the conjunction of justifications which have to be verified in order to infer b.

Defaults in Reiter's default logic are treated as inference rules on the same level of reasoning as *modus ponens* or *modus tollens*. In the basic form the inference methods do not permit creating new inference rules as the result of reasoning. Example 1 shows that returning defaults as the result of reasoning can be beneficial by making answers to queries more informative.

In order to be able to generate answers in the form of defaults, a mechanism is needed to create new defaults based on the current knowledge base. Such rules must have the property that when they are added to the default theory, the theory does not change with respect to the results of reasoning. In other words, the set of extensions of the default theory $ext(\Delta)$ must remain unchanged.

Definition 3. *A* **default transformation** $t : \Delta \rightarrow \mathcal{D}$ *produces a new default* δ *from a default theory* $\Delta = \langle D, W \rangle$ *and is denoted by* $\Delta \mid\sim \delta$.

We define a set of transformations which have very useful features and will be used in the process of default reasoning. A general form of a transformation is $\langle D_t, f_t \rangle \mid\sim \delta$, where $D_t \subseteq D$, $W \models f_t$, and δ is a new concluded default.

Definition 4. *Given well-formed formulae a, b, c, d, e, we define the following transformations:*

a). Prerequisite substitution: $\left\langle \{\frac{a:b\wedge c}{b}\}, d \rightarrow a \right\rangle \quad \mid\sim \frac{d:b\wedge c}{b}$

b). Consequent substitution: $\left\langle \{\frac{a:b\wedge c}{b}\}, b \rightarrow e \right\rangle \quad \mid\sim \frac{a:b\wedge c\wedge e}{e}$

c). Justification reduction: $\left\langle \{\frac{a:b\wedge c\wedge d}{b}\}, d \right\rangle \quad \mid\sim \frac{a:b\wedge c}{b}$

d). Default transitivity: $\left\langle \{\frac{a:b\wedge c}{b}, \frac{b:e\wedge f}{e}\}, \top \right\rangle \mid\sim \frac{a:b\wedge c\wedge e\wedge f}{b\wedge e}$

The set of transformations (a)–(d) will be called *basic transformations*. These transformations can be further used in the communication process. Therem 1 shows the interesting property of these transformations.

Let us define the sequence of default transformations, denoted by $\mid\sim_*$, as follows. Let $D_0, \ldots, D_n$ is a sequence such that $D_0 = D$ and $D_i = D_{i-1} \cup \{\delta_i\}$ where δ_i is obtained by applying a basic transformation on $\langle D_{i-1}, W\rangle$. We write $\langle D, W\rangle \mid\sim_* \delta$ when $\langle D_n, W\rangle \mid\sim \delta$.

Theorem 1. *Given $\Delta = \langle D, W\rangle$ and $\Delta' = \langle D', W\rangle$ where $\forall_{\delta\in D'}\delta \in D \vee D \mid\sim_* \delta$ we have $ext(\Delta) = ext(\Delta')$*

The theorem shows that using the defined basic transformations we can create new defaults, which can be treated as valid rules for default reasoning. Moreover, these newly formed defaults can be treated as intermediate results of inference. For a full proof of the theorem see [17].

4.3 Reasoning with Default Transformations

In a multi-agent system the peers exchange knowledge by means of querying each other and utilising the answers to reach conclusions. Following the inference procedure for Distributed Description Logic proposed in [16], the query, which is passed between ontologies is the subsumption query in the form $A \sqsubseteq B$, which in FOL is denoted as $A(x) \rightarrow B(x)$. Here, we will concentrate on this type of query and we will denote it by writing $?A \sqsubseteq B$ to distinguish it from a DL statement.

For a query $?A \sqsubseteq B$ to a default theory $\Delta = \langle D, W\rangle$ we will presume there are three possible answers:

- *true* if $W \models A \sqsubseteq B$
- *false* if $W \not\models A \sqsubseteq B$,
- *true by default* if the default $\frac{A\,:\,B\sqcap J}{B}$ can be generated using the default transformations

The first two answers are strict and do not require further processing. The last answer can be treated as a partial result and the final answer can be inferred when the justifications are checked.

Algorithm 1. query

```
Input: Theory Δ = ⟨D, W⟩, Query ?A ⊑ B
begin
1   if W ⊨ A ⊑ B then return true;
2   E ← findExtensions(⟨D, W⟩);
    result ← ∅;
3   foreach E ∈ E do
        if A ⊑ B is consistent with E then
            D̄ ← findDefaults(GD(Δ, E), W, ?A ⊑ B);
            result ← result ∪ D̄;
    if result = ∅ then return false;
    result' ← ∅;
4   foreach δ ∈ result do
        δ' ← reduceJustification(δ, W);
        result' ← result' ∪ {δ'};
    return result';
end
```

Throughout the algorithm there are are references to a DL reasoning procedure in the form $W \models A \sqsubseteq B$. These steps can be treated as calls to an inference procedure for Description Logics such as the tableau reasoning algorithm [3].

Algorithm 1 shows the main idea of answering a query such as proposed above. Line 1 checks for a trivial answer based on the factual knowledge. If such an answer cannot be given, the next step is to find all extensions of the default theory (Line 2). This is done using an algorithm such as described in [4]. Iterating over all extensions (Line 3), the procedure gathers defaults in the form $\frac{A\,:\,B \sqcap J}{B}$, possibly from different extensions. This is done by transforming the generating defaults of each extension. Finally the resulting defaults are processed, applying the *reduce justifications* transformation (Line 4).

The procedure of finding defaults that can be treated as intermediate answers to the given query is expressed in Algorithm 2. This procedure applies default transformations (a), (b) and (c) from Definition 4. Line 1 selects the defaults that are qualified for applying the *prerequisite substitution* transformation. Then, each of the selected defaults is checked whether it can be returned as the default answer to the given query (Line 2). If this is not the case, the algorithm is executed recursively (Line 3) to find a sequence of defaults that having applied additionally the *default transitivity* transformation will produce an appropriate default form. Line 4 merges the sequenced defaults to generate the final result.

Algorithm 3 shows the application of the *justification reduction* default transformation. The default's justifications are confronted with the known facts from the knowledge base and if any of them proves to be true in W, then it is removed from the default (Line 1).

In effect the query algorithm generates one of three possible answers, which can be *true*, *false* or a set of defaults which are in the form $\frac{A\,:\,B \sqcap J}{B}$.

Algorithm 2. findDefaults

Input: Defaults $\hat{D}$, Facts W, Query $?A \sqsubseteq B$
begin
1 $D_0 \leftarrow \{\delta \in \hat{D} \mid W \models A \sqsubseteq Pre(\delta)$;
result $\leftarrow \emptyset$;
foreach $\delta \in D_0$ **do**
2 **if** $W \models Con(\delta) \sqsubseteq B$ **then** result $\leftarrow$ result $\cup \{\delta\}$;
else
3 $\bar{D} =$ `findDefaults`$(\hat{D} \setminus \{\delta\}, W \cup \{Con(\delta)\}, ?Con(\delta) \sqsubseteq B)$;
foreach $\bar{\delta} \in \bar{D}$ **do**
4 $\delta' \leftarrow \frac{Pre(\delta) \ : \ Jus(\delta) \wedge Jus(\bar{\delta})}{Con(\delta)}$;
result $\leftarrow$ result $\cup \{\delta'\}$;
return result;
end

Algorithm 3. reduceJustification

Input: Default δ, Facts W
begin
Assume $Jus(\delta) = \beta_1 \wedge \ldots \wedge \beta_n$;
1 $J \leftarrow \{\beta_i \mid W \not\models \beta_i\}$;
$\beta \leftarrow \bigwedge_{\beta \in J} \beta$
return $\frac{Pre(\delta) \ : \ \beta}{Con(\delta)}$;
end

5 Conclusion

One of the main reasoning tasks in description logics is subsumption checking, expressed as $A \sqsubseteq B$. It has been previously shown that default logic can be embedded into the description logic languages. Having default rules in the knowledge base, it would be beneficial to achieve answers to subsumption queries that would retain the information about the assumptions made during default reasoning.

To address this problem, we have presented the formalism of default transformations which can be used to derive answers to a default theory queries in the form of defaults. The proposed transformations generate new defaults in a default theory preserving the inferences that can be made unchanged. Such new defaults can then be treated as intermediate results in the reasoning process.

Default transformations can have an application to answering queries in a multi-agent system. Passing messages between agents in the form of defaults is more informative than strict answers, as the assumptions made during reasoning are not hidden from the querying agent, which in turn can itself validate the justifications to perform the inference locally.

An algorithm based on Distributed Description Logic is being developed for reasoning in a multi-agent environment. The results presented in this paper will be used to provide the means of embedding defaults into distributed knowledge.

Acknowledgements. The research was supported by the Polish National Budget Funds 2009-2011 for science under the grant N N516 3757 36.

References

1. Adjiman, P., Chatalic, P., Goasdoué, F., Rousset, M.-C., Simon, L.: Distributed reasoning in a peer-to-peer setting: application to the semantic web. J. Artif. Int. Res. 25(1), 269–314 (2006)
2. Amir, E., McIlraith, S.: Partition-based logical reasoning for first-order and propositional theories. Artif. Intell. 162(1-2), 49–88 (2005)
3. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The description logic handbook: theory, implementation, and applications. Cambridge University Press, New York (2003)
4. Baader, F., Hollunder, B.: Embedding defaults into terminological knowledge representation formalisms. Technical Report RR-93-20, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (1993)
5. Bao, J., Slutzki, G., Honavar, V.: A semantic importing approach to knowledge reuse from multiple ontologies. In: AAAI, pp. 1304–1309 (2007)
6. Borgida, A., Serafini, L.: Distributed description logics: Assimilating information from peer sources (2003)
7. Cholewinski, P.: Reasoning with stratified default theories. In: Marek, V.W., Truszczyński, M., Nerode, A. (eds.) LPNMR 1995. LNCS, vol. 928, pp. 273–286. Springer, Heidelberg (1995)
8. Cuenca Grau, B., Parsia, B., Sirin, E.: Combining OWL ontologies using $\mathcal{E}$-connections. J. Web Semantics 4(1), 40–59 (2006)
9. Dignum, V.: Using agent societies to support knowledge sharing. In: AAMAS 2003 Workshop on Autonomy, Delegation and Control (2003)
10. Kaleta, M., Palka, P., Toczylowski, E., Traczyk, T.: Electronic trading on electricity markets within a multi-agent framework. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 788–799. Springer, Heidelberg (2009)
11. Kowalski, R., Sadri, F.: From logic programming towards multi-agent systems. Annals of Mathematics and Artificial Intelligence 25(3-4), 391–419 (1999)
12. Reiter, R.: A logic for default reasoning. Artif. Intell. 13, 81–132 (1980)
13. Reiter, R., Criscuolo, G.: On interacting defaults, pp. 94–100 (1987)
14. Ryżko, D., Rybinski, H., Więch, P.: Learning mechanism for distributed default logic based mas - implementation considerations. In: Proceedings of the International IIS 2008 Conference, pp. 329–338 (2008)
15. Ryzko, D., Rybinski, H.: Distributed default logic for multi-agent system. In: IAT 2006: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Washington, DC, USA, pp. 204–210 (2006)
16. Serafini, L., Tamilin, A.: Local tableaux for reasoning in distributed description logics. In: Description Logics (2004)
17. Więch, P., Rybiński, H.: Using default transformations for reasoning in MAS. Technical report, ICS, Warsaw University of Technology (2010)

A Platform for the Evaluation of Automated Argumentation Strategies

Piotr S. Kośmicki

Institute of Computer Science, Faculty of Electronics and Information Technology,
Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
pedrok@wp.pl
http://home.elka.pw.edu.pl/~pkosmick/

Abstract. This paper describes a platform for testing automated argumentation strategies for agents. It is a continuation of the discussion about the Arguing Agents Competition (AAC). The second version of the AAC platform is introduced, including the architecture and the capabilities of the platform, the currently available engine and an automated strategy for the dialogue game.

The argumentation and some of the formalizations for the arguments and dialogues are briefly presented.

1 Introduction

Since the begining of the logic and rhetoric, the ways to convince the other participants of the dialogue were in much interest. So they are today. Computer science introduction to argumentation theory provides benefits to both non-technical theorists and software engineers. A computer simulation of human dialogue (e.g. [15]) could be considered a meeting point, however, there are more benefits for both parties. Traditional argumentation theory has gained software tools that assist in argument analysis (notably diagramming tools [8,12]). Argumentation turned out to be suitable for the modelling of the communication in multi-agent systems, application domains include: legal disputes, business negotiation, labor disputes, team formation, scientific inquiry, deliberative democracy, ontology reconciliation, risk analysis, scheduling, and logistics.

The lack of a common facility for the evaluation of the performance of agents using argumentation was one of the reasons for the commencement of Arguing Agents Competition (AAC). This paper is an attempt to continue the discussion on the AAC and, specifically, on suitability of the platform for testing of automated argumentation strategies.

Paper is organized as follows. Section 2 briefly presents general argumentation issues. Formal models of arguments are introduced in section 3 and formal models of dialogue in section 4. AAC initative and the platform are presented in section 5. Section 6 contains a discussion of the argumentation strategy testing issues. A dialogue game and an automated strategy, currently available for the platform, are presented in section 7. The final section presents conclusions for the future work on the platform.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 494–503, 2010.

2 Arguments and Dialogues

The minimal definition of an argument [8] is that it consists of a set of sentences (propositions) which is divided into a conlusion and premises, and an inference method from the premises to the conclusion. Argumentation inference is based on the content of statements and it is the main difference between the traditional approach based on the deductive logic. An argument may be supported or attacked by other arguments as well as some critical questions may be raised.

Argumentation main tasks are: identification of arguments, analysis of their structure, evaluation of their importance and invention of new relevant arguments. Moreover, they are performed in order to satisfy needs of one or more parties involved in a monologue or a dialogue. The needs may be limited to an investigation of a statement or include communication and pragmatical aims (e.g. to persuade, to reach an agreement).

Studies of good arguments lead recently to an identification of a number of argumentation schemes. Argumentation schemes are abstract argument forms commonly used by people in daily conversations as well as in more restricted situations (e.g. legal); [16] presents how argumentation schemes could be used within formal dialogues. Commonly used schemes are: argument from expert opinion, argument from cause to effect or *ad hominem* argument. For each scheme there is an argument (some premises and a conclusion) and a set of critical questions which should be typically posed.

Argumentation studies efforts were often focused on analysing bad arguments, called fallacies. Many of them are recognized and named. Some of them are presented in [13]. A particular feature of (informal) fallacies is that they are not fixed argument constructions (as it would be the case for invalid arguments in deductive reasoning). A notion of fallacy is based on giving (intentionally or not) unfair arguments which spoil discussion — the same arguments may be valid or not, depending on the context.

Argumentation may appear within a dialogue. This brings the importance of proper interaction between engaged parties, treatment of an audience, basis for the agreement (how it emerges from partial commitments) and reasons for the disagreement. Studies of natural dialogues resulted in the identification of a number of dialogue types and a commonly cited set (e.g. [8,10,13,7]) that include: information-seeking dialogues (participants exchange some information), inquiry dialogues (participants generate some new knowledge), persuasion dialogues (participants resolve a conflict of opinions), negotiation dialogues (participants search for a deal), deliberation dialogues (participants decide about a course of actions) and eristic dialogues (participants fight verbally). This categorisation analyses the information the participants have at the begining of a dialogue, their individual and shared goals for the dialogue. Among basic types, persuasion dialogue is important because a conflict of opinions may arise within any of the other types of dialogues.

During a dialogue, a mixture of its types may occure. Each transition is called a dialectical shift. Subdialogues are called embedded dialogues. The reasons for the occurence of dialectical shifts are miscellaneous, e.g. if during a persuasion

dialogue there is a need for more information, an information-seeking or an inquiry dialogue is launched.

The structure of a dialogue is specific depending on its type. In general, there are three stages: an opening stage when participants determine rules for the dialogue and messages, an argumentation stage and a closing stage when the results are determined. More specific proposition for persuasion dialogue consists of four stages: the confrontation stage, the opening stage, the argumentation stage and the concluding stage; [7]. Moreover, the structure includes the number of possible participants and their roles. For example, for a two person persuasion dialogue common role names are: a proponent (a protagonist of an expressed opinion) and an opponent (an antagonist of the opinion).

An organized course of a dialogue may be maintained when opinions of participants are made explicit. These opinions are called commitments. As dialogue progresses, participants have to take on commitments, respect them and respond to the interlocutor's objections. In some games, it is possible to withdraw from a commitment. Commitments are not only created by participant's own utterances, but also by the responses to the interlocutor's utterances.

3 Formalisations of Argument Models

Natural argumentation inspirations led to more formal approaches, suitable for the computation, which, in turn, allowed the usage of the argumentation in computer systems. A simple and widely used is Dung's Argumentation Framework [4]. An example of a more complicated model is The Carneades Argumentation Framework [5].

Dung's Argumentation Framework (also called *abstract argument system*) is based on the notion of an *argumentation framework* which is a pair $\langle \mathcal{A}, \mathcal{R} \rangle$, where $\mathcal{A}$ is a set of arguments and $\mathcal{R}$ is a binary relation over $\mathcal{A}$ called an attack relation (i.e. $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$). A natural interpretation of an argumentation framework is an argument graph. "Abstrat" means that arguments are analysed neglecting arg structure and the sense of attack relation (which could be based on the syntax or also on the semantics of arguments). In addition, there is no guidance on the construction arguments. While this argumentation framework is simple, it turned out that this model is flexible enough to analyse "properties which are independent of any specific aspect, and, as such, are relevant to any context that can be captured by this very terse formalization of abstract argument systems" (from [9]). The analysis is based on the studies of the acceptability of arguments — what are the rules for the evaluation of an argument status? The formal definition of such a method is called an argumentation semantics. Argumentation semantics corresponds, in general, to two approaches: skeptical and credulous acceptance of arguments. The choice of semantics come from the modelled world or from the required number of accepted arguments. Paper [9] provides a comprehensive review of the semantics identified in the literature.

The Carneades Argumentation Framework is much more sophisticated and was developed especially for legal applications. The framework is based on the

structured arguments (premises and conclusion are statements in an external language; there are three types of premises: ordinary, presumptions and exceptions), explicit dialectical status of statements (undisputed, at issue, accepted or rejected) and it includes the concept of proof standards (indicating the required level of an argument's justification). Thus it allows the burden of proof to be allocated on the proponent or respondent, as appropriate.

4 Formalisations of Dialogue Models

Similarly to argumentation formalisms, some formal approaches have been undertaken to specify course of dialogues. The aim of formalizations is to model a selected dialogue but also to propose rules which makes a dialogue coherent. Coherency of a dialogue means that every utterance furthers the goal of the dialogue. It is a game theoretic approach to argumentation where dialogues can be seen as instances of some dialogue games. "Winning" such a game means that a participant (a player), by making some moves (giving messages, speech acts), has defended the initial point of view, persuaded an interlocutor, reached an acceptable deal, etc. One of the models proposed in the literature is the Prakken's model of dialogue, named the *dialogue system* [6,11]. It is a flexible framework for the specification of dialogue games and there is a specialization for persuasion dialogues.

A *dialogue system* is a mathematical formulation of necessary dialogue elements. It consists of a *topic language* for the content and a *communication language* defininig speech act types. A *goal* states the purpose for a dialogue. There are at least two *participants* who have their *roles*. A dialogue takes place within a *context* of a shared, fixed knowledge. There are three groups of rules: a *protocol* which specifies allowed moves at each stage of a dialogue, in particular, regulating turntaking and termination; *effect rules* which specifies the effects of speech act on commitments of the participants; and *outcome rules* which define the result of a given dialogue.

5 AAC: History and Software Architecture

AAC vision and considerations for the system architecture have been presented to broader audience in [1,2]. The project draws experiences from *Argumento* ("a computer game for abstract argumentation", [3]). The first version of the software was AAC version '08 and this paper describes the second version, '09. The inspiration for the founders was Trading Agents Competition, TAC.

The AAC initiative goal is to develop a distributed on-line competition between heterogenous agents in which they can compete using various argument and dialogue protocols, where the moves and the arguments can be evaluated through a variety of argument computation engines. Those interested in the competition should prepare automated agents that would argue with each other, according to a dialogue game protocol chosen for a specific competition. There should be a correlation between an agent's abilities and the competition results,

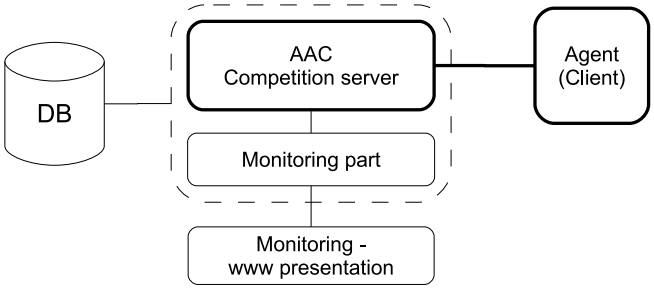


Fig. 1. Main components of the AAC system. Thick lines indicate components involved in agents communication.

so that agent with the best strategy will "win". An important aspect of the AAC is to gather data for the analysis of games played.

Main components of the version '09 of the AAC system are presented in the figure 1. There is a central competition server which is responsible for matching agents accordingly to a competition scenario, and for hosting ongoing dialogues. New dialogue game engines may be easily added. Agents connect to the server using its Web Services. During a dialogue there are two types of messages exchenged: a speech act and a generic message. This two types allow imlementation of many dialogue games.

Logically separated from the competition server is the monitoring part which provides information presented by a web administration interface. The administration interface allows the server and competitions to be configured (one may register agents, upload argument sets, choose dialogue setup: dialogue game, agents, their roles, argument set, topic, additional parameters). All the persistent data is stored in the database, including the history of all played dialogues. Server-side design makes it simple to prepare a virtual machine for an easy deployment. The system is developed using the Java programming language but agent developer is not obliged to follow this choice.

There is an agent implementation for the default dialogue game which contains an automated strategy. The architecture of a client application is presented in the

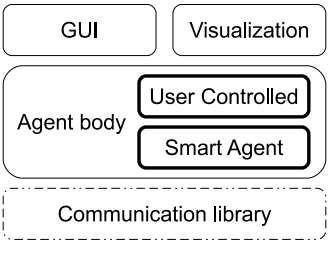


Fig. 2. Client application architecture. An agent is monitored by the GUI. Thick lines indicate exchengeable strategies. There is an automated strategy ("SmartAgent").

figure 2. A small library facilitates agent-server communication programming. Agent implementation may be controlled by different strategies during dialogues as well as it allows a user to take part in the dialogue. A client application has a graphical interface and also contains a visualization module adapted to the default dialogue game.

The mechanism which matches the agents for dialogues works in such a way that when the agents declare their readiness to engage in a dialogue, it chooses them according to the rules prepared for the competition. The rules may specify that any agents can be matched for a dialogue, or just only some or all of them.

Dialogue games have specifications which contain enumerations of dialogue roles for participating agents. It is possible to have dialogue games with more than two participants and with repeating roles. In a complex case, there could be some agents actively arguing and a number of other agents forming an audience.

6 AAC as a Flexible Platform for Argumentation Research

The core AAC platform is an attempt to develop a flexible tool for argumentation experiments. The requirement of the flexibility stems from the fact, that there are various models proposed in the literature for both the arguments (topic language) and the dialogue (communication language), therefore, we could also expect multiple variations for actual applications.

The last paper considering the AAC was [2]. The present state (version '09) of the project shows some progress regarding the issues discussed there. The platform may handle dialogue games with different number of participants and their roles. What is important, the new dialogue game offers a possibility to use the open world assumption. Moreover, the exploration request message introduces a simple information-seeking subdialogue. The platform is structured in such a way that another dialogue game engines may be easily inserted — it permits the use of more complex argumentation frameworks.

There is still a number of open issues though. First is the question if the comparison of agent performance could be based on more factors than win-loss rules. Second is the issue of measuring the time that a participant spends on "thinking" and taking into account the capabilities of remote machines. The current solution is still to consider all agents in an "open" category. Third is the adoption of the Argument Interchange Format (AIF) for storage and interchange of the arguments. AIF was designed to support a range of differing argumentation frameworks [14].

The current platform design has decoupled the competition (the data gathering) from the data analysis by recording all the messages exchanged between agents and the server. The underlying agent-server communication is simple and not bound to any dialogue game. The dialogue history is stored in the database and a separate tool should be developed for the analysis. There is no direct indication of how well an agent is performing. This approach allows different comparison methods to be used: from simple victory counting, through comparison of the performance of the same agents on the same argument set but with

switched roles (inspired by bridge teams tournaments) to more sophisticated processing.

7 Library of Dialogue Games and Argumentation Strategies

The argumentation testing platform, in order to become convinient, should include a number of dialogue game engines as well as a number of automated argumentation strategies for the comparison. The second version includes the Open World Dung dialogue game and one automated strategy.

The Open World Dung dialogue game rules are the rules from *Argumento* with some changes. In particular, the dialogue game has an option to use the open world assumption. In that case, agents (participants) discover the argument set assigned to the dialogue. Argumentation and dialogue rules are implemented as separated engines.

Argumentation rules are based on the Dung's Argumentation Framework. In the beginning, all arguments are supposed to not be undermined. Argument graph may not contain cycles. The argumentation proceeds by indicating arguments not yet used. Each interlocutor has its own commitment set containing chosen arguments. There is no possibility of withdrawal from the given argument. A valid choice can be made from arguments that do not attack nor get attacked by any other that belong to the commitment set. Should there arise an argument, the state of the arguments is changed to the undermined state and the reevalutation process of previously stated arguments is initiated. Each new argument has to change the evaluation of at least one of the already given (Grice's Maxim of Relation).

The set of initially known arguments is the set of all the arguments within a fixed distance (a dialogue parameter) from the topic argument. A participant may select an argument and discover a part of graph which is within a distance from the argument. The distance is calculated over both directions of the attack relation. If the argument is known to the participant but unknown to the interlocutor, such a argument is unveiled together with its attack relations with those arguments known by the intelocutor.

Dialogue rules are based on the Prakken's *dialogue system*. An example of a dialogue course is presented in the figure 3. A topic language is described by the argumentation rules. Each dialogue has a topic which is one of the initially known arguments. Communication language is formed by the following speech acts: *argument* — a regular speech act exchanged by the participants; *submission* — any of the participants may explicitely surrender; and *abstention* — having the current dialogue state favorable, a participant may wait for the next move of the interlocutor. Communication language has an extension, a request of exploration of the argument graph. Interchanging of the request message is known only to a sending participant and dialogue engine.

Dialogue purpose is to overcome the difference of opinions on a dialogue subject (the persuasive dialogue). The goal of each participant is to convince other

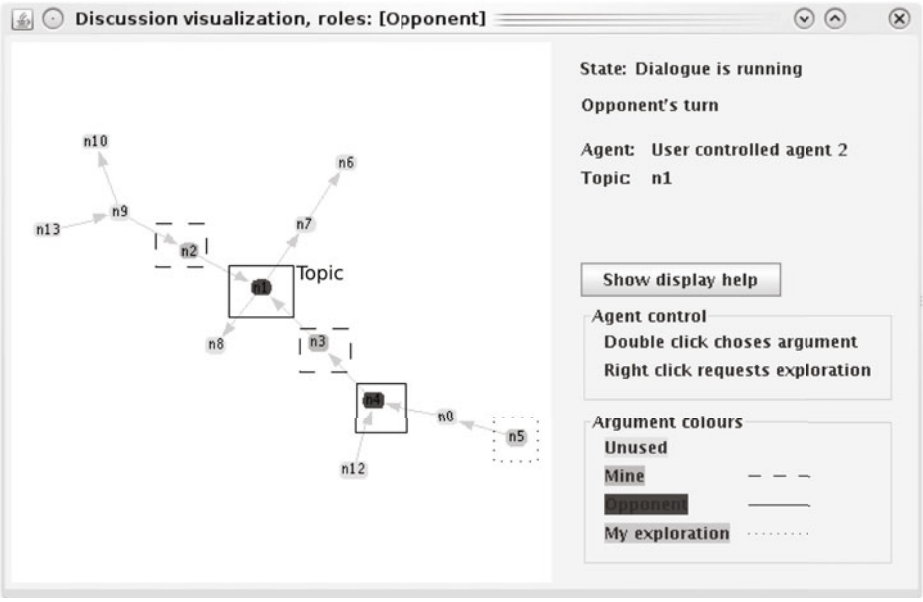


Fig. 3. Client application: visualization of a dialogue. Rectangles distinguish different types of nodes: dashed line for the agent's arguments, dotted line for its explored arguments and solid line for its interlocutor's arguments.

interlocutors. There are two participants (while the platform allows more). One has the role of a proponent and the other one the role of an opponent. At the beginning of each dialogue the participants receive a common set of initially known arguments. The set of arguments is the dialogue context.

Turn taking rules. The dialogue is begun by a topic argument on the behalf of the proponent participant. A participant makes one move by turn and the turn is switched to the interlocutor. The first move is made by the opponent. A dialogue is terminated when: there are no permitted moves to be done for a participant in his turn; a participant has made a submission; a participant has made too many illegal moves (the number is a parameter).

Outcome rules. After a termination, the winner is the proponent if the topic argument is not undermined and the opponent if the topic argument is undermined. The participant who made the submission is the loser. In any vague situation, participant who loses first is the looser.

Automated argumentation strategy algorithm is a revison of the utility-based algorithm presented in [3]. It is extended for the argument graph exploration. The algorithm operates on the local copy of the argument graph used in the dialogue. Every argument is marked whether as undermined or not.

The algorithm begins by checking if the agent is in a safe position and may wait for the interlocutor's move by sending the *abstention* speech act. Safe position for the proponent takes place when the topic argument is not undermined and conversely for the opponent.

In the questionable position, possible arguments are computed: arguments which are not yet used in the dialogue, do not attack any of the agent commitments and attack at least one of the interlocutor's commitments. If a possible argument does not have attackers, it is immediately selected, otherwise an argument with the best utility is selected.

Utility is computed as follows. Utility equals to 1 is the best choice and 0 is the worst choice. For each possible argument, an argument tree is analysed. An argument tree contains all possible argument chains (in inverse direction of attack relation). Nodes on the odd levels represent arguments that the agent may use and, on the even levels, the interlocutor. Only arguments which could be used by each participant are taken into account. Utility for leaves of a tree is equal to the reminder from a division of the depth by 2. Utility for nodes is the sum of the utility of attacking arguments divided by their number.

If there are no possible arguments, the algorithm checks if there are promising arguments to become exploration seeds. A seed is an argument around which arguments will be revealed. The seed with maximal potential (at the border of the known subgraph) is chosen and an exploration is requested. The agent waits for the game engine response and then the algorithm is relaunched. If there are no possible seeds, algorithm surrenders the agent.

8 Conclusions

This paper briefly presented the argumentation and some of the formalizations for the arguments and dialogues, as well as the features of the second version of the platform for testing automated argumentation strategies for agents. The development of the platform has been commenced within the AAC initiative.

The improved testing system is more flexible and generic. It allows the use of different dialogue game engines and the execution of a whole competition for the agents. Such competition may test agents' performance in many dialogues with different argument sets and parameters. The system contains a dialogue game engine and a default agent with a client application visualizing the agent behaviour in a dialogue.

There is a necessity to elaborate more work in the future regarding this subject. A dialogue analysis tool is required. Dialogue courses are recorded in a database and evaluation methods should be applied to verify agents' performance. More interesting work may concern the development of new dialogue game engines which will make the platform become a more complete environment for testing argumentation strategies.

References

1. Yuan, T., Schulze, J., Devereux, J., Reed, C.: Towards an Arguing Agents Competition: Building on Argumento. In: CMNA (2008)
2. Wells, S., Łoziński, P., Phan, M.N.: Towards An Arguing Agents Competition: Architectural Considerations. In: CMNA (2008)

3. Yuan, T., Svansson, V., Moore, D., Grierson, A.: A Computer Game for Abstract Argumentation. In: Proceedings of IJCAI 2007 Workshop on Computational Models of Natural Argument, Hyderabad, India, pp. 62–68 (2007)
4. Dung, P.M.: On the Acceptability of Arguments and Its Fundamental Role In Nonmonotonic Reasoning, Logic Programming and N-Person Games. Artificial Intelligence 77, 321–357 (1995)
5. Gordon, T.F., Walton, D.: The Carneades Argumentation Framework. Using Presumptions and Exceptions to Model Critical Questions. In: Proceedings of COMMA 2006, Computational Models of Argument. IOS Press, Amsterdam (2006)
6. Prakken, H.: Coherence and Flexibility in Dialogue games for Argumentation. Journal of Logic and Computation 15(6) (2005)
7. Norman, T.J., Carbogim, D.V., Krabbe, E.C.W., Walton, D.: Argument and Multi-Agent Systems. In: Reed, C., Norman, T.J. (eds.) Argumentation Machines. New Frontiers in Argument and Computation. Kluwer Academic Publishers, Dordrecht (2004)
8. Walton, D.: Argumentation Theory: A Very Short Introduction. In: Rahwan, I., Simari, G.R. (eds.) Argumentation in Artificial Intelligence. Springer, Heidelberg (2009)
9. Baroni, P., Giacomin, M.: Semantics of Abstract Argument Systems. In: Rahwan, I., Simari, G.R. (eds.) Argumentation in Artificial Intelligence. Springer, Heidelberg (2009)
10. McBurney, P., Parsons, S.: Dialogue Games for Agent Argumentation. In: Rahwan, I., Simari, G.R. (eds.) Argumentation in Artificial Intelligence. Springer, Heidelberg (2009)
11. Prakken, H.: Models of Persuasion Dialogue. In: Rahwan, I., Simari, G.R. (eds.) Argumentation in Artificial Intelligence. Springer, Heidelberg (2009)
12. Dębowska, K., Łoziński, P., Reed, C.: Building Bridges Between Everyday Argument and Formal Representation of Reasoning. Studies in Logic, Grammar and Rhetoric 16(29) (2009)
13. Walton, D.: Informal Logic. A Handbook for Critical Argumentation. Cambridge University Press, Cambridge (2005)
14. Chesñevar, C., McGinnis, J., Modgil, S., Rahwan, I., Reed, C., Simari, G., South, M., Vreeswijk, G., Willmott, S.: Towards an Argument Interchange Format. The Knowledge Engineering Review, 1–25 (2007)
15. Dessalles, J.-L.: A Computational Model of Argumentation in Everyday Conversation: A Problem-Centered Approach. In: Besnard, P., Doutre, S., Hunter, A. (eds.) Proceedings of COMMA 2008, Computational Models of Argument. IOS Press, Amsterdam (2008)
16. Reed, C., Walton, D.: Argumentation schemes in dialogue. In: Hansen, H.V., et al. (eds.) Dissensus and the Search for Common Ground, CD-ROM, pp. 1–11. OSSA, Windsor (2007)

Fuzzy Similarity-Based Relative Importance of MPEG-7 Visual Descriptors for Emotional Classification of Images

EunJong Park[1], SungHwan Jeong[2], and JoonWhoan Lee[2]

[1] Electronics and Telecommunications Research Institute, Daejeon, South Korea
[2] Computer Engineering Department, Chonbuk National University, JeonJu, South Korea
for511@etri.re.kr, {chlee,shjeong}@chonbuk.ac.kr

Abstract. Many kinds of attributes are used for various areas of decision making. Sometimes the attributes have complicated vector-types as in MPEG-7 visual descriptors that prevent us from attaching unequal importance to each descriptor for the construction of content- or emotion-based image retrievals. In this paper, fuzzy similarity-based rough approximation is used for determining the relative importance of MPEG-7 visual descriptors for an emotion. In the methods, the relative importance is given to a descriptor itself rather than a component of the vector of a descriptor or a combined descriptor. Also we propose a method for building a classification system based on representative color images. The experimental result shows the proposed classification method is promising for the emotional classification or evaluation of color images.

Keywords: Fuzzy similarity-based rough set; Fuzzy similarity-based classification; Weight Decision of Attributes; Emotion classification of Images.

1 Introduction

Many kinds of attributes are used for various areas of decision making. Some are scalar-valued and others are vector-valued. Some have nominal values and others have ordinal values. Sometimes the attributes have complicated vector-types as in MPEG-7 visual descriptors. Every visual descriptor in MPEG-7 is represented as a vector with multiple components, and some descriptors such as DCD(Dominant Color Descriptor) do not have fixed dimension. [1][2]

This high and varying dimensionality prevents us from utilizing a combined descriptor and attaching unequal importance to each descriptor. The descriptor vectors cannot be simply concatenated to obtain a high dimensional descriptor, because they may have different dimensions. Even though one can combine huge and fixed dimensional vectors of descriptors, it is difficult to analyze or interpret which descriptor is more important than the others, because it is the components of the combined vectors that are meaningful rather than a descriptor.

Meanwhile, some MPEG-7 descriptors seem to be more important than others for the classification of images according to a specific emotion. In general, emotions are represented by pairs of adjective words with opposite meanings such as “warm-cool”, “heavy-light”, and “dynamic-static”. For example, color descriptors seem to be more

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 504–513, 2010.

important for the classification of warm or cool images, because a red or yellow colored image seems to be warmer than a blue colored one.

In this paper, fuzzy similarity-based rough approximation is used for determining the relative importance of MPEG-7 visual descriptors for an emotion. Also, similarity-based decision making is applied to emotion-based image classification. In the methods, more relative importance is given to a descriptor itself rather than a component of the vector of a descriptor or a combined descriptor. The experimental result shows the proposed classification method is promising for the emotional classification or evaluation of color images.

This paper is organized as follows. In section 2 the weight decision method based on fuzzy similarity-based rough set theory is reviewed with classification rules. In section 3 several components needed for emotional classification are described, including the adjective image scales, MPEG-7 visual descriptors and the training algorithms. The experimental results are shown and discussed in section 4 and the final conclusion is presented in section 5.

2 Fuzzy Similarity-Based Weight Decision and Classification

Classical rough set theory, proposed by Pawlak in 1982, is a mathematical tool to deal with inexact, uncertain or vague knowledge. It is based on upper and lower approximation defined on the indiscernibility relation.[3][4][5] Even though there have been numerous theoretical applications in many fields of artificial intelligence, classical rough set theory can only deal with discrete and symbolic attributes in a decision table[6]. This problem has been solved by the extended notion of similarity-based approximations.[4][5] The similarity relation expresses weaker forms of indiscernibility, which does not lead to equivalence relations.

2.1 Approximation of Fuzzy Similarity-Based Rough Set

Suppose $I = (U, A \cup \{d\})$ is a decision table, where U (Universe) is a finite non-empty set of objects; $A \cup \{d\}$ is the union of a set of condition attributes A and a decision attribute d. For $a \in A$, $f : U \to V_a$, where V_a is the value set of attribute a; and f is an information function. This means that a decision attribute assigns a classification label given by a decision-maker to an object in the universe. The cardinality of the decision $d(U) = \{k : d(x) = k \ for \ x \in U\}$ is called the rank of d and is denoted by $r(d)$. We assume that the set V_d of values of the decision d is equal to $\{1,..., r(d)\}$. Let us observe that the decision d determines the partition $CLASS_A(d) = \{X_1,..., X_{r(d)}\}$ of the universe U, where $X_k = \{x \in U : d(x) = k\}$ for $1 \le k \le r(d)$. Suppose the fuzzy similarity relation is given by R_B for $B \subseteq A$, which can be written as:

$$R_B = \{(x, y) \in U \times U : \mu_{R_B}(x, y)\},$$

where $R_B \in F(U \times U)$, $\mu_{R_B}(x, y)$ is the membership degree of (x, y), $\mu_{R_B}(x, y) \in [0,1]$. If R_B has following properties, then it is called a fuzzy similarity relation.

(1) Symmetry: $\mu_{R_B}(x,y) = \mu_{R_B}(y,x), \forall x, y \in U$,

(2) Reflexivity: $\mu_{R_B}(x,x) = 1, \forall x \in U$.

Therefore, given a non-empty set of finite objects U for any object $x \in U$, we can define the similarity classes on fuzzy similarity relation R_B denoted as $R_B^\lambda(x)$ under threshold λ. $R_B^\lambda(x)$ denotes the set of objects which are similar to object x on the extent of $\mu_{R_B}(x,y) \geq \lambda$:

$$R_B^\lambda(x) = \left\{ y \in U : \forall \mu_{R_B}(y,x) \geq \lambda, yR_B^\lambda x \right\}.$$

Let $X \subseteq U$, then the upper or lower approximation on $B \subseteq A$ in terms of λ based on the fuzzy similarity relation can be defined as:

$$\underline{R}_B^\lambda(X) = \left\{ x \in X : R_B^\lambda(x) \subseteq X \right\},$$
$$\overline{R}_B^\lambda(X) = \bigcup_{x \in X} R_B^\lambda(x).$$

2.2 Weight Decision of Attributes

Using the notion of lower approximation, the $R_B^\lambda(x)$-positive region of objects in set X is defined as $POS_{R_B^\lambda}(X) = \underline{R}_B^\lambda(X)$. Let $X_i = \left\{ x \in U : d(x) = i \right\}$. The set $POS\left(R_B^\lambda, \{d\}\right) = \bigcup_{i=1}^{r(d)} \underline{R}_B^\lambda(X_i)$ is called the R_B^λ- positive region of partition $\left\{ X_i : i = 1,2,...,r(d) \right\}$. The positive region is the union of the lower approximations of the decision classes, and includes only those objects which unambiguously belong to the corresponding decision classes.

A relative reduct in a fuzzy similarity-based rough set model is defined by the minimal number of attributes that does not reduce the positive region of a partition, i.e. a subset $T \subseteq A$ such that $POS\left(R_A^\lambda, \{d\}\right) = POS\left(R_T^\lambda, \{d\}\right)$. Stepaniuk proposed aggregation methods to define a similarity relation for a combined set of condition attributes.[4] In the paper we choose the minimum operator to aggregate the fuzzy similarity relations for a set of combined attributes. That is $\mu_{R_B}(x,y) = \min_{b \in B}\left(\mu_{R_b}(x,y)\right)$. In Stepaniuk's paper, he also proposed two weighting schemes for an attribute according to a fuzzy similarity relation, given as

$$SRC\left(R_A^\lambda, \{d\}, a\right) = \frac{card\left(POS\left(R_A^\lambda, \{d\}\right)\right) - card\left(POS\left(R_{A-\{a\}}^\lambda, \{d\}\right)\right)}{card(U)} \tag{1}$$

$$SGF\left(R_A^\lambda, \{d\}, a\right) = \frac{card\left(POS\left(R_A^\lambda, \{d\}\right)\right) - card\left(POS\left(R_{A-\{a\}}^\lambda, \{d\}\right)\right)}{card\left(POS\left(R_A^\lambda, \{d\}\right)\right)} \tag{2}$$

Thus, in both cases the significance of an attribute reflects the degree of decrease of the positive region as a result of removing attribute a from A. If $T \subseteq A$ is a relative reduct then for $a \in T$, $SRC\left(R_A^\lambda, \{d\}, a\right) > 0$ or $SGF\left(R_A^\lambda, \{d\}, a\right) > 0$. Note that each $SRC\left(R_A^\lambda, \{d\}, a\right)$ and $SGF\left(R_A^\lambda, \{d\}, a\right)$ is a function of the threshold λ and can be

treated as the relative importance of an attribute. An actual weight can be calculated by normalizing $SRC\left(R_A^{\lambda},\{d\},a\right)$ or $SGF\left(R_A^{\lambda},\{d\},a\right)$, that is

$$w_a^{\lambda}=\frac{SRC\left(R_A^{\lambda},\{d\},a\right)}{\sum_a SRC\left(R_A^{\lambda},\{d\},a\right)} \quad \text{or} \quad w_a^{\lambda}=\frac{SGF\left(R_A^{\lambda},\{d\},a\right)}{\sum_a SGF\left(R_A^{\lambda},\{d\},a\right)}.$$

2.3 Fuzzy Similarity-Based Classification

After the calculation of weights it is quite straightforward to perform classification. For a given object x_{new} to be classified we can calculate the total weighted similarity between the unclassified image and the saved representation image. Then we can use a hard decision rule in which the given object is classified into the class that contains the maximally similar object. Also, we can classify an image into an ambiguous class if the maximal similarity is less than a given threshold. In other words, the image that is $\max\{R_B(x_{new},y): y\in U\}\le Th$ can be assigned to "neutral" in the two-class emotional classification. Therefore, there are two kinds of rules for hard classification, as follows.

- Rule 1: Classify x_{new} as $d(x_{new})=d(x)$,
  where $R_B(x_{new},x)=\max\{R_B(x_{new},y): y\in U\}$
- Rule 2: Classify x_{new} as $d(x_{new})=d(x)$,
  where $R_B(x_{new},x)=\max\{R_B(x_{new},y): y\in U\}\ge Th$
  Otherwise it belongs to neutral class.

For the calculation of total weighted similarity, Stepaniuk also suggested several aggregation operators [4]. In the paper we choose the weighted average operator given as

$$R_B(x_{new},y)=\sum_{a\in B} w_b R_b(x_{new},y).$$

3 MPEG-7 Visual Descriptors for Emotion Classification of Images

For the emotional classification of images, we have to define several components. Those are the emotion space to categorize images, visual descriptors or attributes to be used for the classification, and the training/classification algorithms.

3.1 Emotion Space

In the paper we assumed the emotion space can be defined by three adjective pairs including "warm-cool", "dynamic-static", and "heavy-light". Note that each one consists of two adjectives of opposite meaning. Actually, numerous adjectives are used to represent a feeling evoked when someone sees a color image. Three pairs of adjectives are selected based on the analysis of 13 pairs of adjectives, using PCA(Principal Component Analysis) for the data collected from psychological experiments[7]. Fig. 1 shows the three-dimensional emotion space, where a color image is categorized according to the feeling evoked when someone sees the image.

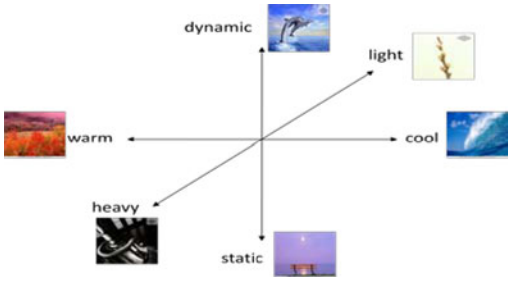


Fig. 1. 3-dimensional Emotion Space

3.2 Visual Descriptors

In order to categorize a color image depending on the emotion, it should be compared with respect to some descriptors or attributes. We propose to use MPEG-7 visual descriptors that were originally recommended for content-based image retrievals, because they are standardized and completely verified, and content-based image retrieval is closely related to our emotional classification scheme. The descriptors that we use in our method include 4 color-related ones: CLD(Color Layout Descriptor), SCD(Scalable Color Descriptor), CSD(Color Structure Descriptor) and DCD(Dominant Color Descriptor), and 2 texture-related ones: EHD(Edge Histogram Descriptor) and TBD(Texture Browsing Descriptor).

CLD is a DCT(Discrete Cosine Transform)-based descriptor that represents the spatial distribution of colors in an image with a 12-dimensional vector. CSD expresses the local color structure in an image with a structuring element. In the proposed emotional classification method we use a 32-dimensional CSD. A DCD describes the representative color distributions and features in an image or a region of interest. Depending on the number of dominant colors, different images can have different dimensional DCDs. A SCD is a color histogram in HSV color space, which is encoded by a Haar transform. Its binary representation is scalable in terms of bin numbers and bit representation accuracy. In the proposed method, we chose 64 bins and 4 bits, which results in 64-dimensional vector representation. EHD describes the edge distribution with a histogram based on the local edge distribution in an image. Because there are 16 sub-images to be considered in an image and 5 directional edges for each sub-image, it is an 80-dimensional vector. TBD is a 5-dimensional vector that relates to the perceptual characterization of texture, in terms of regularity, directionality and coarseness. It is useful for coarse classification of textures.

3.3 Training of the Classifier

Algorithm 1 shows the procedure of our training scheme, in which classification data is captured in representative images for each pair of emotions, and the weights to represent the relative importance of the MPEG-7 descriptors to determine a specific emotion. In the first step of the training phase, a group of human subjects evaluate training samples of images and select 5 representative images for each emotion represented by an adjective. Then there were 10 representative images for a pair of

emotions represented by two adjectives of mutually opposite meaning. In general, the number of representative images depends on the emotion, but it should be more than one. Even though descriptors that are extracted from different images are dissimilar, they can evoke the same emotion. Because human feelings evoked when we look at images are pretty complex, it is clear that the number should be more than one, even though we don't know what constitutes a sufficient number of representative images. In our scheme we chose 5 images for each emotion, for the sake of convenience.

In the analysis, the MPEG-7 visual descriptors mentioned above were extracted from the representative images, and the similarity with respect to each descriptor was calculated for each pair of images. Then a fuzzy similarity relation $R_b^e(x, y)$ for each descriptor b and each pair of emotions e was constructed by normalizing the similarities of pairs of images. A fuzzy similarity relation $R_b^e(x, y)$ can be defined as a matrix with a size of 10x10 for each pair of emotions and each descriptor, because there are 10 representative images for a pair of emotional adjectives. After construction of those similarity relations, the weights of the MPEG-7 descriptors showing the relative importance can be determined using the methods in Section 2.

Algorithm 1. Training Phase of Emotion Classifier

```
Select the representative images by human subjects
For each pair of emotions among "warm-cool", "dynamic-static" and "heavy-light"
    For each representative image
        Extract the 6 MPEG-7 visual descriptors
    End for
    For each MPEG-7 visual descriptor
        Construct the fuzzy similarity relation by pair-wise comparisons of images
    End for
    Determine the weights of the descriptors using the method in Section II
End for
```

3.4 Emotional Classification of Images

After the training is completed, the representative images and the weight of the descriptors are determined for each pair of emotions, as mentioned before. For an input image to be emotionally classified, the MPEG-7 visual descriptors are extracted and compared with those of the representative images, for each pair of emotions. Note that this comparison in terms of similarity measures of MPEG-7 descriptors is just a browsing process for content-based image retrieval.

In the paper, we propose rule 2 (Section 2), because it provides more flexibility supporting the collection of undecided or ambiguous images in the classification method, and can be used to perform additional processing. For example, additional processing can include the fact that human subjects can choose representative images for specific emotions among the undecided ones and retraining can accommodate new ones for refining the classifier. After the classification process is complete, we can obtain a 3-diminsional vector of labeled emotions that can be located in the emotion space. Algorithm 2 shows the classification process proposed in the paper.

Algorithm 2. Classification Phase of Color Image

```
For a given input image to be emotionally classified
Extract the 6 MPEG-7 visual descriptors
For each pair of emotions among "warm-cool", "dynamic-static" and "heavy-light"
    Find the most similar representative image using the weights of descriptors
    If the maximum similarity is larger that a threshold
        then classify the image to the corresponding emotion
    Otherwise, classify the image into "neutral"
    End for
End of classification
```

4 Experimental Results and Discussion

For the experiment, in order to evaluate the performance of the proposed emotional classification scheme we have constructed a small scale database of various kinds of color images including natural, sports, indoors, human body, interior and sculpture scenes. The 5 representative color images for each emotion, which was expressed with an adjective, have been selected from the database by graduate students. Then there are 10 representative color images for a pair of emotions represented by two adjectives of mutually opposite meaning. Because there are 3 pairs of emotions, as shown in Fig. 1, a total of 30 images have been used as representatives. Fig. 2 shows the representative color images depending on the emotions.

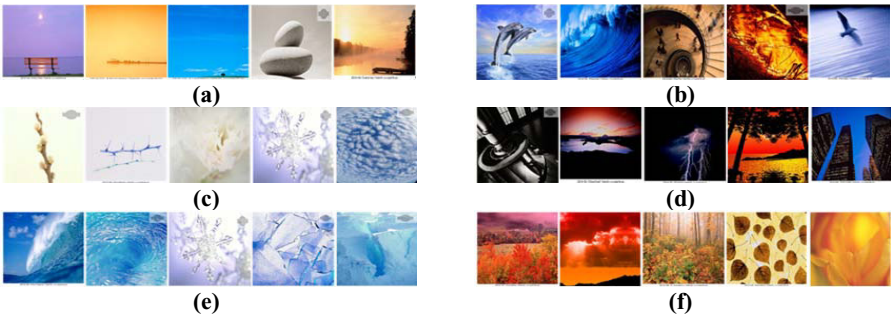


Fig. 2. Sets of Representative Color Images for Emotions. (a)Static, (b)Dynamic, (c)Light, (d)Heavy, (e)Cool, (f) Warm.

4.1 Weight Decision of Descriptors

The MPEG-7 visual descriptors have been extracted from the set of 10 representative color images for a given pair of emotions. Then the similarities have been calculated for each pair of images with respect to a descriptor, which can be represented as a 10x10 symmetric matrix, as shown in Fig. 3. Fig 3(a) and (b) show the similarity relations of representative images for a given pair of emotions "static-dynamic" with respect to EHD and CSD, respectively. In the figure, U represents the set of the images from 0 to 10, and d stands for the decision value of the emotion. Note that for each $x \in U$

$$d(x)=\begin{cases}0 & \text{when } x=1,2,3,4,5\\ 1 & \text{when } x=6,7,8,9,10\end{cases}.$$

The decision value of 0 and 1 means the emotion represented by "static" and "dynamic", respectively. The shaded elements of the matrix denote components for which the similarity between a pair of objects is larger than a given threshold (λ =0.55). Note that $POS\left(R_{EHD}^{0.55},\{d\}\right)=U$, but $POS\left(R_{CSD}^{0.55},\{d\}\right)=\varnothing$ as seen in Fig. 3, which means that EHD is an important attribute and should be included in a relative reduct, but CSD is meaningless and can be removed from the classification process.

From the analysis of the similarity relations for each emotion with respect to the MPEG-7 visual descriptors, we have obtained the weight of importance, as shown in Table 1. As we expect from the previous discussion, EHD is the most important attribute to evaluate a color image in terms of the emotion "Static-Dynamic". Also, the color descriptors are more important than the texture descriptors for emotional evaluation in terms of "Cool-Warm".

U	1	2	3	4	5	6	7	8	9	10	*d*
1	1	0.86	0.86	0.72	0.79	0.43	0.18	0.32	0.15	0.35	0
2	0.86	1	0.93	0.71	0.74	0.34	0.14	0.27	0.12	0.25	0
3	0.86	0.93	1	0.71	0.77	0.38	0.18	0.29	0.12	0.30	0
4	0.72	0.71	0.71	1	0.69	0.54	0.32	0.45	0.29	0.44	0
5	0.79	0.74	0.77	0.69	1	0.51	0.26	0.44	0.28	0.34	0
6	0.43	0.34	0.38	0.54	0.51	1	0.48	0.57	0.49	0.56	1
7	0.18	0.14	0.18	0.32	0.26	0.48	1	0.48	0.43	0.34	1
8	0.32	0.27	0.29	0.45	0.44	0.57	0.48	1	0.52	0.33	1
9	0.15	0.12	0.12	0.29	0.28	0.49	0.43	0.52	1	0.33	1
10	0.35	0.25	0.30	0.44	0.34	0.56	0.34	0.33	0.33	1	1

(a)

U	1	2	3	4	5	6	7	8	9	10	*d*
1	1	0.61	0.62	0.54	0.59	0.60	0.68	0.50	0.51	0.71	0
2	0.61	1	0.80	0.71	0.78	0.51	0.61	0.62	0.65	0.69	0
3	0.62	0.80	1	0.67	0.60	0.68	0.79	0.46	0.48	0.71	0
4	0.54	0.71	0.67	1	0.62	0.43	0.50	0.55	0.43	0.57	0
5	0.59	0.78	0.60	0.62	1	0.38	0.49	0.81	0.73	0.53	0
6	0.60	0.51	0.68	0.43	0.38	1	0.72	0.28	0.22	0.75	1
7	0.68	0.61	0.79	0.50	0.49	0.72	1	0.39	0.36	0.72	1
8	0.50	0.62	0.46	0.55	0.81	0.28	0.39	1	0.67	0.40	1
9	0.51	0.65	0.48	0.43	0.73	0.22	0.36	0.67	1	0.38	1
10	0.71	0.69	0.71	0.57	0.53	0.75	0.72	0.40	0.38	1	1

(b)

Fig. 3. Similarity Relations of Representative Images for "dynamic-static" with respect to (a) EHD and (b) CSD

Table 1. Resulting Weights of Importance with respect to Descriptors

	CLD	CSD	DCD	SCD	EHD	EHD	TBD
Static-Dynamic	0.11	0.00	0.26	0.05	0.53	0.53	0.05
Light-Heavy	0.07	0.36	0.36	0.18	0.03	0.03	0.00
Cool-Warm	0.30	0.14	0.14	0.30	0.09	0.09	0.03

4.2 Classification Results with Discussions

The proposed method of emotional classification of color images is evaluated in terms of the degree of accordance of classification the results between the system and human subject. For the performance evaluation we have selected 60 color images from the small database including. Those various color images have been classified by the proposed method in terms of three pairs of emotions, and each classification result of an image has been presented to a human subject to see whether he or she agrees with the results.

The user interface for emotional classification is shown in Fig. 4. A human subject can use it to express his or her opinion for the result of a decision made by the classification system. The different thresholds in classification rule 2 can be set through the lower-left window. The upper-middle widow shows the classification result of the color image presented in the upper-left window by the system. A human subject can express his or her opinion for the classification result of the system (either "valid" or "invalid") through the right-most window.

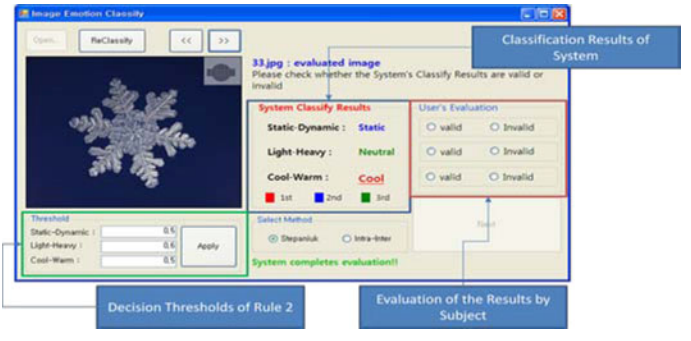


Fig. 4. User Interface for Performance Evaluation

10 male and female graduate students were involved in the experiment, and Table 2 shows the ratio of classification on average that agreed with the opinion of the human subjects. The number in parenthesis represents the standard deviation of the correct ratio.

The classification in terms of the "light-heavy" adjective achieves the best agreement with the opinion of the human subjects. But the classification in terms of the "static-dynamic" adjective achieves the worst agreement. This could be due to a psychological artifact that "static-dynamic" is a more complicated concept than "light-heavy", because it relies more on human experience than visual information. In other words "light-heavy" can be perceived better than "static-dynamic" only by visual information. We expect that more representative color images for "static-dynamic" will provide better results.

Table 2. Evaluation Result: Degree of accordance(Standard Dev.)

Static-Dynamic	*Light-Heavy*	*Cool-Warm*	*Total*
70.67%(6.45)	**86.50%(2.85)**	82.67%(4.35)	79.97%(7.37)

5 Conclusion

Many kinds of attributes are used for various areas of decision making. Sometimes the attributes have complicated vector-types as in MPEG-7 visual descriptors that prevent us from utilizing a combined descriptor and attaching unequal importance to each descriptor for the construction of content- or emotion-based image retrieval.

In this paper, fuzzy similarity-based rough approximation is used for determining the relative importance of MPEG-7 visual descriptors to determine an emotion. Also, a similarity-based decision is applied to emotion-based image classification. In the methods, more relative importance is given to a descriptor itself rather than a component of the vector of a descriptor or a combined descriptor. Also, we propose a method for building a classification system based on representative color images. The experimental result shows the proposed classification method is promising for the emotional classification or evaluation of color images.

Acknowledgments. This work was supported by the Postal Technology R&D program of Korea Post. [2006-X-001-02, Development of Real-time Postal Logistics System]

References

1. Yamada, A., Pickering, M., Jeannin, S., Cieplinski, L., Ohm, J.R., Kim, M.: MPEG-7 Visual part of experimentation Model Version 10.0. ISO/IEC JTC1/SC29/WG11/N4063, March 2001, pp. 20–33 (2001)
2. Cieplinski, L., Kim, M., Ohm, J.-R., Pickering, M., Yamada, A.: Text of ISO/IEC 15938-3/FCD Information technology Multimedia content description interface Part 3 Visual. ISO/IEC JTC1/SC29/WG11/N4062, pp. 30–53 (2001)
3. Park, E.J., Kim, S.Y., Lee, J.W.: Rough Set Based Interpretation of Color Emotion. In: 2007 Proceedings of the Korean Society for Emotion and Sensibility Conference, pp. 109–113 (2007)
4. Stepaniuk, J.: Approximation Spaces in Extensions of Rough Set Theory. In: Polkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS (LNAI), vol. 1424, pp. 290–297. Springer, Heidelberg (1998)
5. Jiang, Y.-j., Chen, J., Ruan, X.-y.: Fuzzy similarity-based rough set method for case-based reasoning and its application in tool selection. International Journal of Machine Tools & Manufacture 46, 107–113 (2006)
6. Pawlak, Z.: Rough sets: theoretical aspects of reasoning about data. Kluwer Academic Publishers, Dordrecht (1991)
7. Kim, S.W., Eum, K.B., Chung, S.S., Lee, J.W.: A Study on the Adjectives for Selection of Color Patterns. Korean Society for Emotion & Sensibility 8(4) (2005)

The Impact of Recommendation Sources on the Adoption Intention of Microblogging Based on Dominance-based Rough Set Approach

Yang-Chieh Chin[1], Chaio-Chen Chang[2], Chiun-Sin Lin[3], and Gwo-Hshiung Tzeng[4,5]

[1] Department of Management Science, National Chiao Tung University, 1001 Ta-Hsueh Road, Hsinchu 300, Taiwan
jerry110888@gmail.com
[2] Department of International Business, National Dong Hwa University, No.1, Sec. 2, Da Hsueh Rd., Shoufeng, Hualien, 97401, Taiwan
aka@mail.ndhu.edu.tw
[3] Department of Management Science, National Chiao Tung University, 1001 Ta-Hsueh Road, Hsinchu 300, Taiwan
netec7@yahoo.com.tw
[4] Department of Business Administration, Kainan University, No. 1 Kainan Road, Luchu, Taoyuan 338, Taiwan
ghtzeng@mail.knu.edu.tw
[5] Institute of Management of Technology, National Chiao Tung University, 1001 Ta-Hsueh Road, Hsinchu 300, Taiwan
ghtzeng@cc.nctu.edu.tw

Abstract. Microblogging is a social media tool that allows users to write short text messages to public and private networks. This research focuses specifically on the microblogging on Facebook. The main purposes of this study are to investigate and compare what recommendation sources influence the intention to use microbloggings and to combine gender, daily internet hour usage and past use experience to infer the usage of microbloggings decision rules using a dominance-based rough-set approach (DRSA). Data for this study were collected from 382 users and potential users. The analysis is grounded in the taxonomy of induction-related activities using DRSA to infer the usage of microbloggings decision rules. Finally, the study of the nature of microblogging reflects essential practical and academic value.

Keywords: Microblogging, Dominance-based Rough Set Approach (DRSA) Recommendation source, Adoption intention.

1 Introduction

Microblogging is a new communication channel with which people can share information. Microblogging platforms, primarily on social network sites such as Twitter and Facebook, have become popular. The concept of a social network is that two of your friends would have a greater probability of knowing each other than would two

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 514–523, 2010.

people chosen at random from the population [5]. Extensions of microblogging communications include status updates from social networks such as Facebook, and message-exchange services such as Twitter. User growth on Facebook, one of the biggest social networking sites in the world, is still expanding. Statistics from www.checkfacebook.com showed that Facebook's international audience totaled 350 million people at the beginning of 2010, including more than 5 million Taiwan users engaged in platform applications. A site that allows users to share daily updates through microblogging helps people to keep in touch [16], and businesses can increase sales as well by improving communications to and collaborations with customers [2]. With the growth of users on the microblogging services, the biggest benefit of microblogging is its ability to generate platform revenues by means of advertisements [28] and other applications. Thus, how to stimulate the microblogging adoption intention becomes a critical issue to platform marketers.

Even though microblogging offers conveniences and benefits, some people are concerned about the use of microblogging as another form of background check and that their privacy may be lost in cyberspace [27]. However, such concerns can be addressed by better and more accurate recommendations because people are influenced by others' recommendations when making decisions [17]. These recommendations can be classified as interpersonal sources, impersonal sources [1] and neutral sources [7]. Researchers have shed some light on the importance of recommendation sources in the context of product purchases [21], but little has been done on the relevance of these recommendation sources in the context of microblogging usage. Thus, our primary goal in this study is to fill that gap by increasing our understanding of how the three primary categories of recommendation sources—interpersonal recommendations (e.g., word-of-mouth recommendations), impersonal recommendations (e.g., advertising recommendations), and neutral recommendations (e.g., expert recommendations)—influence users intention to adopt microbloggings.

The classical rough set theory (RST) was proposed by Pawlak [23] as an effective mathematical approach for discovering hidden deterministic rules. However, the main restriction for the use RST is that the domain of attributes is preference ordered. To help fill the gap, Greco et al. [12] proposed an extension of the rough set theory based on the dominance principle to incorporate the ordinal nature of the preference data into the classification problem—what is called dominance-based rough set approach (DRSA). It substitutes the indiscernibility relation used in the classical rough set approach with a dominance relation that is reflexive and transitive [15]. DRSA derives a set of decision rules from preference-ordered data [30], which are then used in a classifier [4].

In addition, the DRSA approach was motivated by representing preference models for multiple criteria decision analysis (MCDA) problems, where preference orderings on domains of attributes are quite typical in exemplary based decision-making [18,19]. Therefore, another purpose of this study is to combine control variables (gender, daily internet hour usage, and past use experience), grounded in the taxonomy of induction-related activities using the DRSA, to infer the microblogging-related decision rules.

2 Literature Review

2.1 Microblogging

Microblogging systems provide a lightweight, easy form of communication that enables users to broadcast and share information about their current activities, thoughts, opinions and status. Compared to regular blogging, microblogging lowers the investment of the time and thought required to generate content and fulfills a need for a faster and more immediate mode of communication [17]. Microblogging, communication via short, real-time message broadcasts, is relatively a new communication channel for people to share information about their daily activities that they would not otherwise publish using other media (e.g., e-mail, phone, IM or weblogs). In a microblogging community, users can publish brief messages and tag them with keywords. Others may subscribe to these messages based on who publishes them or what they are about [15]. Popular microblogging platforms such as Facebook have risen to prominence in recent years.

2.2 Adoption Intention

Adoption is a widely researched process that is often used to investigate the spread of information technology [10,25,26]. According to the literature on information technology adoption, adoption intention is an individual's intention to use, acquire, or accept a technology innovation [26].

2.3 Recommendation Source

Prior studies have suggested that peer communications (such as families, friends, and colleges) may be considered the most trustworthy type of recommendation source in making decisions [24]. In addition, advertising recommendations, such as recommendations from site-sponsored advertisements, may be also regarded as a credibility cue [31]. Previous research has also demonstrated that the perceived level of expertise positively impacts acceptance of source recommendations [8]. These recommendations may be also considered a credibility cue when making decisions [31].

3 Basic Concepts of the Dominance-Based Rough Set Approach

3.1 Data Table

DRSA uses an ordered information table where in each row represents an object, which is defined a respondent to our survey, and each column represents an attribute, including preference-ordered domain and regular (no preference-ordered domain) [14]. Thus, the entries of the table are attribute values. Formally, an information system can be represented by the quadruple $IS = (U, Q, V, f)$, where U is a finite and non-empty set of objects (universe), $Q = \{a_1, a_2, ..., a_m\}$ is a non-empty finite set of ordered or non-ordered attributes, V_a is the domain of attribute a, $V = \bigcup_{a \in Q} V_a$, and

$f: U \times Q \to V$ is a total information function such that $f(x,a) \in V_a$ for every $a \in Q$ and $x \in U$. The set Q is usually divided into set C of ordered or non-ordered attributes and set D of decision attributes [12,13,29,30].

3.2 Approximation of the Dominance Relation

According to Greco et al. [13], first, let $\succeq_a$ be an outranking relation on U with respect to criterion $a \in Q$, such that $x \succeq_a y$ means "x is at least good as with respect to criterion a." Suppose that $\succeq_a$ is a complete preorder. Furthermore, let $\boldsymbol{Cl}=\{Cl_t, t \in T\}$, $T=\{1,2,...,n\}$, be a set of decision classes of U that each $x \in U$ belongs to one and only one class $Cl_t = \boldsymbol{Cl}$. Assume that, for all $r, s \in T$ such that $r \succ s$, the elements of Cl_r are preferred to the elements of Cl_s. Given the set of decision class $\boldsymbol{Cl}$, it is possible to define upward and downward unions of classes, respectively,

$$Cl_t^{\geq} = \bigcup_{s \geq t} Cl_s, \; Cl_t^{\leq} = \bigcup_{s \leq t} Cl_s, \; t = 1,2,...,n \tag{1}$$

In dominance-based approaches, we say that x dominates y with respect to $P \subseteq C$ if $x \succeq_a y$ for all $a \in P$. Given $P \subseteq C$ and $x \in U$, let $D_p^+(x) = \{y \in U : y \succeq x\}$ represent a set of objects dominating x, called a P-dominating set, and $D_p^-(x) = \{y \in U : x \succeq y\}$ represent a set of objects dominated by x, called a P-dominated set. We can adopt $D_p^+(x)$ and $D_p^-(x)$ to approximate a collection of upward and downward unions of decision classes.

The P-lower approximation of $\underline{P}(Cl_t^{\geq})$ of the unions of class $Cl_t^{\geq}$, $t \in \{2,3,...,n\}$, with respect to $P \subseteq C$ contains all objects x in the universe U, such that objects y that have at least the same evaluations for all the considered ordered attributes from P also belong to class Cl_t or better, as

$$\underline{P}(Cl_t^{\geq}) = \{x \in U : D_P^+(x) \subseteq Cl_t^{\geq}\} \tag{2}$$

Similarly, the P-upper approximation of $\overline{P}(Cl_t^{\geq})$ is composed of all objects x in the universe U, whose evaluations on the criteria from P are not worse than the evaluations of at least one object y belonging to class Cl_t or better, as

$$\overline{P}(Cl_t^{\geq}) = \{x \in U : D_P^-(x) \cap Cl_t^{\geq} \neq \varnothing\} \tag{3}$$

Analogously, the P-lower and P-upper approximations of $\underline{P}(Cl_t^{\leq})$ and $\underline{P}(Cl_t^{\leq})$, respectively, of the unions of class $Cl_t^{\geq}$, $t \in \{2,3,...,n\}$, with respect to $P \subseteq C$ are defined as

$$\underline{P}(Cl_t^{\leq}) = \{x \in U : D_P^-(x) \subseteq Cl_t^{\leq}\} \tag{4}$$

$$\overline{P}(Cl_t^{\leq}) = \{x \in U : D_P^+(x) \cap Cl_t^{\leq} \neq \varnothing\} \tag{5}$$

The *P*-boundaries (P-doubtable regions) of $Cl_t^{\geq}$ and $Cl_t^{\leq}$ are defined as

$$Bn_P(Cl_t^{\geq}) = \overline{P}(Cl_t^{\geq}) - \underline{P}(Cl_t^{\geq}) \tag{6}$$

$$Bn_P(Cl_t^{\leq}) = \overline{P}(Cl_t^{\leq}) - \underline{P}(Cl_t^{\leq}) \tag{7}$$

with each set $P \subseteq U$ we can estimate the accuracy of approximation of $Cl_t^{\geq}$ and $Cl_t^{\leq}$ using

$$\alpha_P(Cl_t^{\geq}) = \left| \frac{\underline{P}(Cl_t^{\geq})}{\overline{P}(Cl_t^{\geq})} \right| \quad \alpha_P(Cl_t^{\leq}) = \left| \frac{\underline{P}(Cl_t^{\leq})}{\overline{P}(Cl_t^{\leq})} \right| \tag{8}$$

and the ratio

$$\gamma_P(\boldsymbol{Cl}) = \left| \frac{U - (\bigcup_{t \in \{2,\ldots,n\}} Bn_P\left(Cl_t^{\geq}\right)}{U} \right| = \left| \frac{U - (\bigcup_{t \in \{1,\ldots,n-1\}} Bn_P\left(Cl_t^{\leq}\right)}{U} \right| \tag{9}$$

3.3 Extraction of Decision Rules

A decision rule can be expressed as a logical manner of the if (antecedent) then (consequence) type of decision. The procedure of capturing decision rules from a set of initial data is known as induction [23]. According to [13,30], for a given upward union of classes, $Cl_t^{\geq}$, the decision rules included under the hypothesis that all objects belonging to $\underline{P}(Cl_t^{\geq})$ are positive and the others are negative. There are two types of decision rules as follows:

(1) $D_{\geq}$ decision rules ("at least" decision rules)

If $f(x,a_1) \geq r_{a_1}$ and $f(x,a_2) \geq r_{a_2}$ and … $f(x,a_P) \geq r_{a_P}$, then $x \in Cl_t^{\geq}$

(2) $D_{\leq}$ decision rules ("at most" decision rules)

If $f(x,a_1) \leq r_{a_1}$ and $f(x,a_2) \leq r_{a_2}$ and … $f(x,a_P) \leq r_{a_P}$, then $x \in Cl_t^{\leq}$

4 An Empirical Example of Microblogging

Microblogging appeals to a wide range of individuals for various purposes, such as finding new friends or connecting with the ones they have more effectively. In this section, we use the JAMM software [29] to generate decision rules. The results are used to understand the influence of recommendation sources on the intention to adopt microbloggings.

4.1 Rules for the Intention to Adopt Microblogging

In this study, the research subjects are users and potential users of Facebook. A total of 382 undergraduate and graduate students from a university in northern Taiwan participated in the survey. The participants were then asked to complete a

self-reported questionnaire containing study measures for their intentions to use microblogging sites such as Facebook. In addition, because daily internet hour usage, past use experience [32] and gender information [6] can also reflect the composition of the users, we also included the three variables as controls in this study. The personal attributes of the participants (gender, daily internet hour usage, and past use experience) and the attributes of the recommendation sources (WOM, advertising, and expert) were conducted. In addition, one decision attribute, the adoption intention, is also included to pre-process the data to construct the information table, which represents knowledge in a DRSA model. The attributes of recommendation sources were measured in three dimensions: WOM (friend or classmate reviews), advertising, and expert recommendations. The respondents were asked to choose the recommendation source they would normally consult and to indicate the extent to which the source was perceived as an influence of recommendation on a 5-point Likert-type scale, with 1 = not very important, 3 = neutral, and 5 = very important. Furthermore, the participants were asked to evaluate their microblogging usage intentions. The survey also presented statements and participants were asked to indicate their level of agreement using multi-item scales, measured on a 5-point Likert-type scale where 1 = strongly disagree, 3 = neutral, and 5 = strongly agree. The domain values of these personal attributes and recommendation sources are shown in Table 1.

Based on the decision rules extraction procedures of the DRSA, a large number of rules related to the intention to use microblogging can be generated. We classified our samples into two classes: "at least 4" (corresponds to having the intention to adopt microbloggings) and "at most 3" (corresponds to having no or little intention to adopt microbloggings). The accuracy of classification for the two decision classes was 99% and 98%, respectively, so most samples of the data were correctly classified.

Table 1. Attribute specification for adoption intention to use microbloggings analysis

Attribute Name	Attribute Values	Preference
Condition attributes		
Gender (a_1)	1: Male; 2: Female	Non-ordered
Daily internet hour usage (a_2)	1: <2 ; 2: 2-4; 3: >5	Non-ordered
Past use experience (a_3)	1: Yes; 2: No	Non-ordered
Word-of-mouth recommendations (a_4)	1: Not very important; 2: Not important; 3: Neutral; 4: Important; 5: Very important	Ordered
Advertising recommendations (a_5)	1: Not very important; 2: Not important; 3: Neutral; 4: Important; 5: Very important	Ordered
Expert recommendations (a_6)	1: Not very important; 2: Not important; 3: Neutral; 4: Important; 5: Very important	Ordered
Decision attributes		
Adoption intention (d_1)	1: Very disagree; 2: Disagree; 3: Neutral; 4: Agree; 5: Very agree	Ordered

Table 2. Rules on adoption intention of use microbloggings

	Rules	Support	Certainty	Strength	Coverage
The person has intention to adopt microblogging ($d_1 \geq 4$)					
1	IF ($a_4 \geq 4$) & ($a_5 \geq 4$) THEN (Having adoption intention to use microbloggings)	241	1	0.63	0.97
2	IF ($a_4 \geq 4$) & ($a_6 \geq 4$) THEN (Having adoption intention to use microbloggings)	240	1	0.63	0.97
3	IF ($a_1 = 1$) & ($a_5 \geq 4$) THEN (Having adoption intention to use microbloggings)	213	1	0.56	0.86
4	IF ($a_2 = 1$) & ($a_6 \geq 4$) THEN (Having adoption intention to use microbloggings)	211	1	0.55	0.85
5	IF ($a_2 = 1$) & ($a_4 \geq 4$) THEN (Having adoption intention to use microbloggings)	210	1	0.55	0.85
6	IF ($a_3 = 1$) & ($a_4 \geq 4$) THEN (Having adoption intention to use microbloggings)	209	1	0.55	0.84
7	IF ($a_3 = 1$) & ($a_5 \geq 4$) THEN (Having adoption intention to use microbloggings)	209	1	0.55	0.84
8	IF ($a_3 = 1$) & ($a_6 \geq 4$) THEN (Having adoption intention to use microbloggings)	209	1	0.55	0.84
9	IF ($a_1 = 1$) & ($a_2 = 1$) & ($a_3 = 1$) THEN (Having adoption intention to use microbloggings)	194	1	0.51	0.78
The person has no or little intention to adopt microblogging ($d_1 \leq 3$)					
1	IF ($a_1 = 2$) & ($a_5 \leq 3$) THEN (No or little adoption intention to use microbloggings)	121	1	0.32	0.90
2	IF ($a_2 = 2$) & ($a_4 \leq 3$) & ($a_6 \leq 3$) THEN (No or little adoption intention to use microbloggings)	110	1	0.29	0.82
3	IF ($a_2 = 2$) & ($a_5 \leq 3$) & ($a_6 \leq 3$) THEN (No or little adoption intention to use microbloggings)	108	1	0.28	0.81

Through DRSA analysis, we generated 12 rules, of which 9 rules apply to the "at least 4" class and 3 rules apply to the "at most 3" class, as illustrated in Table 2. The coefficients of certainty, strength, and coverage associated with each rule are also illustrated. Under the different decision rules, the rule set generates relative strength and coverage. The antecedents of the "at least 4" class of rules explain which attributes microblogging companies need to attract, and the "at most 3" class of rules tells the microblogging companies what attributes they should avoid. Therefore, as Table 2 shows, some of the variables had a higher degree of dependence and may impact the intention to adopt microbloggings more than others. These results illustrate the different degrees of importance of variables for effecting the adoption intention, which could help managers develop marketing strategies.

4.2 Discussions and Managerial Implications

This investigation examined how personal variables and recommendation sources influence the users' intention to adopt microbloggings. In the "at least 4" class, the analytical results showed that users users who trust in recommendation source are more likely to adopt microbloggings and that WOM recommendations influenced the subjects' intention to adopt microbloggings more than advertising recommendations and expert recommendations did. In addition, users who have more daily internet hours usage and who are more familiar than others with microbloggings rely on recommendation sources to adopt microbloggings. Finally, males who trust expert recommendations are more likely to adopt microbloggings. In the "at most 3" class, the anlaytical results showed that the intentions of females who have no confidence in expert recommendations to adopt microbloggings would decrease, as would those of users who have fewer daily internet hours use and users who doubt recommendation sources.

The results of this study have implications for decision-makers. One implication is that marketers may use recommendation sources, especially WOM recommendations, to promote microbloggings usage. For instance, the platform providers can design recommendation activities where users who recommend microbloggings to others are rewarded. Especially in an online environment, our suggestion is consistent with Park et al. [22], who pointed out that marketers should consider providing user-generated information services and recommendations by previous users in the form of electronic word-of-mouth (eWOM). Another implication is that different types of recommendations can attract different types of users. There are differences in how recommendation sources impact the two genders, so the platform providers can apply different recommendation strategies, such as targeting mass media (e.g., a news report) to male users and alternative media (e.g., a discussion forum) to female users.

5 Conclusions and Remarks

DRSA has not been widely used in predicting microbloggings usage, especially in the context of social networks. This study uses DRSA to identify microbloggings decision rules that infer the antecedents of the intent to adopt microbloggings under the effects of different recommendation sources. Future research can extend this study to

apply the other data-mining approaches to extracting the attributes of the intention to adopt microbloggings. The study is limited in that actual behavior was not assessed, so the links between intention and actual behavior in this context remain unexamined.

References

1. Andreasen, A.R.: Attitudes and Customer Behavior: A Decision Model. In: Kassarjian, H.H., Robertson, T.S. (eds.) Perspectives in Consumer Behavior, Scott, Foresman and Company, Glenview, IL, USA, pp. 498–510 (1968)
2. Awareness: Enterprise Social Media: Trends and Best Practices in Adopting Web 2.0 (2008), http://www.awarenessnetworks.com/resources/resources-whitepapers.asp
3. Bagozzi, R.P., Baumgartner, H., Yi, Y.: State versus Action Orientation and The Theory of Reasoned Action: An Application to Coupon Usage. J. Con. Res. 18, 505–518 (1992)
4. Blaszczynski, J., Greco, S., Slowinski, R.: Multi-Criteria Classification – A New Scheme for Application of Dominance-Based Decision Rules. Euro. J. Oper. Res. 181, 1030–1044 (2007)
5. Carchiolo, V., Malgeri, M., Mangioni, G., Nicosia, V.: Emerging Structures of P2P Networks Induced by Social Relationships. Comp. Commun. 31, 620–628 (2008)
6. Cheong, M., Lee, V.: Integrating Web-Based Intelligence Retrieval and Decision-Making from the Twitter Trends Knowledge Base. In: Proceeding of the 2nd ACM Workshop on Social Web Search and Mining, pp. 1–8 (2009)
7. Cox, D.F.: Risk Taking and Information Handling in Consumer Behavior. In: Cox, D.F. (ed.) Risk Taking and Information Handling in Consumer Behavior, pp. 604–639. Boston University Press, Boston (1967)
8. Crisci, R., Kassinove, H.: Effects of Perceived Expertise, Strength of Advice, and Environmental Setting on Parental Vompliance. J. Soci. Psych. 89, 245–250 (1973)
9. Crow, S.M., Fok, L.Y., Hartman, S.J., Payne, D.M.: Gender and Values: What is the Impact on Decision Making? Sex Rol. 25, 255–268 (1991)
10. Davis, F.D.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quart. 13, 319–340 (1989)
11. Gefen, D., Straub, D.W.: Gender Differences in the Perception and Use of E-Mail: An Extension to the Technology Acceptance Model. MIS Quart. 21, 389–400 (1997)
12. Greco, S., Matarazzo, B., Slowinski, R.: A New Rough Set Approach to Evaluation of Bankruptcy Risk. In: Zopounidis, C. (ed.) Operational Tools in the Management of Financial Risk, pp. 121–136. Kluwer Academic Publishers, Boston (1998)
13. Greco, S., Matarazzo, B., Slowinski, R.: Rough Set Theory for Multicriteria Decision Analysis. Euro. J. Oper. Res. 129, 1–47 (2001)
14. Greco, S., Matarazzo, B., Slowinski, R.: Rough Set Methodology for Sorting Problems in Presence of Multiple Attributes and Criteria. Euro. J. Oper. Res. 138, 247–259 (2002)
15. Greco, S., Matarazzo, B., Slowinski, R.: Dominance-based Rough Set Approach as a Proper Way of Handling Graduality in Rough Set Theory. In: Peters, J.F., Skowron, A., Marek, V.W., Orłowska, E., Słowiński, R., Ziarko, W.P. (eds.) Transactions on Rough Sets VII. LNCS, vol. 4400, pp. 36–52. Springer, Heidelberg (2007)
16. Günther, O., Krasnova, H., Riehle, D., Schöndienst, V.: Modeling Microblogging Adoption in the Enterprise. In: Proceedings of the 15th Americas Conference on Information Systems, San Francisco, CA, USA (2009)

17. Häubl, G., Trifts, V.: Consumer Decision Making in Online Shopping Environments: The Effects of Interactive Decision Aids. Market. Sci. 19, 4–21 (2000)
18. Java, A., Song, X., Finin, T., Tseng, B.: Why We Twitter: An Analysis of a Microblogging Community. In: Zhang, H., Spiliopoulou, M., Mobasher, B., Giles, C.L., McCallum, A., Nasraoui, O., Srivastava, J., Yen, J. (eds.) WebKDD 2007. LNCS, vol. 5439, pp. 118–138. Springer, Heidelberg (2009)
19. Liou, J.-H., Tzeng, G.-H.: A Dominance-based Rough Set Approach to customer behavior in the airline market. Inform. Sci. (in press, 2010)
20. Liou, J.-H., Yen, L., Tzeng, G.-H.: Using Decision Rules to Achieve Mass Customization of Airline Services. Euro. J. Oper. Res. (in press, 2010)
21. Murray, K.B.: A Test of Services Marketing Theory: Consumer Information Acquisition Activities. J. Market. 55, 10–25 (1991)
22. Park, D.H., Lee, J., Han, I.: The Effect of On-line Consumer Reviews on Consumer Purchasing Intention: The Moderating Role of Involvement. Inter. J. Elect. Comm. 11, 125–148 (2007)
23. Pawlak, Z.: Rough Set. Inter. J. Comp. Inform. Sci. 11, 341–356 (1982)
24. Richins, M.L., Root-Shaffer, T.: The Role of Involvement and Opinion Leadership in Consumer Word-of-Mouth: An Implicit Model Made Explicit. Adv. Con. Res. 15, 32–36 (1988)
25. Rogers, E.M.: The 'Critical Mass' in the Diffusion of Interactive Technologies in Organizations. In: Kraemer, K.L. (ed.) Information Systems Research Challenge.: Survey Research Methods, Harvard Business School Research Colloquium, vol. 3, pp. 245–264. Harvard Business School, Boston (1991)
26. Rogers, E.M.: Diffusion of Innovations, 5th edn. Free Press, New York (1995)
27. Sankey, D.: Networking sites used for background checks, http://www2.canada.com/components/print.aspx?id=ba0dcc0d-f47d-49c9-add4-95d90fd969ab
28. Sledgianowski, D., Kulviwat, S.: Social Network Sites: Antecedents of User Adoption and Usage. In: Proceedings of the Fourteenth Americas Conference on Information Systems (AMCIS), Toronto, ON, Canada, pp. 1–10 (2008)
29. Slowinski, R.: The International Summer School on MCDM 2006, Class note, Kainan University, Taiwan (2006), http://idss.cs.put.poznan.pl/site/software.htm
30. Slowinski, R., Greco, S., Matarazzo, B.: Rough Set in Decision Making. In: Meyers, R.A. (ed.) Encyclopedia of Complexity and Systems Science, pp. 7753–7786. Springer, New York (2009)
31. Smith, D., Menon, S., Sivakumar, K.: Online Peer and Editorial Recommendation, Trust, and Choice in Virtual Markets. J. Interact. Market. 19, 15–37 (2005)
32. Sung, S.K., Malhotra, N.K., Narasimhan, S.: Two Competing Perspectives on Automatics Use: A Theorectical and Empirical Comparison. Inform. Sys. Res. 17, 418–432 (2005)

Fault Effects Analysis and Reporting System for Dependability Evaluation

Piotr Gawkowski, Monika Anna Kuczyńska, and Agnieszka Komorowska

Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, Warsaw 00-665, Poland

Abstract. The paper describes the concept and the architecture of the data warehouse and reporting modules dedicated to distributed fault injection testbench. The purpose of this warehouse is to collect the data from fault simulation experiments and support researchers in exploration and analysis of these results. The data model of the warehouse, main ETL processes, multidimensional structure of OLAP cube, and predefined reports are discussed. Practical advantages of the presented system are illustrated with some exemplary analyses of the experimental results collected during dependability evaluation of the chemical reactor control algorithm with software implemented fault injection approach.

Keywords: software implemented fault injection, dependability evaluation, data warehouse, OLAP, data exploration.

1 Introduction

Fault injection techniques allow one to disturb the tested application with a predefined fault (in accordance with the given fault model) and consequently analyse the fault propagation, effectiveness of fault detection/tolerance techniques, and check the application behavior [3,5,6]. The fault injection experiment consists of tests - each of which is an execution of the application under tests (AUT) disturbed by the randomly generated fault in respect of the given fault model and distribution strategy. In typical software implemented approach (SWIFI) the execution of the AUT is paused by the fault injection system at the given time instant and the target resource (fault location related to the tested application) is corrupted. After that, the execution of disturbed application resumes and the fault injection system monitors its behavior to examine fault effects.

The possible fault space (in respect to the fault model, location, injection time instant, etc.) is usually huge, so, it is impossible in practice to get 100% fault coverage in an experiment. Numerous fault injection parameters have to be randomised. Even in an experiments with quite low test coverage the obtained dependability properties are representative [3,5]. Nevertheless, in practice the number of tests to assure statistically representative results is high - usually ranging from 1000 to even millions of tests. Moreover, higher test coverage helps to discover the dependability weak points and other phenomena of the AUT. All that makes the fault injection experiment a very time consuming and computing

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 524–533, 2010.

power demanding task. The volume of the collected data makes the manual multidimensional analysis or exploration practically impossible and it is reasonable to handle these tasks with the data warehouse. This approach was already successfully used in dependability evaluation experiments for the DBench project [2,4]. However, our SWIFI tools are oriented mostly towards the evaluation of the hardware-like fault sensitivity (e.g. single event upsets [9]) of software applications. So, different aspects have to be taken into account in the data warehouse: specific to our tool capabilities and goals. In [10], our first data warehouse based system (SOWES) for the analysis of fault effects in computer systems was reported. Now we continue and extend this research with the new data warehouse and (to a large extend) the reporting modules. The new system, FEARS (Fault Effects Analysis and Reporting System) is based on SAS software and provides generic, parametric and user friendly procedures to help in the analysis and exploration of complex simulation experiments results. FEARS proved to be very effective in practical usage, a sample of which is presented in the paper.

The paper is organised as follows. The concept of the DInjector fault injection system is presented in Sect. 2. Section 3 describes the architecture of the data warehouse and prepared reports. Their practical usage example (the evaluation of six versions of chemical reactor control application) is given in Sect. 4. The directions of future research and conclusions are presented in Sect. 5.

2 DInjector Overview

To overcome the SWIFI efficiency bottleneck, our SWIFI tools (FITS for Win32 operating systems and LIN for Linux [6]) evolved from simple standalone systems to the complicated distributed system called DInjector ([3,7,8] and references therein). It integrates several simulators (for different hardware/software platforms) and admits to prepare experiments in heterogeneous environments.

DInjector's architecture is depicted in Fig. 1. The system is available to many users at the same time, so they can conduct their experiments simultaneously. All the phases of experiment can be realised within any available computing node of the farm with installed fault injection core (as a background workload). Nodes are grouped into clusters managed by cluster coordinator (CC_j). Many clusters can be connected to the database. Each coordinator CC_j checks the

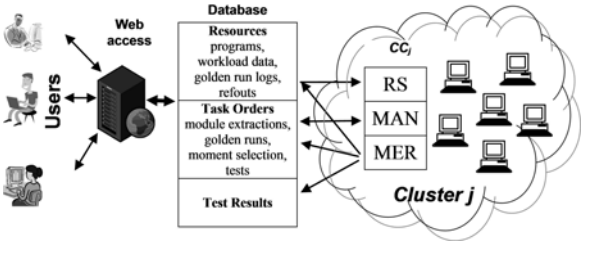


Fig. 1. DInjector architecture

registered jobs in database, creates and distributes the corresponding tasks for the computer nodes, monitors their execution and collects the results. Cluster coordinator has to handle many critical situations, like nodes restarts, primary node workloads or connectivity problems. All these aspects are discussed in [7].

The system's users provide applications to be tested (program code, input data). The experiment is preceded with collecting the reference profile (saved in golden run log - GRL) from non disturbed AUT execution. Then the user specifies experiment configuration, the number, localisation and classes of faults to be injected, etc. [7,8]. All the above mentioned data are stored in the DInjector's database and are described in more detail in [10]. During a test the details of injected faults are logged and the simulator monitors further AUT execution. In particular, up to 10 machine instructions' trace is saved as well as possible exception occurrences and messages from AUT. At the end of the test, the fault injector compares the results produced by the AUT during the test with those found during the golden (referential) run. In general, DInjector distinguishes 5 classes of test results: COR - correct result, INC - incorrect result, SYS - fault detected by the system, TMO - time-out, U - user messages (generated by the program, if an error is detected). Additionally, more detailed qualification of test results' quality is also available.

Distribution of tests between all available computers resulted in almost linear speed-up of executed experiments [7] and, in consequence, the capability of more exhaustive testing. However, the collected volume of information makes the manual analysis ungovernable and it is impossible to analyse them without specialised tools. Moreover, reports obtained directly from the DInjector's transaction database would cause problems of system availability and performance. The developed data warehouse and analytical system addresses these issues.

3 FEARS - Fault Effects Analysis and Reporting System

Fault Effects Analysis and Reporting System (FEARS) moved complicated analytic queries and reporting from DInjector database to the data warehouse (DWH). Splitting up the functionality of transaction database (MS SQL Server 2005 - DInjector) and data warehouse (SAS 9.1.3 [1]) admitted to accommodate and optimise both modules to their specific tasks. Another purpose of this project was to build full automatic process of data analysis, so users can focus on their tasks, not on technical aspects of data loading, processing and reporting. Multidimensional structure of DWH can be directly used in reporting modules reducing the time of reports preparation. Users working in the OLAP cubes (thanks to the SAS Format) see the business model of the database instead of the physical one. Module based construction of SAS software allows one to decompose applications on different computers, to balance the load and get better performance of the whole system. SAS data processing servers and supporting servers (IIS and Tomcat) may be installed on independent computers, remote to clients' software. Users' daily work with FEARS can be done in a web browser, while the programing and administration tools are standalone applications.

FEARS stores data in relational database FEARS_DB. It is organised in classical star schema (see Fig. 2), with one facts table (TESTS) containing main data and measures with granulation of single test (see Sect. 2). It also contains 14 dimensions tables, describing test in different scopes. The model contains EXPERIMENTS table (classification of carried out experiments), GRL_* tables (common data about GRLs for many experiments) and dictionary of possible injection locations (INJECTION_LOCATION table). Facts table TESTS has references for all this tables. In turn tables EXCEPTION_*, INSTRUCTION_*, MESSAGES, VARS and MASKS are in one-to-one relationship with table TESTS. They are dimensions joined with the facts table by foreign keys. As the DInjector collects the trace of the test execution after the fault injection up to 10 machine instructions, exceptions, and messages, the corresponding FEARS tables (EXCEPTION_*, INSTRUCTION_*, and MESSAGES) are transposed and denormalised (i.e. subsequent instructions in subsequent table columns).

Data warehouse contains also two data marts: first to optimise reports (denormalised table TESTS_GRL_VARS_INSTRMNEM_EXC) and the second one with 37 calculations summarising every test (table VARS mentioned before). The purpose of creating this calculation was to help end users to find phenomena taking place during tests, but without manual calculations. From created variables it is possible for instance to monitor changes of operation code in the moments of fault injection, examine influence of thrown exceptions to application work or control change of current instruction address.

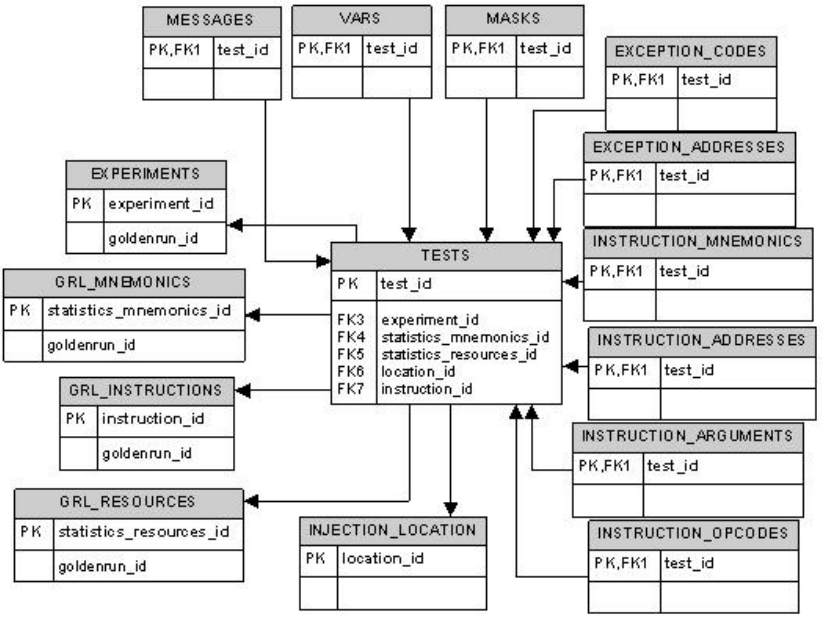


Fig. 2. An overview of the data model

ETL process importing the data to FEARS starts from parsing GRLs and tests results into relational databases: DINJ_GRL and DINJ_TR. This stage runs automatically as triggers on respective inserts into these databases. The next stage of ETL process, filling FEARS_DB, runs on user's request. User simply specifies the identifier of the experiment to be imported. It is also possible to delete data of one experiment from DWH. Both processes are available from SAS Stored Process Web Application. Three groups of data are loaded to the DWH from DInjector's database: programs info and experiments configuration, data from parsing GRLs, and parsed data from strings with test results. Transformation of the source tables into star schema takes place in temporary staging area. After transforming, the data are moved into FEARS_DB and temporary area is cleaned.

After loading experiment into the DWH, a user has to refresh the OLAP cube named *TESTS*. This can be done in a web browser by running SAS stored process. Prepared cube contains 44 measures, which can be explored in 17 dimensions. Reports from OLAP cube show the whole data in cube, which may be explored on different levels and, as opposed to static reports from data tables, allow users to modify reports during data analyses. Users can access cube by SAS Web OLAP Viewer for Java web application. With this application they can not only access the data in the cube, but can also create and save their own OLAP reports (called *explorations* in SAS). OLAP reports can be presented as tables or in graphical mode (as several customisable kinds of charts). In table mode the user can drill down the data in selected hierarchies or focus on drill through one snapshot. Additionally, it is possible to limit the data in the report by using filters. Despite these, some ready-to-use explorations are prepared, which can be used for the first look at the data (also fully customisable):

- TESTS TERM vs INJECTION_LOCATION and MASK - tests termination modes for selected injection locations and fault masks.
- PROGRAMS VS INJECTION_LOCATIONS - programs info and quantity or percent of tests with different fault locations;
- MNEMONICS PROPAGATION - analysis of instruction sequences;
- EXCEPTIONS PROPAGATION - analysis of thrown exceptions;

The prepared set of reports (based on queries from relational FEARS_DB database) is stored on SAS server as SAS Stored Processes and available as web application. If necessary, users can also explore data directly in tables (in SAS Enterprise Guide tool) or use multidimensional reporting on cube. A set of 22 reports is prepared, grouped in 4 categories:

- GRL - overview of all GRLs for selected application;
- EXPERIMENTS - summary of experiments for selected application;
- TESTS SUMMARY - summary of tests for different versions of applications;
- TERM CORR - detail reports, showing executed tests in different configurations, with possibility of selecting granulation level (program or experiment), type of measure (quantity or percent) and parametric. It allows rough analysis of experiments in web interface, as the layout of variables, filters and calculation method can be set-up by user.

Provided reports show a wide substantial spectrum and deep level of analysis. Examined matter (for example, the influence of fault mask and the position of the corrupted bit for the test result) can be considered in different aspects (with additional grouping categories, e.g. mask shift, instruction length, injection location). Additionally, users can create their own variables' combinations using filters in TERM CORR report.

4 Experiments

The developed FEARS was used to investigate dependability properties of several software applications. Results from the DInjector system were loaded to the FEARS and finally, their analyses performed. This section summarises these analyses as well as presents the advantages of such analytical system. In this section an overview of the tested application is given. As several versions are considered, their differences are described. Analyses conducted with FEARS are presented and illustrated with some charts prepared and taken directly from the predefined set of graphical reports of FEARS. It is worth to note that the FEARS is not dedicated to this particular application and can support the analysis of fault injection experiments of any application under tests.

The application under tests is a controller for the chemical reactor. Some research of its dependability with the fault injection approach is already presented in [11]. The goal of the analysed controller is to control the chemical process (in the reactor) using two manipulated variables (i.e. based on periodic readings from two sensors). The application was prepared in 6 different versions (with more sophisticated fault detection/tolerance mechanisms implemented in subsequent versions 1-6). They were compiled with two Microsoft Visual C++ compilers (6.0 and 2005) giving a total number of 12 application versions. The injected faults are single inversions of single bit at random bit position within the target fault location (i.e. CPU registers and executed instructions' codes - in RAM and cache). The triggering moment distribution was random within the execution time domain as well as in the applications' code space. The number of tests for different experiments varied from 1000 to more than 100 000.

The first aspect of the analysis is the validation of the triggering moments' distribution policy. Reports proved that in the case of *random-in-time* policy, the number of times the given instruction address was a fault triggering address was correlated only with the percentage of instruction's executions. On the contrary, in the *random-in-space* policy, each application instruction becomes a trigger with equal probability. Moreover, experiments with different number of triggering moments (test coverage) proved our earlier research ([7,10]), that even significantly low test coverage assures statistically valid results.

Report presented in Fig. 3 summarises experimental results. The differences between subsequent versions as well as the impact of used compilers are clearly stated. In versions 2, 3, and 4 the hardware exception handling is introduced (more and more sophisticated but it works correctly only in VC++ 2005) and, as a result, the significant decrease of system termination cases is observed (see

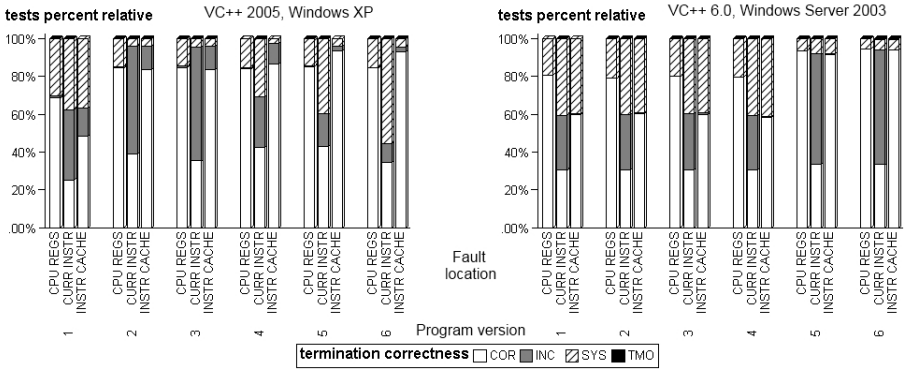


Fig. 3. The summary of experimental results

Section 2). In the case of VC++ 6.0 the improvement is noticeable starting from version 5, in which the pure software fault detectors are introduced (FPU validation in version 5 and acceptance assertion added in version 6). The iterative execution of the control algorithm is a specific property. Naturally renewable resources (e.g. CPU register states, instruction cache) show different susceptibility to faults than those in which the injected fault remains to the end of the test execution (current instruction). It is worth to note that for versions 4-6 in more than 75% of the test cases presented here as incorrect ones the control algorithm has detected an erroneous condition that could not be repaired. Such cases are signalled to the user before the termination. Due to the complexity of judging such resilience cases the analysis requires additional application specific report.

Another interesting property is the analysis of how the fault triggering moment impacts the fault sensitivity. This allows one to identify the weakest points in the application. However, it is made at the machine code level, so the mapping tools are needed to project instruction addresses into the source code level. It is also worth to note that the same source code compiled with different compiler or with just other compilation flag differs in fault sensitivity.

Figure 4 presents the impact of single-bit faults within the instruction code. The corruption of the original instruction IA (I - mnemonic, A - operands) may lead to change of the mnemonic ([I]) or operands([A]). The fault may also change both parts ([I][A]) as well as change the whole instruction code length (and in consequence cascading misinterpretation of the followed instructions codes). It is worth to note, that incorrect result is the most probable for [I]A cases (see the left graph in Fig. 4) if the instruction length wasn't changed. In such cases the exception occurrence is also the most probable. If the length of the instruction changed, a very high probability of exception is observed (please note, that the length changes only if [I] corruption takes place). FEARS allows one to investigate the impact of the particular faulty bit positions (Fig. 5). Some bits within the instruction codes are clearly more sensitive than others (e.g. the first code byte defines the mnemonic or even also operands). The OLAP cube provides

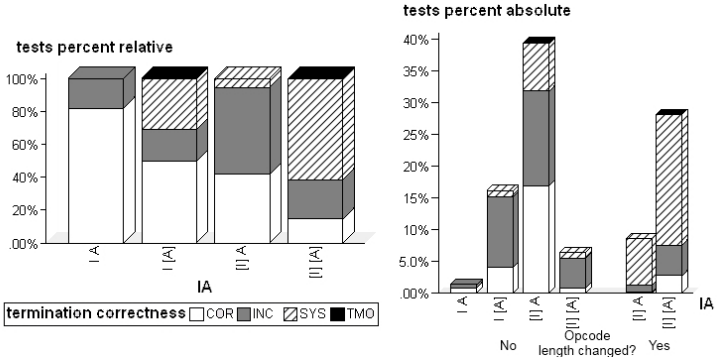


Fig. 4. Effects of changes in instruction code

also the whole insight into the sequence of executed instructions after the fault was injected and into the distribution of corrupted instructions' changes.

The analysis of mnemonic changes showed that some machine instructions are more likely to provoke specific system exceptions (or incorrect results) than others. Figure 6 presents the distribution of exceptions' latency monitored up-to 10 machine instructions starting from the corrupted one (at the fault injection time instant). The immediate exception is denoted as 1. On the other hand, the -1 denotes tests in which the exception was not observed at all or the exception occurred beyond the scope of 10 machine instructions (SYS termination). Note that all bars on the chart sum up to 100% (the actual percentage of tests for given fault location has to be tripled as the equal number of tests for all three presented fault locations were made). Shorter fault latency in terms of detection gives bigger chance for successful recovery, as the corruption within the application and system context is probably lower than in the case of long latency or no

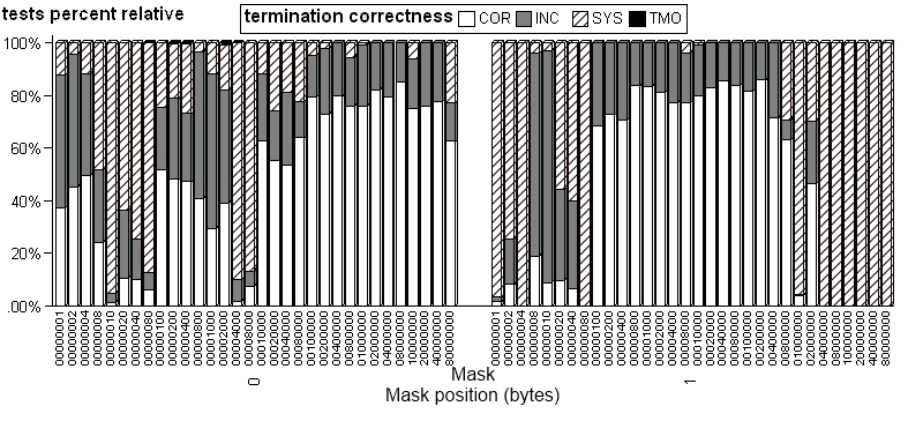


Fig. 5. The impact of the faulty bit positions within instruction code

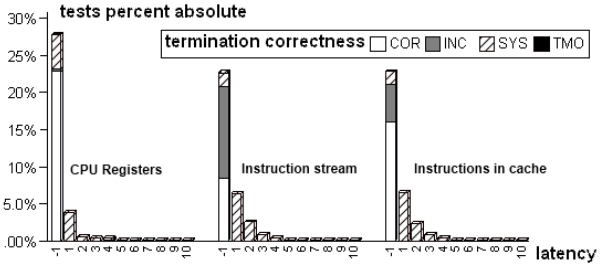


Fig. 6. System exceptions' latency (in the number of machine instructions)

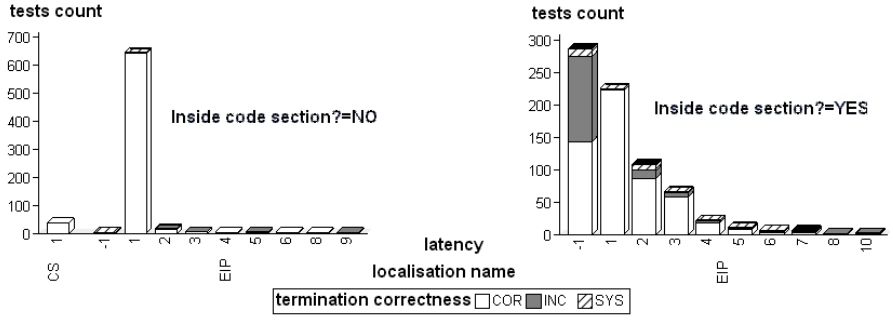


Fig. 7. The impact of exception latency on handling efficiency

detection at all. Longer latency may result in multiple corruptions in a wide range of hard to identify resources. Additionally, the long-running applications (e.g. control algorithms) may collect dormant faults and as a result they can prevent proper error handling later on. Note that multiple bit fault model (e.g. MBU effects) is ironically easier to detect and handle as it raises the probability of early exception occurrence (shorter latency).

Another example of this is given by the results from fault injected into instruction counter register (EIP) presented in Fig. 7. If the corrupted address of the instruction to be executed lies outside the application's memory sections, the access violation exception is raised immediately (latency=1). Simple exception handling can easily correct that in 100% of cases. If the erroneous address fits within the application's memory space, the observed latencies are longer and then the EIP correction does not guarantee the successful processing completion.

5 Conclusion

The developed data warehouse facilitates handling results of simulation experiments targeted at system dependability evaluation. It is supported with some analytical tools. The practical usefulness of the proposed system has been verified in the analyses of the real experimental data. The deep insight into abundance

of details and a huge volume of experimental results from distributed fault injection system discovered several interesting fault-sensitivity dependencies, patterns, and properties. They would not be easy to find with the manual analyses. The gained knowledge helps to improve the fault-robustness of the software applications by introducing dedicated software fault detection and tolerance mechanisms as it was presented in the subsequent versions of the exemplary controller of the chemical reactor.

The positive experience with the FEARS encourages us to continue the system development. Firstly, the enrichment with the data mining capabilities (e.g. association rules, rough sets) will hopefully automate the phenomena discovery. However, the preliminary research in this area showed that the analysed matter requires a high degree of the expert knowledge to discern the most valuable results.

Acknowledgements. This work was supported by the Polish Ministry of Science and Higher Education grant 4297/B/T02/2007/33.

References

1. SAS 9.1.3 Documentation, http://support.sas.com/documentation
2. Madeira, H., Costa, J., Vieira, M.: The OLAP and Data Warehousing Approaches for Analysis and Sharing of Results from Dependability Evaluation Experiments. In: Int'l. Conf. on Dep. Systems and Networks, pp. 22–25. IEEE, Los Alamitos (2003)
3. Gawkowski, P., Sosnowski, J.: Developing Fault Injection Environment for Complex Experiments. In: 14th IEEE Int'l. On-Line Testing Symp., pp. 179–181. IEEE, Los Alamitos (2008)
4. Pintér, G., Madeira, H., Vieira, M., Majzik, I., Pataricza, A.: Integration of OLAP and data mining for analysis of results from dependability evaluation experiments. Int'l. J. of Knowledge Management Studies 2, 480–498 (2008)
5. Benso, A., Prinetto, P. (eds.): Fault injection techniques and tools for embedded systems reliability evaluation. Kluwer Academic Publisher, Dordrecht (2003)
6. Sosnowski, J., Lesiak, A., Gawkowski, P., Włodawiec, P.: Software Implemented Fault Inserters. In: IFAC Work. on Progr. Dev. and Sys., pp. 293–298 (2003)
7. Sosnowski, J., Tymoczko, A., Gawkowski, P.: An approach to distributed fault injection experiments. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) PPAM 2007. LNCS, vol. 4967, pp. 361–370. Springer, Heidelberg (2008)
8. Gawkowski, P., Sosnowski, J.: Experiences with software implemented fault injection. In: Int'l. Conf. on Arch. of Comp. Sys., pp. 73–80. VDE Verlag GMBH (2007)
9. Karnik, T., Hazucha, P., Patel, J.: Characterization of soft errors caused by single event upsets in CMOS processes. IEEE Transactions on Dependable and Secure Computing 1(2), 128–143 (2004)
10. Sosnowski, J., Zygulski, P., Gawkowski, P.: Developing data warehouse for simulation experiments. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 543–552. Springer, Heidelberg (2007)
11. Gawkowski, P., Ławryńczuk, M., Marusak, P.M., Tatjewski, P., Sosnowski, J.: On improving dependability of the numerical GPC algorithm. In: European Control Conference, Budapest, Hungary, pp. 1377–1382 (2009)

Solving the Reporting Cells Problem Using a Scatter Search Based Algorithm

Sónia M. Almeida-Luz[1], Miguel A.Vega-Rodríguez[2], Juan A. Gómez-Pulido[2], and Juan M. Sánchez-Pérez[2]

[1] Polytechnic Institute of Leiria, School of Technology and Management, Department of Informatics Engineering, 2400 Leiria, Portugal
sluz@estg.ipleiria.pt
[2] University of Extremadura, Dept. Technologies of Computers and Communications, Escuela Politécnica. Campus Universitario s/n. 10003 Cáceres. Spain
{mavega,jangomez,sanperez}@unex.es

Abstract. The Location Management problem is an important issue of mobility management, which is responsible for determining the network configuration, with the major goal of minimizing the involved costs. One of the most common strategies of location management is the Reporting Cells (RC) scheme, which mainly considers the location update and the paging costs. In this paper we propose a Scatter Search (SS) based approach applied to the Reporting Cells as a cost optimizing solution, with the objective of achieving the best network configuration defining a subset of cells as reporting cells and the others as non-reporting cells. With this work we want to define the most adequate values of the SS parameters, when applied to the RC problem, using twelve test networks that represent 4 distinct groups divided by size. We also want to compare the performance of this SS based approach with a previous study based on Differential Evolution and also with other approaches presented in the literature. The results obtained are very interesting because they outperform those obtained with other approaches exposed in the literature.

Keywords: Scatter Search, Cost Optimization, Location Management, Reporting Cells Problem, Mobile Networks.

1 Introduction

Nowadays, the increase of mobile networks' users is an important fact that must be considered, because it also involves the growth of network dependent services and applications. Due to this, mobile communication networks [1] must maintain a good response, without losing quality or availability, supporting the increase of users and their respective applications. With the objective that mobile networks keep this quality and availability, it is necessary to consider the Location Management (LM) when the network infrastructures are designed.

The location management problem corresponds to the definition of the network configuration with the objective of minimizing the cost involved, mainly those associated to the user movements and respective tracing [2]. There exist a variety of

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 534–543, 2010.

strategies of LM that are divided into two main groups: static and dynamic schemes [2], [3]. Static schemes, like the Reporting Cells (RC) strategy, are the most common ones in actual mobile networks, because they consider the same network behavior, for all the users.

Finding an optimal set of reporting cells is an NP-complete problem. So, this paper presents a Scatter Search (SS) based approach applied to the reporting cells planning as a cost optimization problem. The major goal of the RC problem is to optimize the configuration planning of mobile networks, by means of reporting cells and no-reporting cells, in a process of minimizing the involved costs. In Section 2 we expose the location management problem, the related costs and also the tuning to the reporting cells scheme. In section 3 we present a succinct description of SS algorithm. Section 4 includes the implementation details. In section 5 we expose the experimental results and respective analysis. In section 6 we compare the performance of our approach with previous work and with approaches proposed by other authors. Finally, section 7 includes conclusions and future work.

2 Location Management Problem

In mobile networks, the LM is one of the major processes of mobility management, because it is responsible for enabling the network to find the most up to date location of each mobile terminal, allowing the users to receive or make calls, independently of their location and time of the day.

Location update and location paging are the two main operations of LM over the mobile networks. The location update (LU) is used to report the current location, executed by mobile terminals when they change their location. The location paging (P) corresponds to the operation of determining the location of the mobile terminal, which is performed by the network when it needs to forward an incoming call to the user.

The generic formula used to determine the LM cost and also considered in previous studies and experiments [4], [5], is:

$$Cost = \beta \times N_{LU} + N_P . \tag{1}$$

The cost of location updates is given by N_{LU}, the cost of paging transactions is given by N_P, and finally β is a ratio constant used in a location update relatively to a paging transaction in the network. Most of the time, mobile users move from one cell to another, in the network, without performing a call. Due to this, the cost of a location update is generally considered to be 10 times greater than the cost of paging, so we have β=10 [6].

In the following subsection we will explain the Reporting Cells strategy and detail how the general formula of location management cost (1) can be readjusted.

2.1 Reporting Cells Problem

The Reporting Cells strategy, proposed by Bar-Noy and Kessler in [7], has the objective of minimizing the location management cost of tracing mobile users. It is characterized by selecting and designating a subset of cells as reporting cells and setting the

others as non-reporting cells (nRC). Considering Fig 1, we observe the configuration planning of a 4x4 network (see Fig.1.a) where RC are represented with value 1 and nRC are represented with value 0 (see Fig. 1b). The location update of a mobile user is performed only when its mobile terminal enters in a reporting cell. For routing an incoming call, the search is restricted to all the cells that compound the vicinity of the last known RC.

We must take in consideration that, for each cell in the network, it is necessary to calculate the vicinity value, which corresponds to the maximum number of cells that the user might page when an incoming call occurs.

The vicinity value of a RC corresponds to the number of nRCs that are reachable from this RC, without pass over other reporting cells, and considering the RC itself. Considering, as an example, the calculus of vicinity value for the cell number 9 in Fig. 1b, we need to count all the neighbor cells that are nRC, respectively cells 8, 13, 14 and 15; and also the RC itself. With this calculus we obtain the vicinity value of 5 for the RC number 9.

The calculus of vicinity value, for a non-reporting cell, must consider the maximum vicinity value among the RCs from where this nRC can be achieved. This is, if the nRC belongs to the neighborhood of several RCs, the calculus must be performed for each of them and then select the highest vicinity value. For example, considering the cell number 2 in Fig. 1b, we observe that it belongs to the neighborhood of the RCs number 4, 5, 6 and 7, which have respectively the vicinity values of 6, 6, 5 and 5. In this process we must select the highest, so we must set the vicinity value of nRC number 2 as 6.

Using the RC planning of Fig.1a and calculating the vicinity values, for all the cells, we obtain the final result shown in Fig. 1c.

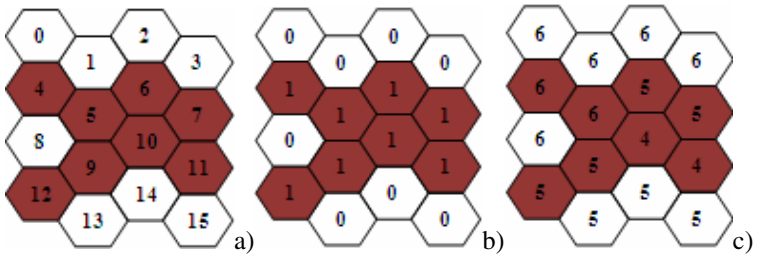


Fig. 1. a) Reporting Cells Network Planning; b) RC (1) and nRC(0); c) Vicinity values

As we mentioned earlier, in the RC scheme the location updates only are performed when a mobile user enters in a reporting cell, so the vicinity value of each cell must be considered. Due to this, the generic formula given by (1) must be readjusted and re-formulated as [4], [8]:

$$Cost = \beta \times \sum_{i \in S} N_{LU}(i) + \sum_{i=0}^{N} N_P(i) \times V(i) . \quad (2)$$

Here we can see that $N_{LU}(i)$ is the total number of location updates for RC i, S indicates the subset of cells defined as RCs, $N_P(i)$ is the number of incoming calls

attributed for cell *i*, N is the total number of cells that compound the mobile network configuration and *V(i)* is the vicinity value calculated for cell *i*. In this work, we want to define what cells will be set as RCs, in each mobile network configuration, with the main goal of minimizing the LM costs.

3 Scatter Search Algorithm

In 1977, Glover [9] introduced Scatter Search (SS) as new evolutionary algorithm. The SS algorithm is characterized by 5 major components [10], [11]: *Diversification Generation method*, *Improvement method*, *Reference Set Update method*, *Subset Generation method* and *Solution Combination method*. Fig. 2 shows the outline of the SS algorithm (see [10], [11] for more details).

SS Algorithm

1: Create Population with PSize different solutions.
Using Diversification Generation method and Improvement method.
2: Define a *RefSet* = $\{x^1,\ldots,x^b\}$ with *b/2* best solutions and *b/2* most diverse solutions of *P*.
3: Order the *RefSet* of solutions, applying their fitness function.
4: Set NewSolution=TRUE.
5: **while** (Exist (NewSolution))
6: Make NewSolution=FALSE
7: Create all different pairs of subsets using the Subset Generation method
8: **while**(Exist (subsets not examined))
9: Apply the Solution Combination method to the solutions of the subset
10: Improve each new solution x with the Improvement method
11: **if** $(f(x) < f(x^b)$ and $(x \notin RefSet)$
12: Set $x^b = x$ and order solutions of *RefSet*
13: Make NewSolution = TRUE

Fig. 2. Outline of Scatter Search algorithm

4 Implementation Details

In this section we detail the decisions taken about implementation details of SS when applied to the RC scheme. We start presenting the test networks used, follow explaining the fitness function used to evaluate the solutions accomplished and finally, expose the major considerations for the parameters definition.

4.1 Test Networks

With the objective of testing our approach, and comparing the results accomplished, we decided to use a set of twelve test networks, available in [12] as benchmark, which we also used in a previous work [4]. We have selected this set of networks because they are based on realistic data and patterns [8] and are divided by size in four different groups.

In Table 1 we expose, as an example, the test network 1, which represents a 4x4 cells configuration. The first column includes the cell identification, the second column represents the number of location updates *NLU* and the third column corresponds to the number of incoming calls *NP*.

Table 1. Test Network 1 – NLU and NP values

Cell	NLU	NP	Cell	NLU	NP	Cell	NLU	NP
0	452	484	6	816	438	12	529	470
1	767	377	7	574	415	13	423	376
2	360	284	8	647	366	14	1058	569
3	548	518	9	989	435	15	434	361
4	591	365	10	1105	510			
5	1451	1355	11	736	501			

4.2 Fitness Function and Parameters Definition

The fitness function is responsible for evaluating each solution generated. In this work, our fitness function will be implemented according to the equation (2), presented in section 2.1.

Considering the outline of the SS algorithm, shown in Fig. 2, we implemented the *diversification generation* method, which is applied to the generation of the initial population, considering the attribution of RC or nRC to each cell, with a probability of fifty percent. As the *improvement* method we decided to apply a local search, characterized by switching a RC with one of their neighbors that are nRC. For the *subset generation* method we decided to use only subsets of size 2. Respectively to the *combination* method we developed a crossover that could be applied to four crossover points taking into account a probability previously determined.

Concluding, our implementation of SS considers four core parameters: initial population size *PSize*; reference set size *RSSize*; probability of combination (crossover) *Cr*; and finally, the number of iterations of local search *nLS*. Furthermore, we must consider that the reference set size is divided into two other parameters, which are respectively, the size of the quality solutions (*nQrs*) and the diversity solutions (*nDrs*).

We have set the following initial values of parameters: *PSize*=100; *RSSize*=10; *nQrs*=5; *nDrs*=5; *Cr*=0.2; *nLS*=1, taking in consideration the suggestion of several authors [10], [11].

5 Experimental Results and Analysis

In this section we expose the different experiments executed, over the twelve test networks, with the objective of studying in detail the use of SS algorithm, when applied to the Reporting Cells problem.

After that, we will analyze the results achieved, determining the best SS configuration and present the network configuration for the best solutions.

Finally we want to compare the results obtained with our previous work, based on Differential Evolution (DE) algorithm, and also with those accomplished by

Alba et al. in [8], which use two approaches based on Hopfield Neural Network with Ball Dropping (HNN-BD) and Geometric Particle Swarm Optimization (GPSO).

5.1 Experiments and Results

In this study we have performed five main experiments, each one adjusted to the most important components of the SS algorithm. To assure the statistical relevance of the results achieved, we decided to perform 30 independent runs, for each experiment and every combination of parameters. In each experiment we set the values of each parameter as the initial one, referred in section 4.2 (if the respective experiment was not performed) or as the one determined in its respective experiment.

The first point of the SS algorithm is generating an initial population with a defined number of distinct solutions (parameter *PSize*). Due to that, we started the experiments with the goal of defining the ideal number of solutions, which will compose the initial population. We have tested *PSize* with the following number of solutions: 10, 25, 50, 75, 100, 125, 150, 175 and 200. Analyzing the results we noticed that the lower costs, for the best and average fitness, were obtained with *PSize*=175, setting it for the following experiments.

The next step was to define the size of the reference set (parameter *RSSize*), which must include the best solutions and also the most diverse solutions from these ones. Fixing *PSize*=175, we checked *RSSize* with the following size values: 2, 4, 6, 8, 10, 12, 14, 16, 18 and 20. We observed that the statistical results were improving with the increase of the *RSSize* and, due of that, we decided to proceed for the next experiments with *RSSize*=20. Moreover, we checked and noticed that improving the *RSSize*, with bigger values than 20, was not significant.

After achieving the most adequate value for *RSSize*, it was necessary to determine its best division between quality solutions (*nQrs*) and most diverse solutions (*nDrs*). To accomplish this task, we tested all the possible combinations between *nQrs* and *nDrs*, knowing that their sum must be 20. Evaluating all the statistical results accomplished, we noticed that the division of *RSSize* between *nQrs*=16 and *nDrs*=6 was the one that performed better.

The following experiment had the purpose of defining the *crossover* probability (*Cr*) to be applied in the *solution combination* method. To proceed with this experiment, we used all the values of parameters already achieved, and follow testing all the next values for *Cr*: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9. Analyzing the best and the average fitness costs, we noticed that the most adequate was *Cr*=0.6.

Finally, in the last experiment, the goal was determining the best number of local search iterations *nLS*, which could allow the best performance of the *improvement* method. To perform this experiment we checked the following configurations of *nLS*: 1, 2, 3, 4 and 5. Observing the results achieved (see Table 2 for the best and average costs obtained of the twelve test networks), we concluded that *nLS*=4 should be elected, because it was the configuration that performs better for all the networks used. Furthermore, with *nLS*=5 the statistical results became worst so, because of this, we elected *nLS*=4 for the final configuration of SS values of parameters.

Table 2. Defining the best number of local search iterations (A – Average cost; B – Best cost)

nLs	TN1	TN2	TN3	TN4	TN5	TN6	TN7	TN8	TN9	TN10	TN11	TN12
1-A	98535	97156	95038	174566	182368	174541	311735	289925	264705	389568	361119	376435
1-B	98535	97156	95038	173701	182331	174519	307695	287149	264204	386688	358167	371990
2-A	98535	97156	95038	174647	182331	174596	310910	287543	264688	388144	360333	375344
2-B	98535	97156	95038	173701	182331	174519	307695	287149	264204	385927	358397	370868
3-A	98535	97156	95038	174703	182331	174585	310573	287821	264589	387399	359586	373451
3-B	98535	97156	95038	173701	182331	174519	307695	287149	264204	385927	358167	370868
4-A	98535	97156	95038	174593	182331	174563	310411	287500	264506	387142	359079	373194
4-B	98535	97156	95038	173701	182331	174519	307695	287149	264204	385927	357714	370868
5-A	98535	97156	95038	174629	182331	174563	310769	287705	264439	387626	359106	372415
5-B	98535	97156	95038	173701	182331	174519	307695	287149	264204	385927	358033	370868

5.2 Analysis and Comparison of Results

Experiments resulted in obtaining the best configuration for the SS parameters when applied to the reporting cells problem, i.e.: *PSize*=175, *RSSize*=20, which is divided in *nQrs*=14, *nDrs*=6; *Cr*=0.6 and *nLS*=4.

With the objective of reinforcing the conclusions obtained over the results, we have performed a statistical analysis using the ANOVA test. For that, we have considered a confidence level of 95% (this is, a significance level of 5% or p-value under 0.05), which means that the differences are unlikely to have occurred by chance with a probability of 95%. We have concluded that the fitness differences have been found as significant in almost all the cases, when we use distinct values for each SS parameter.

Table 3. Comparison of best LM costs for the twelve test networks

	TN1	TN2	TN3	TN4	TN5	TN6	TN7	TN8	TN9	TN10	TN11	TN12
SS	98535	97156	95038	173701	182331	174519	307695	287149	264204	385927	357714	370868
DE	98535	97156	95038	173701	182331	174519	308401	287149	264204	386681	358167	371829
HNN-BD	98535	97156	95038	173701	182331	174519	308929	287149	264204	386351	358167	370868
GPSO	98535	97156	95038	173701	182331	174519	308401	287149	264204	385972	359191	370868

Another goal of this study was to compare the results achieved by the SS approach with a previous study where we have used an approach based on DE algorithm [4]. That was the following task after obtaining the final results. With this comparison we concluded that SS shows the best performance, because it always obtains equal (for test networks 1 to 6, 8 and 9) or best results (using the test networks 7, 10, 11 and 12) than DE, as it is shown in Table 3.

Finally, we also compared the results achieved by SS with those presented by Alba et al. in [8], using HNN-BD and GPSO. Once again, if we observe Table 3, we can conclude that our approach outperforms the results obtained by HNN-BD and GPSO for the test networks 7, 10 and 11; and it equals the lowest fitness costs for all the other ones.

In Fig. 3 we present the configuration for the best solutions for test networks 7, 10 and 11, those where the SS approach surpassed all the other ones.

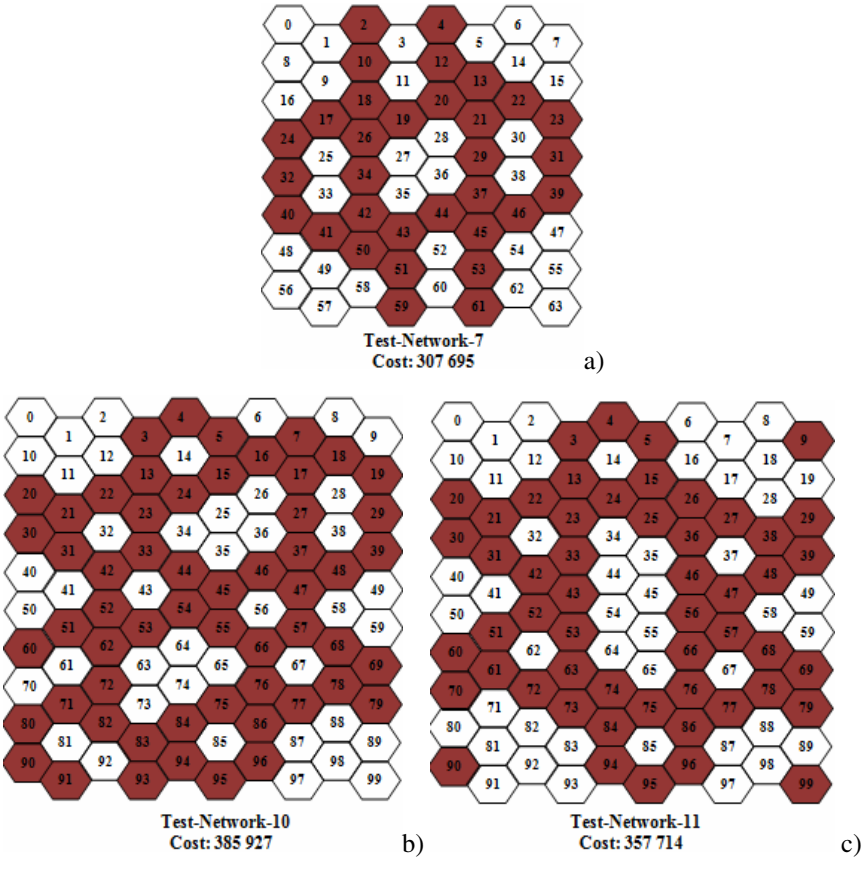


Fig. 3. a) Test Network 7; b) Test Network 10; c) Test Network 11

6 Comparison with Results from Other Applied Algorithms

Once we have finalized the comparison of results, using the set of twelve test networks, we decided to test the performance of our approach, considering the best SS configuration obtained, when applied to additional test networks, also used by other authors.

Initially, we have selected 3 distinct test networks shown in [13], because they were also used in [4] and [8]. With these networks, which represent respectively 4x4, 6x6 and 8x8 instances, we have the objective of comparing the results accomplished by SS, with those achieved by Genetic Algorithms (GA), Ant Colony algorithm (AC) and Tabu Search (TS). Again, we have confirmed the good performance of SS approach, because it always obtains equal or better fitness solutions (lower LM costs), as it is possible to analyze in Table 4.

Table 4. Comparison of best LM costs for the twelve test networks

Test Network	SS	DE	GA	AC	TS
4x4	92833	92833	92833	92833	92833
6x6	211278	211278	229556	211291	211278
8x8	436269	436269	436283	436886	436283

Furthermore, as a final test we selected two bigger networks (7x9 and 9x11 instances) provided in [5] that use a combination of HNN (Hopfield Neural Network) and BDT (Ball Dropping Technique); and also used in [4], [14].

Applying our approach to the 7x9 instance allowed us to surpass all the other approaches with those LM costs achieved. That is, with SS we obtained the fitness value of 120052, with 27 RCs, while with DE [4] the best solution represents a fitness of 120904 with 28 RCs, and the HNN-BDT [5], [14] accomplished the fitness value of 123474 with 27 RCs.

Considering the 9x11 instance, the fitness value achieved is 242914 cost units, with 44 RCs, which surpasses the lowest costs obtained by DE (243957 cost units, with 47 RCs) and also by HNN-DT (243414 cost units, with 43 RCs).

Finishing all of the additional experiments, it is possible to conclude that this SS based approach applied to the reporting cells problem, is very competitive, because it outperforms the results achieved using other artificial techniques.

7 Conclusions and Future Work

This paper focuses on Reporting Cells problem, presenting a SS based approach applied to it, with the objective of minimizing the location management costs.

We have executed several experiments with the intention of achieving the best configuration for the values of SS parameters. For this purpose we have selected twelve distinct test networks, and subsequently to a big number of runs, we set the most adequate parameter values like: *PSize*=175, *RSSize*=20, divided in *nQrs*=14 and *nDrs*=6; *Cr*=0.6 and *nLS*=4. With the results obtained we concluded that our approach has a good performance, and can be effectively applied to the RC problem, since, comparing with the results obtained by other authors, which use GSPO and HNN-BD, we achieve equal or even better fitness (i.e., location management costs).

We also tested our approach with other artificial life techniques like genetic algorithms (GA), ant colony algorithm (AC), tabu search (TS), differential evolution (DE) and a combination of hopfield neural network (HNN) and ball dropping technique (BDT); the results are very encouraging, because our approach always equals or outperforms the results achieved by the other techniques.

Respectively in future work we want to compare the results accomplished with the Reporting Cells strategy with the ones obtained with the Location Areas strategy (this is another well-known static scheme of location management). Performing that study, we desire to determine the most adequate strategy, because both consider the location update and paging costs of the location management.

Acknowledgments. This work was partially funded by the Spanish Ministry of Science and Innovation and FEDER under the contract TIN2008-06491-C04-04 (the M* project). Thanks also to the Polytechnic Institute of Leiria, for the economic support offered to Sónia M. Almeida-Luz to make this research.

References

1. Pahlavan, K., Levesque, A.H.: Wireless Information Networks. John Wiley & Sons, Chichester (1995)
2. Tabbane, S.: Location Management Methods for Third Generation Mobile Systems. IEEE Communications Magazine 35, 72–84 (1997)
3. Wong, V.W.S., Leung, V.C.M.: Location Management for Next-Generation Personal Communications Networks. IEEE Network 14(5), 18–24 (2000)
4. Almeida-Luz, S.M., Vega-Rodríguez, M.A., Gómez-Púlido, J.A., Sánchez-Pérez, J.M.: Differential evolution for solving the mobile location management. Applied Soft Computing (2009) (in Press), doi:10.1016/j.asoc.2009.11.031
5. Taheri, J., Zomaya, A.Y.: A Modified Hopfield Network for Mobility Management. Wireless Communications and Mobile Computing 8, 355–367 (2008)
6. Gondim, P.R.L.: Genetic Algorithms and the Location Area Partitioning Problem in Cellular Networks. In: 46th IEEE Vehicular Technology Conf. Mobile Technology for the Human Race, vol. 3, pp. 1835–1838 (1996)
7. Bar-Noy, A., Kessler, I.: Tracking Mobile Users in Wireless Communications Networks. IEEE Transactions on Information Theory 39, 1877–1886 (1993)
8. Alba, E., García-Nieto, J., Taheri, J., Zomaya, A.Y.: New Research in Nature Inspired Algorithms for Mobility Management. In: Giacobini, M., Brabazon, A., Cagnoni, S., Di Caro, G.A., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Fink, A., McCormack, J., O'Neill, M., Romero, J., Rothlauf, F., Squillero, G., Uyar, A.Ş., Yang, S. (eds.) EvoWorkshops 2008. LNCS, vol. 4974, pp. 1–10. Springer, Heidelberg (2008)
9. Glover, F.: Heuristics for Integer Programming Using Surrogate Constraints. Decision Sciences 8, 156–166 (1977)
10. Martí, R., Laguna, M., Glover, F.: Principles of Scatter Search. European Journal of Operational Research 169, 359–372 (2006)
11. Laguna, M., Hossell, K.P., Martí, R.: Scatter Search: Methodology and Implementation in C. Kluwer Academic Publishers, Norwell (2002)
12. Test Networks Benchmark, `http://oplink.lcc.uma.es/problems/mmp.html` (accessed on February 2010)
13. Subrata, R., Zomaya, A.Y.: A Comparison of Three Artificial Life Techniques for Reporting Cell Planning in Mobile Computing. IEEE Transactions on Parallel and Distributed Systems 14(2), 142–153 (2003)
14. Taheri, J., Zomaya, A.: Bio-inspired Algorithms for Mobility Management. In: Proceedings of ISPAN 2008 - The International Symposium on Parallel Architectures, Algorithms, and Networks, pp. 216–223. IEEE Computer Society, Los Alamitos (2008)

Learning Age and Gender Using Co-occurrence of Non-dictionary Words from Stylistic Variations

R. Rajendra Prasath*

Department of Computer and Information Science (IDI)
Norwegian University of Science and Technology (NTNU),
Sem Sælands vei 7-9, NO - 7491, Trondheim, Norway
rajendra@idi.ntnu.no
http://www.idi.ntnu.no/~rajendra

Abstract. This work attempts to report the stylistic differences in blogging for gender and age group variations using slang word co-occurrences. We have mainly focused on co-occurrence of non dictionary words across bloggers of different gender and age groups. For this analysis, we have focused on the feature *use of slang words* to study the stylistic variations of bloggers across various age groups and gender. We have modeled the co-occurrences of slang words used by bloggers as graph based model where nodes are slang words and edges represent the number of cooccurrences and studied the variations in predicting age groups and gender. We have used demographically tagged blog corpus from ICWSM Spinner dataset for these experiments and used Naive Bayes classifier with 10 fold cross validations. Preliminary results shows that the concurrence of of slang words could be a better choice for predicting age and gender.

Keywords: Stylometrics, Demographic analysis, Slang / Out of vocabulary words, Classification, Graph Clustering.

1 Introduction

Information Retrieval (IR) techniques are useful in stylistic classification and can improve the results achieved through by identifying documents that matches a certain demographic profile. The common demographic features like age and gender are used for analyzing stylistic variations as the blogs generally contain these information provided by the author. Writing style is a result of the subconscious habit of the humans who use a number of available options to present the same thing. The writing style varies with the evolution of the usage of the language in certain period, genre, situation or individuals. Variations are of two types - variation within a norm which is grammatically correct and deviation from the norm which is ungrammatical. The variations can be described in linguistic as well as statistical terms[1]. Concept and themes[2] can be determined from variations within the norm while the usage of non-dictionary words or *slang* is an example of deviation from a norm.

* This work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 544–550, 2010.

2 Related Work

Owing to paucity of insufficient annotated data, the research on the usage of language pattern by different social groups was heavily constrained. The effective analysis with learning bloggers' age and gender from weblogs, based on usage of keywords, parts of speech and other grammatical constructs, has been presented in [3,4,5,6]. Pennebaker, et al. [7], Pennebaker and Stone [8] and Burger and Henderson, 2006 [4] reported age linked variations. Linguistic styles based on gender (males and females) were characterized by J. Holmes [9]. Expert used spoken language [1], Palander worked on electronic communications[10], and S. Herring analyzed correspondence[11]. Patton and F.Can analyzed four novels using six style markers which showed the best separation for "most frequent words" and "sentence lengths" [12]. Also they analyzed the change of writing style with time and demonstrated that there is a decrease in average word length as the age of the news column increases[13]. Simkins reported no difference between male and female writing style in formal contexts [14]. Koppel et al. estimated author's gender using the British National Corpus text [15]. By using function words and part-of-speech, Koppel et al. reported 80% accuracy for classifying author's gender. Koppel et al. also stated that female authors tend to use pronoun with high frequency, and male authors tend to use numeral and representation related numbers with high frequency. Corney et al. estimated author's gender from e-mail content [16]. In addition to function words and part-of-speech and n-grams [5,15], they used HTML tags, the number of empty lines, average length of sentences for features for SVM [17]. Recently, results of stylistic differences in blogging for gender and age group variations are reported based on two mutually independent features: use of slang words and the variation in average length of sentences[18].

3 Dataset

A blog corpus[1] is made available by ICWSM 2009[20] and the blogs in this corpus did not have any tag for demographic information. However, it had the resource link which had the URL of the blogger's home page. In the above corpus, blogs from blog.myspace.com had the maximum occurrence and had the demographic details of the blogger in its home page. The home page of these URLs were crawled and processed to retrieve gender, status (married, unmarried), age, zodiac sign, city, state and country corresponding to each URL. With the available valid URL list, the downloaded data from these URLs gives 342,514 files. The blogs in which the blogger's age has been reported as below 20 has been grouped in 10s age group, those in the age group of 20 to 29 as 20s, those in 30 to 39 as 30s and 40 and above has been put in 40s age group. The distribution of these files over age and gender is given in Figure 1. The total number of males and females are 159,729 and 182,785 respectively and for the experiments, we considered only 95,245 blog posts in which 44,434 are males and 50,811 are females. The blogs distributions across age groups with respect to gender is given in Table 1.

[1] Provided by Spinn3r.com [19].

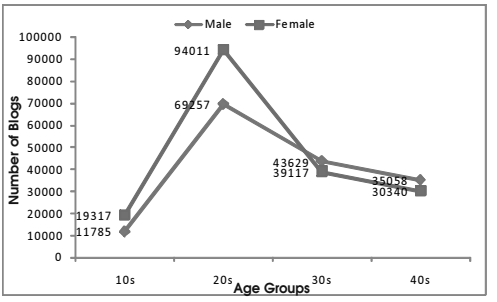


Fig. 1. Number of files in age groups and gender

Table 1. Blogs distribution across Age Groups and Gender

	10s	20s	30s	40s	Total
Male	4,104	19,625	11,411	9,294	44,434
Female	6,142	25,323	10,551	8,795	50,811

4 Feature Selection

Finding good features is always a challenging task and such feature selection provides accuracy improvements in classification. It is straight forward from IR perspectives that the words with many occurrences collected from a corpora may not be a good distinguishing feature. Still, an analysis of the words occurring many times in a subset of a corpora can be the marker [21]. For example, reference to 'attending school' leads to an instant 'teenage' classification. Features for stylistic variations are generally based on character or morphological features or lexical features. In our experiments we used cooccurrence of non-dictionary words as features. As per our literature survey, the usage of slang words has not yet been explored well for the study of stylistic variation.

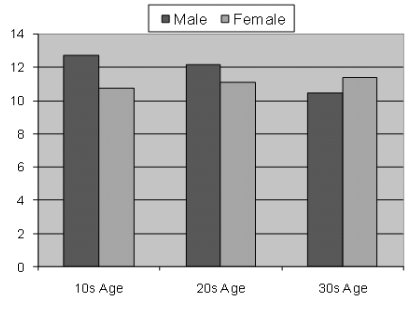


Fig. 2. Average Sentence length across Age vs Gender[18]

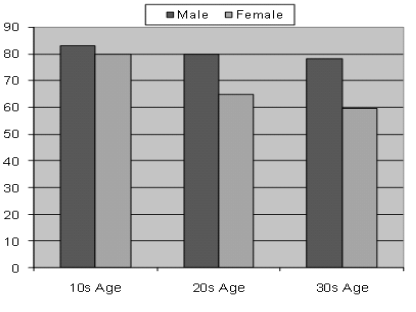


Fig. 3. Non-dictionary words (per 1000 words) across age groups[18]

Koppel [15] used a list of 30 words (as a distinguishing factor) for learning age and gender respectively. These words were detected to be having an extreme variation in

usage across gender and age groups. Similarly, in [22], out-of-dictionary words were augmented to increase the accuracy of results. For the purpose of learning age and gender classifier, each document is represented as a numerical vector in which each entry represent the normalized frequency of a corresponding word in the feature set. Many Stylistic features were applied on formal writings and especially on classical works of literature and results have been reported using average sentence length as a feature. The analysis of blogs based on average sentence length variations is challenging as blogs lack editorial and grammatical checks. The variation of average sentence length on age and gender basis is given in Figure 2. Similarly Figure 3 shows the usage of non-dictionary words per 1000 words across various age groups(refer to [18]).

As blogs are informal writing to express their opinions without any bounds, blogosphere has slowly filled up with many non-dictionary words that are understandable and commonly used by online community. We refer to some of them as *slangs*, *smiley*, *out of vocabulary words*, *chat abbreviations*, etc. The named entities are also non-dictionary words. There are words that are intentionally misspelled, repeated, extended or shortened to have a different effect on the reader, express emotion or save the time of blogging. All these words and even the frequency of use of such words are contributable features in stylistics. For our experiments with non-dictionary words, Ispell[2] [23] was run and the frequency of all the non-dictionary words used by males and females for detecting gender variation was obtained. From these, only those words were selected as feature which had an occurrence of >10. This generated a list of 667 words.

Table 2. List of a few Content word frequency per 10000 words in age groups

	$\frac{WC}{(\sum \text{ WC in that age grp})\text{x}10000}$		
Word	10s age	20s age	30s age
college	4.433	1.173	0.829
bored	2.399	1.892	0.789
boring	0.966	0.687	0.618
dumb	1.266	0.870	0.447
semester	1.333	0.813	0.263
apartment	0.599	1.205	0.487
beer	0.466	0.826	0.908
album	0.966	1.463	1.684
development	0.099	0.176	0.171
local	0.499	0.706	1.803
son	30.26	28.80	28.55
workers	0.099	0.233	0.394

5 Classification Algorithm and Tool

Clustering is commonly used to identify a pattern / structure in the bunch of unlabeled data. In general, clustering organizes data into groups whose members are related in some way and two or more data can be grouped into the same cluster if they are, in

[2] http://www.gnu.org/software/ispell/ispell.html

some way, falling close to each others' context. Clustering has many useful applications like finding a group of people with similar behavior, processing orders, grouping plants and animals, grouping web blog data to find access to similar patterns. In this work, we have attempted to identify the similar usage of slang words across different age gender groups of bloggers using kernel-based multilevel clustering algorithm proposed by Dhillon et al. [24] . This attempt first makes slang cluster vectors using unsupervised out of words cooccurrences and then using them, supervised learning is performed for learning age and gender of the bloggers.

Algorithm 1. Learning Age and Gender through Clustering Non Dictionary Words

Input: A set of n blogs $D = \{d_1, d_2, d_3 \cdots, d_n\}$;
A set of predefined category labels C: *Age*(= 10s, 20s, 30s)
(similarly for *Gender* [= male, female] also)

Procedure:

1: Extract the list of all out of vocabulary words from the blog descriptions
2: **for** each slang word f_i in the list **do**
3: identify the existence of edges from f_i to all slang words with nonzero positive weight.
4: Store the co-occurring information with its corresponding edge weight
5: **end for**
6: Use kernel-based multilevel graph clustering algorithm and perform clustering to generate cluster IDs
7: For every cluster ID, generate slang cluster vectors using the list of out of vocabulary words
8: Augment the blog description with cluster ID mappings and generate the output data file in Weka format.
9: Build classifier using this data file and record the classification accuracy

Output: The category label for bloggers' *Age* (similarly for *Gender*)

Table 3. List of a few Content word frequency per 10000 words in gender

	$\frac{\text{Male Occ}}{10000}$	$\frac{\text{Female Occ}}{10000}$
mom	4.543	7.844
software	0.131	0.051
nations	0.464	0.142
economic	0.159	0.079
shopping	0.304	0.845
cried	0.159	0.759
pink	0.256	0.497
cute	0.671	1.662
kisses	0.096	0.217
boyfriend	0.297	1.411
husband	0.297	1.765
hubby	0.034	0.359

Naive Bayes classifier for predicting the blogger's age group or gender from the stylistic features were trained using the WEKA toolkit [25]. During training, classifiers

are created by the selection of a set of variables for each feature and classifier parameters are tuned through 10 fold cross-validation. To evaluate the classifier, the given data is split into training and test data and the trained classifier is used to predict the blogger's demographic profile on the test data [26].

Co-occurrence of non-dictionary words are exhibiting good cluster similarities in the co-occurrence graph. Presently it is tested on the subset of blog data described in section 3. The formed clusters reveals that the odd minded people use same style across age and gender. The detailed results are in progress. Also inter - relationships with Age and Gender variations are closely predictable with these graph based clustering approach.

6 Conclusion and Future Work

It is clearly seen that Teenage bloggers use more out-of-dictionary words than the adult bloggers. Furthermore, for bloggers of each gender, there is a clear distinction between usage of a few slangs[18]. Based on these results, we analyzed and found that generally in bloggers age, teenagers use smaller sentences compared to the adult bloggers with slight variations. With the available data and the existing experiments, it cannot be confirmed that the average sentence length increases or decreases with age. The stylistic difference in usage of slang predicts the age and gender variation with certain accuracy. Average sentence length itself is not a good feature to predict the variation as there is a wide variation in sentence length in informal writing. However, the feature of average sentence length can be augmented with slang to slightly increase its prediction efficiency. Both these features when augmented with other features like content words reported earlier, increases the prediction accuracy by a good amount.

The usage of slang can also be a good feature to predict the geographical location or the ethnic group of the user due to the heavy usage of a particular out-of-dictionary word or named entities at certain regions. A sufficiently huge corpus collected over a decade will be useful to study the variation of sentence length of users with age and variations in individuals language use over the course of their lives. This corpus can also be used to study the evolution and death of the slang words with time.

References

1. McMenamin, G.R.: Forensic Linguistics: Advances in Forensic Stylistic. CRC Press, Boca Raton (2002)
2. Leximancer Manual V.3: Leximancer (2009), http://www.leximancer.com (last accessed on January 22, 2009)
3. Argamon, S., Koppel, M., Avneri, G.: Routing documents according to style. In: Proc. of First Int. Workshop on Innovative Inform. Syst. (1998)
4. Burger, J.D., Henderson, J.C.: An exploration of observable features related to blogger age. In: Proc. of the AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs (2006)
5. Schler, J., Koppel, M., Argamon, S., Pennebaker, J.: Effects of age and gender on blogging. In: Proc. of the AAAI Spring Symposia on Computational Approaches to Analyzing Weblogs (April 2006)

6. Yan, R.: Gender classification of weblog authors with bayesian analysis. In: Proc. of the AAAI Spring Symp. on Computational Approaches to Analyzing Weblogs (2006)
7. Pennebaker, J.W., Francis, M.E., Booth, R.J.: Liwc 2001, Linguistic Inquiry and Word Count (2001)
8. Pennebaker, J.W., Stone, L.D.: Words of wisdom: Language use over the lifespan. Journal of Personality and Social Psychology 85, 291–301 (2003)
9. Holmes, J.: Women's talk: The question of sociolinguistic universals. Australian Journal of Communications 20(3) (1993)
10. Palander-Collin, M.: Male and female styles in 17th century correspondence: I think. Language Variation and Change 11, 123–141 (1999)
11. Herring, S.: Two variants of an electronic message schema. In: Herring, S. (ed.) Computer-Mediated Communication: Linguistic, Social and Cross-Cultural Perspectives, vol. 11, pp. 81–106 (1996)
12. Patton, J.M., Can, F.: A stylometric analysis of yaşar kemal's İnce memed tetralogy. Computers and the Humanities 38, 457–467 (2004)
13. Can, F., Patton, J.M.: Change of writing style with time. Computers and the Humanities 38, 61–82 (2004)
14. Simkins-Bullock, J., Wildman, B.: An investigation into relationship between gender and language Sex Roles, vol. 24. Springer, Netherlands (1991)
15. Koppel, M., Argamon, S., Shimoni, A.R.: Automatically categorizing written texts by author gender. Literary and Linguistic Computing 17(4), 401–412 (2002)
16. Corney, M., de Vel, O., Anderson, A., Mohay, G.: Gender-preferential text mining of e-mail discourse. In: 18th Annual Computer Security Appln. Conference (2002)
17. Brank, J., Grobelnik, M., Milic-Frayling, N., Mladenic, D.: Feature selection using support vector machines. In: Proc. of the 3^{rd} Int. Conf. on Data Mining Methods and Databases for Engg., Finance, and Other Fields, pp. 84–89 (2002)
18. Rustagi, M., Prasath, R.R., Goswami, S., Sarkar, S.: Learning age and gender of blogger from stylistic variation. In: Chaudhury, S., Mitra, S., Murthy, C.A., Sastry, P.S., Pal, S.K. (eds.) PReMI 2009. LNCS, vol. 5909, pp. 205–212. Springer, Heidelberg (2009)
19. Spinn3r: Spinn3r - indexing blogosphere, http://www.spinn3r.com (last accessed on March 01, 2009)
20. ICWSM 2009: Icwsm 2009 (May 2009); ICWSM 2009 Spinn3r Dataset
21. Datta, S., Sarkar, S.: A comparative study of statistical features of language in blogs-vs-splogs. In: AND 2008: Proc. of the second workshop on Analytics for noisy unstructured text data, pp. 63–66. ACM, New York (2008)
22. Goswami, S., Sarkar, S., Rustagi, M.: Stylometric analysis of bloggers' age and gender. To appear in Proc. of ICWSM (2009)
23. Ispell: Ispell (2009), http://www.gnu.org/software/ispell/ (last accessed on March 02, 2010)
24. Dhillon, I.S., Guan, Y., Kulis, B.: Weighted graph cuts without eigenvectors a multilevel approach. IEEE Trans. Pattern Anal. Mach. Intell. 29(11), 1944–1957 (2007)
25. Witten, I.H., Frank, E.: DataMining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
26. Estival, D., Gaustad, T., Pham, S.B., Radford, W., Hutchinson, B.: Tat: an author profiling tool with application to arabic emails. In: Proc. of the Australasian Language Technology Workshop, pp. 21–30 (2007)

Disturbance Measurement Utilization in Easily Reconfigurable Fuzzy Predictive Controllers: Sensor Fault Tolerance and Other Benefits

Piotr M. Marusak

Institute of Control and Computation Engineering, Warsaw University of Technology, ul. Nowowiejska 15/19, 00–665 Warszawa, Poland
P.Marusak@ia.pw.edu.pl

Abstract. The easily reconfigurable predictive controllers are supplemented with a mechanism of disturbance measurement utilization. It is done in such a way that the main advantage of the controllers – their simplicity – is maintained. The predictive controllers under consideration are based on fuzzy Takagi–Sugeno (TS) models in which step responses are used as local models. These models are supplemented with the parts describing the influence of disturbances on the outputs of the control plant. Then, the controllers are formulated in such a way that the control signals are easily generated. Efficiency and usefulness of the predictive controllers utilizing disturbance measurement is demonstrated in the example control system of a nonlinear control plant with delay.

Keywords: fuzzy systems, fuzzy control, predictive control, nonlinear control, fault–tolerant control, Dynamic Matrix Control.

1 Introduction

The topic of the paper is the result of continuation of research concerning easily reconfigurable fuzzy predictive controllers proposed in [6]. Unlike other types of fuzzy predictive controllers, the discussed ones are formulated in such a way that the control law can be easily obtained analytically. Necessity of solving an optimization problem or calculating numerically a matrix inverse is avoided. Thanks to easiness of reconfiguration, the discussed controllers can be successfully used, e.g. when modification of the control law is needed after actuator fault [6].

In the paper, easily reconfigurable fuzzy predictive controllers are supplemented with the mechanism of taking the disturbance measurement into consideration. This mechanism is important especially when a sensor fault occurs in the control system. It is because in such a case one of the feedback loops is affected by the failure and is in fact interrupted. In such a situation control quality strongly depends on the quality of the model used for prediction. If the model

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 551–559, 2010.

is supplemented with information about influence of the disturbance (which can be measured or estimated) on the process then the operation of the controller based on such a model can be significantly improved.

In the next section the easy to reconfigure, analytical Fuzzy DMC (FDMC) controllers utilizing disturbance measurement are formulated. The modification does not complicate the controllers significantly. Moreover, calculation of the control action is still very simple for SISO (Single Input Single Output) control plants as well as for plants with two inputs and two outputs. Sect. 3 contains description of experiments performed in a control system of a nonlinear control plant with significant delay. The obtained results illustrate well benefits obtained thanks to the mechanism of disturbance measurement utilization applied in the controllers under consideration. Last section contains a short summary.

2 Efficient Fuzzy Analytical Predictive Controllers with Disturbance Measurement

The predictive control algorithms, during control signal generation, predict future behavior of the control plant many sampling instants ahead using a process model [1,3,8,10]. The control signal is derived in such a way that the prediction fulfills assumed criteria. In current research the following optimization problem is to be solved at each iteration of the algorithm:

$$\min_{\Delta \boldsymbol{u}} \sum_{j=1}^{n_y} \sum_{i=1}^{p} \kappa_j \cdot \left(\overline{y}_k^j - y_{k+i|k}^j \right)^2 + \sum_{j=1}^{n_u} \lambda_j \cdot \left(\Delta u_{k|k}^j \right)^2 \quad , \tag{1}$$

where $\overline{y}_k^j$ is a set–point value for the j^{th} output, $\kappa_j \geq 0$ and $\lambda_j \geq 0$ are weighting coefficients for the predicted control errors of the j^{th} output and for the changes of the j^{th} manipulated variable, respectively, p is the prediction horizon, n_y and n_u denote number of output and manipulated variables, respectively, $\Delta \boldsymbol{u} = \left[\Delta u_{k|k}^1, \ldots, \Delta u_{k|k}^{n_u} \right]^T$ is the vector of future changes of manipulated variables (obtained as a solution of the optimization problem), $y_{k+i|k}^j$ is a value of the j^{th} output for the $(k+i)^{\text{th}}$ sampling instant predicted at the k^{th} sampling instant using a control plant model.

The proposed fuzzy controllers are based on fuzzy models which have local models in the form of step responses. Such a model is relatively easy to obtain. It is sufficient to collect a few sets of step responses (near a few operating points). Then, the membership functions can be chosen using expert knowledge and tuned, if necessary, using, e.g. fuzzy neural network. The fuzzy model is thus composed of the following rules which also contain model of influence of disturbance on outputs of the plant:

Rule f: (2)

$$\text{if } y_k^{j_y} \text{ is } B_1^{f,j_y} \text{ and } \ldots \text{ and } y_{k-n+1}^{j_y} \text{ is } B_n^{f,j_y} \text{ and}$$
$$u_k^{j_u} \text{ is } C_1^{f,j_u} \text{ and } \ldots \text{ and } u_{k-m+1}^{j_u} \text{ is } C_m^{f,j_u}$$
$$\text{then } \hat{y}_k^{j,f} = \sum_{m=1}^{n_u} \sum_{n=1}^{p_d-1} a_n^{j,m,f} \cdot \Delta u_{k-n}^m + a_{p_d}^{j,m,f} \cdot u_{k-p_d}^m + \sum_{n=1}^{p_z-1} b_n^{j,f} \cdot \Delta z_{k-n} + b_{p_z}^{j,f} \cdot z_{k-p_z} \ ,$$

where $y_k^{j_y}$ is the $j_y{}^{\text{th}}$ output variable value at the k^{th} sampling instant, $u_k^{j_u}$ is the $j_u{}^{\text{th}}$ manipulated variable value at the k^{th} sampling instant, z_k is the disturbance variable estimate at the k^{th} sampling instant, $B_1^{f,j_y}, \ldots, B_n^{f,j_y}, C_1^{f,j_u}, \ldots, C_m^{f,j_u}$ are fuzzy sets, $a_n^{j,m,f}$ are coefficients of step responses in the f^{th} local model describing influence of the m^{th} manipulated variable on the j^{th} output, $b_n^{j,f}$ are coefficients of disturbance step response in the f^{th} local model describing influence of the disturbance on the j^{th} output, p_d, p_z are equal to the number of sampling instants after which the coefficients of the step responses can be assumed as settled, $j_y = 1, \ldots, n_y$, $j_u = 1, \ldots, n_u$, $f = 1, \ldots, l$, l is number of rules.

The output value of the fuzzy model (2) is calculated at each iteration using current values of process variables and fuzzy reasoning:

$$\hat{y}_k^j = \sum_{m=1}^{n_u} \left(\sum_{n=1}^{p_d-1} \widetilde{a}_n^{j,m} \cdot \Delta u_{k-n}^m + \widetilde{a}_{p_d}^{j,m} \cdot u_{k-p_d}^m \right) + \sum_{n=1}^{p_z-1} \widetilde{b}_n^j \cdot \Delta z_{k-n} + \widetilde{b}_{p_z}^j \cdot z_{k-p_z} \ , \quad (3)$$

where $\widetilde{a}_n^{j,m} = \sum_{f=1}^{l} \widetilde{w}_f \cdot a_n^{j,m,f}$, $\widetilde{b}_n^j = \sum_{f=1}^{l} \widetilde{w}_f \cdot b_n^{j,f}$, $\widetilde{w}_f$ are the normalized weights calculated using fuzzy reasoning, see e.g. [7,9]. The obtained model may be interpreted as the step response model which describes behavior of the control plant near the current operating point.

The output value for the $(k+i)^{\text{th}}$ sampling instant predicted at the k^{th} sampling instant is then calculated using the following formula:

$$\hat{y}_{k+i|k}^j = \sum_{m=1}^{n_u} \left(\sum_{n=1}^{i} \widetilde{a}_n^{j,m} \cdot \Delta u_{k-n+i}^m + \sum_{n=i+1}^{p_d-1} \widetilde{a}_n^{j,m} \cdot \Delta u_{k-n+i}^m + \widetilde{a}_{p_d}^{j,m} \cdot u_{k-p_d+i}^m \right) + \sum_{n=1}^{p_z-1} \widetilde{b}_n^j \cdot \Delta z_{k-n+i} + \widetilde{b}_{p_z}^j \cdot z_{k-p_z+i} + d_k^j \ , \quad (4)$$

where z_{k-n+i} are estimates or measured values of the disturbance. If future values of the disturbance cannot be estimated then it is reasonable to assume that it does not change after the current instant (it is done so in the next part of the paper); $d_k^j = y_k^j - \hat{y}_k^j$ is the DMC–type disturbance model. It means that it is assumed to be the same at each sampling instant in the prediction horizon.

It should be however stressed that despite the disturbances are assumed to be constant on the whole prediction horizon, their values will be updated in the next sampling instant.

Assuming that the manipulated variable can change only once during the prediction horizon, (4) can be transformed into the following form:

$$\begin{aligned}\hat{y}^j_{k+i|k} = &\sum_{m=1}^{n_u}\left(\sum_{n=i+1}^{p_d-1}\widetilde{a}^{j,m}_n\cdot\Delta u^m_{k-n+i}+\widetilde{a}^{j,m}_{p_d}\cdot\sum_{n=p_d}^{p_d+i-1}\Delta u^m_{k-n+i}-\sum_{n=1}^{p_d-1}\widetilde{a}^{j,m}_n\cdot\Delta u^m_{k-n}\right)\\ &+y^j_k+\sum_{n=i+1}^{p_z-1}\widetilde{b}^j_n\cdot\Delta z_{k-n+i}+\widetilde{b}^j_{p_z}\cdot\sum_{n=p_z}^{p_z+i-1}\Delta z_{k-n+i}-\sum_{n=1}^{p_z-1}\widetilde{b}^j_n\cdot\Delta z_{k-n}\\ &+\sum_{m=1}^{n_u}\widetilde{a}^{j,m}_i\cdot\Delta u^m_{k|k}=\widetilde{y}^j_{k+i|k}+\sum_{m=1}^{n_u}\widetilde{a}^{j,m}_i\cdot\Delta u^m_{k|k}\ ,\end{aligned}\tag{5}$$

where only the last component depends on future changes of manipulated variables $\Delta u^m_{k|k}$. Other components, grouped in $\widetilde{y}^j_{k+i|k}$, depend only on values of the input signals from the past.

The prediction can be expressed in a vector–matrix form as:

$$\boldsymbol{y}=\widetilde{\boldsymbol{y}}+\boldsymbol{A}\cdot\Delta\boldsymbol{u}\ ,\tag{6}$$

where $\boldsymbol{y}=\left[\boldsymbol{y}^1,\ldots,\boldsymbol{y}^{n_y}\right]^T$, $\boldsymbol{y}^j=\left[y^j_{k+1|k},\ldots,y^j_{k+p|k}\right]$ is a vector of predicted values of output variables, $\widetilde{\boldsymbol{y}}=\left[\widetilde{\boldsymbol{y}}^1,\ldots,\widetilde{\boldsymbol{y}}^{n_y}\right]^T$, $\widetilde{\boldsymbol{y}}^j=\left[\widetilde{y}^j_{k+1|k},\ldots,\widetilde{y}^j_{k+p|k}\right]$ is called a free response of the plant because it contains future output values calculated assuming that the control signal does not change in the prediction horizon. $\boldsymbol{A}$ is the dynamic matrix composed of the step response coefficients:

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \ldots & \boldsymbol{A}_{1n_u}\\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} & \ldots & \boldsymbol{A}_{2n_u}\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{A}_{n_y1} & \boldsymbol{A}_{n_y2} & \ldots & \boldsymbol{A}_{n_yn_u}\end{bmatrix}\ ,\tag{7}$$

where $\boldsymbol{A}_{jm}=\left[\widetilde{a}^{j,m}_1\ \widetilde{a}^{j,m}_2\ \ldots\ \widetilde{a}^{j,m}_p\right]^T$. Thus, the optimization problem (1) can be written in the following form:

$$\min_{\Delta\boldsymbol{u}}\left\{(\overline{\boldsymbol{y}}-\widetilde{\boldsymbol{y}}-\boldsymbol{A}\cdot\Delta\boldsymbol{u})^T\cdot\boldsymbol{\kappa}\cdot(\overline{\boldsymbol{y}}-\widetilde{\boldsymbol{y}}-\boldsymbol{A}\cdot\Delta\boldsymbol{u})+\Delta\boldsymbol{u}^T\cdot\boldsymbol{\Lambda}\cdot\Delta\boldsymbol{u}\right\}\ ,\tag{8}$$

where $\overline{\boldsymbol{y}}=\left[\overline{\boldsymbol{y}}^1,\ldots,\overline{\boldsymbol{y}}^{n_y}\right]^T$, $\overline{\boldsymbol{y}}^j=\left[\overline{y}^j_k,\ldots,\overline{y}^j_k\right]$, $\boldsymbol{\kappa}=\left[\boldsymbol{\kappa}_1,\ldots,\boldsymbol{\kappa}_{n_y}\right]\cdot\boldsymbol{I}$, $\boldsymbol{\kappa}_j=[\kappa_j,\ldots,\kappa_j]$, $\boldsymbol{\lambda}=[\lambda_1,\ldots,\lambda_{n_u}]\cdot\boldsymbol{I}$ are vectors and matrices of appropriate dimensions, $\boldsymbol{I}$ is the identity matrix.

The performance function in the problem (8) depends quadratically on decision variables $\Delta\boldsymbol{u}$. Thus, if the problem without constraints is considered, the vector minimizing this performance function is given by the following formula:

$$\Delta\boldsymbol{u}=\left(\boldsymbol{A}^T\cdot\boldsymbol{\kappa}\cdot\boldsymbol{A}+\boldsymbol{\lambda}\right)^{-1}\cdot\boldsymbol{A}^T\cdot\boldsymbol{\kappa}\cdot(\overline{\boldsymbol{y}}-\widetilde{\boldsymbol{y}})\ .\tag{9}$$

In the simplest case of a SISO plant, when $n_y = 1$ and $n_u = 1$, without loss of generality one can assume that $\kappa = 1$. Then, the change in the manipulated variable can be obtained using the following simple and easy to calculate formula [6]:

$$\Delta \boldsymbol{u}_{k|k} = \frac{\sum_{i=1}^{p} \widetilde{a}_i \cdot \left(\overline{y}_k - \widetilde{y}_{k+i|k}\right)}{\sum_{i=1}^{p} \left(\widetilde{a}_i\right)^2 + \lambda} \quad . \tag{10}$$

Most of MIMO (Multiple Input Multiple Output) control systems described in the literature are systems designed for processes with two manipulated inputs and two outputs. Let us now consider this case. Let

$$\boldsymbol{K} = \boldsymbol{A}^T \cdot \boldsymbol{\kappa} \cdot \boldsymbol{A} + \boldsymbol{\lambda} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \quad . \tag{11}$$

Then, the values of manipulated variables can be calculated using the following formula:

$$\begin{bmatrix} \Delta u^1_{k|k} \\ \Delta u^2_{k|k} \end{bmatrix} = \frac{1}{k_{11} \cdot k_{22} - k_{12} \cdot k_{21}} \cdot \begin{bmatrix} k_{22} & -k_{12} \\ -k_{21} & k_{11} \end{bmatrix} \cdot \boldsymbol{A}^T \cdot \boldsymbol{\kappa} \cdot (\overline{\boldsymbol{y}} - \widetilde{\boldsymbol{y}}) \quad ; \tag{12}$$

more details one can find in [6].

Remark 1. Calculation of the control signal in two cases considered above is very simple. It consists in making a number of basic arithmetic operations. Inclusion of the disturbance measurement in the controllers does not influence the simplicity of the approach significantly. It is because, the model the algorithm is based on is extended appropriately and then only the free response is changed (contains additional components dependent on the estimated disturbances).

Remark 2. For systems with more inputs and outputs the solution (9) can be obtained using a numerical procedure to inverse the matrix $\boldsymbol{K}$. In such a case, however, one can take into consideration application of numerical predictive control algorithms [4,5].

Remark 3. If one needs to take the control signal constraints into consideration then a mechanism of control projection on constraint set can be applied. The mechanism is simple and consists in application of the following rules of modification of increments of manipulated variables:

- for changes of the manipulated variables:
  - if $\Delta u^j_{k|k} < \Delta u^j_{\min}$, then $\Delta u^j_{k|k} = \Delta u^j_{\min}$,
  - if $\Delta u^j_{k|k} > \Delta u^j_{\max}$, then $\Delta u^j_{k|k} = \Delta u^j_{\max}$;
- for values of the manipulated variables:
  - if $u^j_{k-1} + \Delta u^j_{k|k} < u^j_{\min}$, then $\Delta u^j_{k|k} = u^j_{\min} - u^j_{k-1}$,
  - if $u^j_{k-1} + \Delta u^j_{k|k} > u^j_{\max}$, then $\Delta u^j_{k|k} = u^j_{\max} - u^j_{k-1}$.

3 Simulation Experiments

The algorithms with disturbance measurement mechanism were tested in the control system of a nonlinear plant with delay. The control plant is a distillation

column and is described by the Hammerstein model (designed at the Institute of Control and Computation Engineering jointly with specialists from the Institute of Industrial Chemistry). It means that the linear dynamic block is preceded by the nonlinear static block. Structure of the model is shown in Fig. 1. (More information about Hammerstein models one can find, e.g. in [2].)

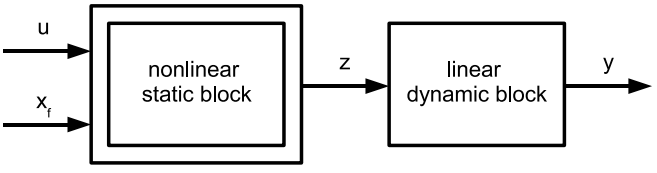


Fig. 1. Structure of the Hammerstein model of the distillation column; symbols are detailed in the text

In the model, the output variable y is the impurity of the distillation product (counted in ppm). The manipulated variable u is the reflux to product ratio (the higher it is the purer product is obtained). The disturbance variable x_f is feed composition; z is the output of the static block and input of the linear dynamic block.

The static characteristics of the plant are shown in Fig. 2. These characteristics were modeled using a polynomial of the fifth order.

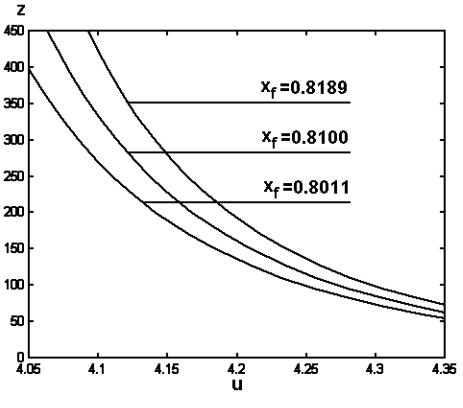


Fig. 2. Static characteristics of the distillation column

The transfer function describing the linear part of the model is as follows:

$$G(s) = \frac{e^{-80s}}{150s + 1} \quad , \tag{13}$$

where time constants are given in minutes.

In order to design the FDMC controller, a Takagi–Sugeno model ([2]) was obtained; a sampling time $T_s = 40$ min was assumed. The step responses were collected near three operating points; the assumed membership functions are shown in Fig. 3.

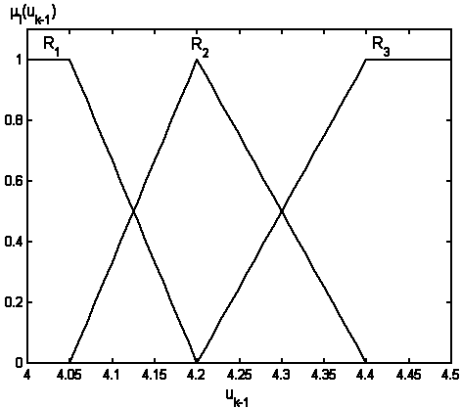


Fig. 3. Membership functions of the control plant model

The example results are shown in Fig. 4. The assumed values of parameters of the FDMC controller are: prediction horizon $p = 22$ and $\lambda = 2 \cdot 10^7$. Responses to the change of the set–point value to $\overline{y}_1 = 300$ ppm and to $\overline{y}_2 = 400$ ppm can be observed during the first 1000 minutes. The character of both responses is the same. Thus, the nonlinear FDMC controller used during the test manifests its advantages.

In the 1000^{th} minute the disturbance variable x_f changed from $x_{f0} = 0.81$ to $x_{f1} = 0.82$. In these conditions the mechanism of disturbance measurement utilization was tested. It was assumed that the change of the disturbance is detected during 40 minutes. It is a reasonable assumption because composition measurement should be possible even in shorter time. After application of the mechanism the output value changes much less (solid lines in Fig. 4) comparing to the case when the mechanism was not used (dotted lines in Fig. 4). It can be also noticed in the control signal that thanks to using the disturbance measurement the controller reacts much earlier (80 minutes earlier) to the change of the disturbance. It illustrates that the proposed mechanism is very useful especially in the case of control plants with large delays.

There was also made another experiment (Fig. 5). The same change of the disturbance was made as in the first experiment but also failure of the sensor was simulated. It was assumed that the sensor brakes down in the 800^{th} minute and indicates all the time the same value. In such a case, change of the x_f disturbance is not compensated at all (dotted lines in Fig. 5). In the case, when the mechanism of disturbance measurement utilization was applied (solid lines in Fig. 5), the disturbance is compensated to a large extent. It is not compensated completely because of modeling inaccuracy. It should be, however, noticed that

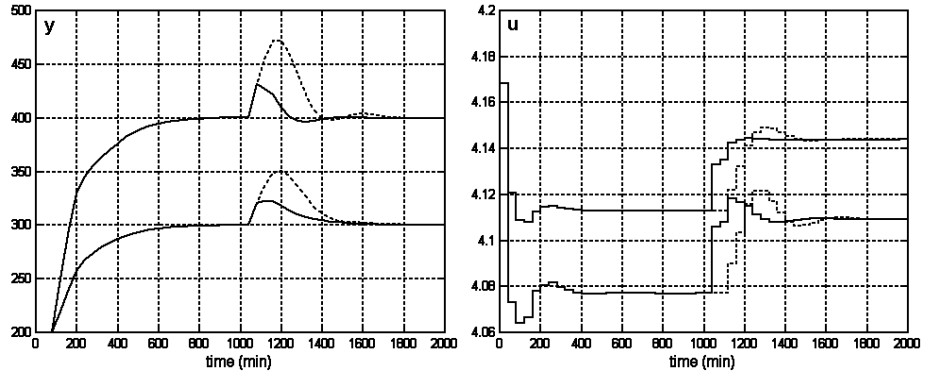


Fig. 4. Responses of the control system with the FDMC controller to the changes of the set–point value to $\overline{y}_1 = 300$ ppm and $\overline{y}_2 = 400$ ppm and to the change of the disturbance x_f in the 1000^{th} minute; disturbance not measured (dotted lines), measured with delay of 40 minutes (solid lines); right – output signal, left – control signal

the result is surprisingly good taking into account delay of the measurement and that it was assumed that the sensor failure is undetected (no reconfiguration is done).

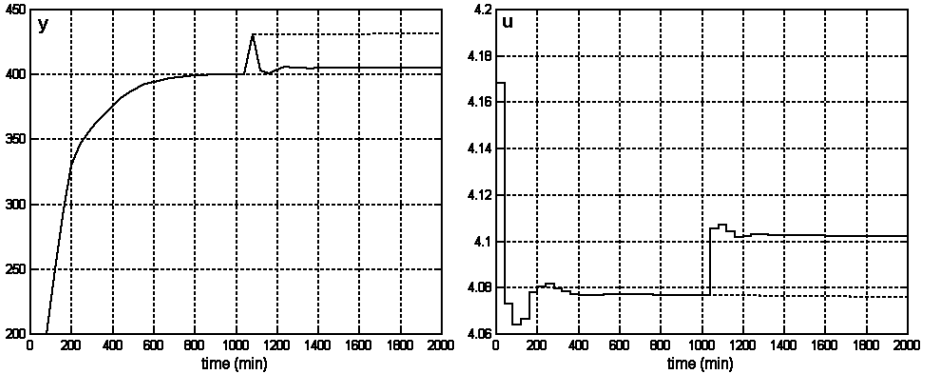


Fig. 5. Responses of the control system with FDMC controller to the change of the set–point value to $\overline{y}_2 = 400$ ppm and to the change of the disturbance x_f in the 1000^{th} minute, after sensor failure; disturbance not measured (dotted lines), measured with delay of 40 minutes (solid lines); right – output signal, left – control signal

4 Summary

The mechanism of disturbance measurement utilization added to the easily reconfigurable predictive controllers is discussed in the paper. The controllers were modified in such a way that their simplicity is maintained. Thanks to the method

of their formulation the control law is easily obtained at each iteration. Using an example, it is shown that the discussed mechanism of disturbance measurement utilization, though simple, can bring significant improvement of control performance. The mechanism is of crucial importance when a sensor fault occurs in the system. Then, at least one of the control loops is broken and quality of the model becomes especially important. Thus it is advisable to use a nonlinear (e.g. fuzzy) process model instead of a linear one and fully exploit all available information about disturbances.

Acknowledgment. This work was supported by the Polish national budget funds for science 2009–2011.

References

1. Camacho, E.F., Bordons, C.: Model Predictive Control. Springer, Heidelberg (1999)
2. Janczak, A.: Identification of nonlinear systems using neural networks and polynomial models: a block–oriented approach. Springer, Heidelberg (2005)
3. Maciejowski, J.M.: Predictive control with constraints. Prentice Hall, Harlow (2002)
4. Marusak, P., Tatjewski, P.: Effective dual–mode fuzzy DMC algorithms with on–line quadratic optimization and guaranteed stability. International Journal of Applied Mathematics and Computer Science 19, 127–141 (2009)
5. Marusak, P.: Advantages of an easy to design fuzzy predictive algorithm in control systems of nonlinear chemical reactors. Applied Soft Computing 9, 1111–1125 (2009)
6. Marusak, P.: Easily reconfigurable analytical fuzzy predictive controllers: actuator faults handling. In: Kang, L., Cai, Z., Yan, X., Liu, Y. (eds.) ISICA 2008. LNCS, vol. 5370, pp. 396–405. Springer, Heidelberg (2008)
7. Piegat, A.: Fuzzy Modeling and Control. Physica-Verlag, Berlin (2001)
8. Rossiter, J.A.: Model-Based Predictive Control. CRC Press, Boca Raton (2003)
9. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modeling and control. IEEE Trans. Systems, Man and Cybernetics 15, 116–132 (1985)
10. Tatjewski, P.: Advanced Control of Industrial Processes; Structures and Algorithms. Springer, London (2007)

Biometric-Based Authentication System Using Rough Set Theory

Hala S. Own[1], Waheeda Al-Mayyan[2], and Hussein Zedan[2]

[1] Department of Solar and Space Research,
National Research Institute of Astronomy18. and Geophysics
Cairo, Egypt
halaown@gmail.com
http://www.medicalnewstoday/articles/51455.php
[2] Software Technology Research Lab, De Montfort University
The Gateway, Leicester LE1 9BH, England, UK
{walmayyan,hzedan}@dmu.ac.uk

Abstract. In this paper we have proposed a biometric-based authentication system based on rough set theory. The system employed signature for authentication purpose. The major functional blocks of the proposed system are presented. Information is extracted as time functions of various dynamic properties of the signatures. We apply our methodology to global features extracted from a 108-users database. Thirty-one features were identified and extracted from each signature. Rough set approach has resulted in a reduced set of nine features that were found to capture the essential characteristics required for signature identification. Low error rates obtained in experiments illustrate the feasibility of using Rough Set as a promising technique for online signature identification systems.

Keywords: Biometric, Rough Set, Online Signature Identification, Global Features, Naïve Bayes.

1 Introduction

With the rapid progress in application areas such as enterprise wide network security infrastructures, government IDs, secure electronic banking and investing, health and social services, the need for highly secure identification and personal verification technologies is becoming apparent[18].

Biometrics are automated methods of authenticating an individual's identity based upon physical or behavioural characteristics. The aim of such systems is to differentiate between the characteristics and behaviours of each person and thus help to identify a person immediately [4]. Identification of physiological traits is based on the measurement of certain parts of the body; amongst other that are used as working tools are fingerprints, facial factions, iris, geometry of the hand, DNA or retina [19]. However, The behavioural identification include certain activities of the person such as, voice, handwriting, signature, walking gait and the manners of using mouse or keys a keyboard [15].

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 560–569, 2010.

The two most widely used approaches for signature identification are offline and online. In Offline signature recognition the presence of the user is not necessary as it compares the various characteristics of a pre-recorded signature image in order to reach to a correct decision. On the other hand, online approach utilizes a number of parameters associated with the stylus and electronic writing pad for determining the authenticity of the signature [1],[3]. These parameters include speed, direction, pressure of the stylus, number order of the strokes, etc.

Research on online signature-based biometric has been ongoing since 1977 due to the improvement carried out in acquisition techniques. Yanikoglu and Kholmatov [16] have used the Dynamic Time Warping (DTW) to align signatures based on two local features (Δx and Δy). Afterward, three reference set is calculated with respect to that user's training set. Next, Principle Component Analysis (PCA) is performed to decorrelate the three distances and classify on this last measure. Nevertheless, there is still two main drawbacks of using DTW; namely heavy computational overhead and that the resampling process usually include the lost of important local details so that at the end forged signatures are closely matched with the genuine ones. Khan et al. [8] proposed a new interesting stroke-based algorithm that splits velocity signal into various bands. Based on these bands, strokes are extracted which are smaller and simpler in nature. Training of the proposed system revealed that low- and high-velocity bands of the signal are unstable, whereas the medium-velocity band can be used for discrimination purposes. Euclidean distances of strokes extracted on the basis of medium-velocity band are used for verification purpose. The experiments conducted show improvement in discriminative capability of the proposed stroke-based system with an Equal Error Rate (ERR) of 2.39% with a database of 15000 signatures gathered from 25 signers, each signer was asked to contribute with 600 signatures.

Nanni and Lumini [12] proposed an online signature verification system based on local information and on a one-class classifier; the Linear Programming Descriptor classifier (LPD). The information was extracted as time functions of various dynamic properties of the signatures, then the discrete 1-D wavelet transform (WT) was performed on these features. The Discrete Cosine Transform (DCT) was used to reduce the approximation coefficients vector obtained by WT to a feature vector of a given dimension. The Linear Programming Descriptor classifier was trained using a little subset of the DCT coefficients. Smaneh and Mohsen [5] introduce a new method based on image registration, discrete wavelet transform and image fusion for identification and verification of Persian signatures.

Rough set theory, a relatively new mathematical theory which was introduced by Pawlak in early 1980s [14] [15]. It has quickly gained popularity in the field of artificial intelligence, robotics and uncertainty management. Its usefulness is apparent in handling knowledge.

Rough Set theory is very useful, especially in handling imprecise data and extracting relevant patterns from crude data for proper utilization of knowledge.

In this paper we introduce a rough set as a new methodology in signature identification system. Rough Set theory was used to improve the classification performance of signature identification systems as well as a dimensionality reduction technique to discover the most information rich feature from our data set.

This paper is organized as follows. Section 2 gives a brief introduction to rough sets. Section 3 discusses our proposed online signature authentication system in detail. Experimentation is covered in Section 4 including data preparation and its characteristic, analysis, results and discussion of the results and finally, conclusions are provided in Section 5.

2 Rough Sets: Basic Notation

The original rough set theory was proposed by Pawlak [14] [15]. This theory is concerned with the analysis of deterministic data dependencies.

Definition 1 (Information System). Information system is a tuple (U, A), where U consists of objects and A consists of features. Every a ∈ A corresponds to the function $a : U \rightarrow V_a$ where V_a is the value set of a. In the applications, we often distinguish between conditional features C and decision feature D, where C ∩ D = ϕ. In such cases, we define decision systems (U, C, D).

Definition 2 (Lower and Upper Approximation). In rough sets theory, the approximation of sets is introduced to deal with inconsistency. A rough set approximates traditional sets using a pair of sets named the lower and upper approximation of the set. Given a set $B \subseteq A$, the lower and upper approximations of a set $Y \subseteq U$ are defined by, respectively,

$$\underline{B}Y = \bigcup_{x:[x]_B \subseteq X} [x]_B , \tag{1}$$

$$\overline{B}Y = \bigcup_{x:[x]_{B \cap X \neq \Phi}} [x]_B \tag{2}$$

The positive region of X is defined as:

$$POS_C(D) = \bigcup_{X:X \in U / \overline{Ind}_D} \underline{C}X \tag{3}$$

$POS_C(D)$ is the set of all objects in U that can be uniquely classified by elementary sets in the partition U/Ind_D by means of C [6]. The negative region $NEG_C(D)$ is defined by:

$$NEG_C(D) = U - \bigcup_{X:X \in U / Indo} \overline{C}X \tag{4}$$

is the set of all objects can be definitely ruled out as member of X. The boundary region is the difference between upper and lower approximations of a set X that consists of equivalence classes having one or more elements in common with X; it is given by the following formula:

$$BND_B(X) = \underline{B}X - \overline{B}X \ . \tag{5}$$

Definition 3 (Degree of Dependency). Given a decision system, the degree of dependency of D on C can be defined as:

$$\gamma(C,D) = \left|POS_C(D)\right| / \left|U\right|, \tag{6}$$

A reduct is a subset $R \subseteq C$ such that

$$\gamma(C,D) = \gamma(R,D). \tag{7}$$

The reduct set is a minimal subset of attributes that preserves the degree of dependency of decision attributes on full condition attributes. The intersection of all the relative reduct sets is called core.

3 The Proposed Online Signature Authentication System

In our proposed system, Topaz's IdGem 1x5 is used for acquisition purpose, which is a non sensitive pressure tablet with a visual feedback with an LCD screen that gives the signer a natural feeling of signing on ordinary paper. At each sample point, we obtain the (x,y) trajectories (x[n], y[n]), n = 1, . . . ,Ns, where Ns is the number of samples of the signature of the signature trajectory along with the time stamp and number of pen-ups are all recorded. Figure 1 show the block diagram of our proposed online signature authentication system which comprises of three main phases. Follows will be introducing a detailed description of the model.

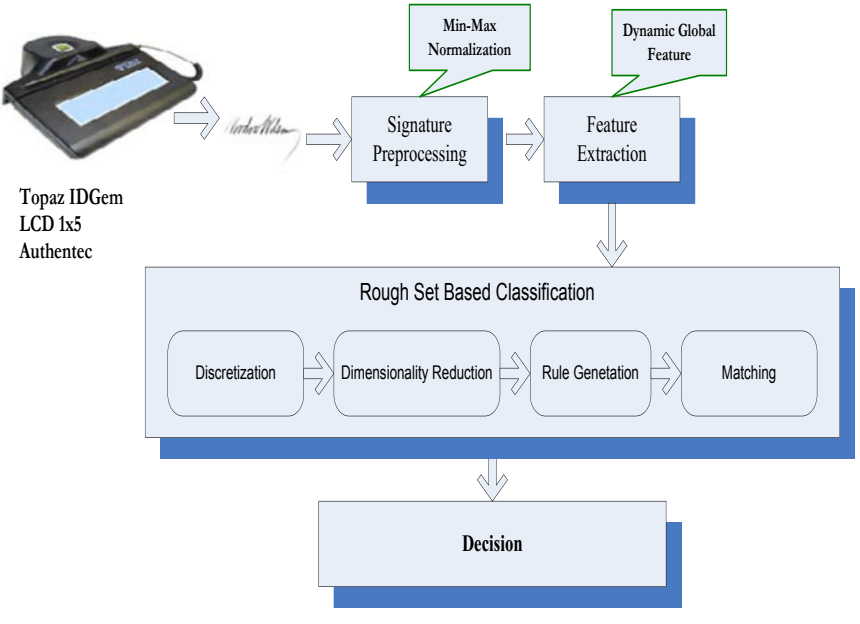


Fig. 1. The overall framework of the proposed system

3.1 Preprocessing Phase

Preprocessing operations may include normalization with respect to size, placement and orientation, re-sampling and smoothing of signature. Since the captured signatures typically have different dynamic ranges, we have adapted a simple approach to minimize this range with respect to the maximum and minimum values [11]. Min-max normalization is the simplest of the score normalization techniques. The normalization shifts the minimum and maximum scores to range between 0 and 1, respectively, thus this normalization does not change the underlying distribution of the data except for a scaling factor. This is performed as shown in the following equation [10]:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}, \; y' = \frac{y - \min(y)}{\max(y) - \min(y)} \tag{8}$$

3.2 Feature Extraction Phase

Our interest is to find the most reliable and suitable set of dynamic features to be used in our approach, so we decided to consider global features. Table 1 shows a list of 31 global features that we have used in this paper. They represent a collection of some of the statistical features that have been widely used, studied, and reported in literature [2] [6] [9] [12] [13].

Table 1. Set of the 31 global features and their description

Feature		Description
1.	SNx	Mean of all normalized coordinates in X direction
2.	SNy	Mean of all normalized coordinates in Y direction
3.	Smax	Number of times the pen was lifted over the entire signature.
4.	Svx	Mean of velocity over all coordinates in X direction
5.	Svy	Mean of velocity over all coordinates in Y direction
6.	Sax	Mean of acceleration over all coordinates in X direction
7.	Say	Mean of acceleration over all coordinates in Y direction
8.	SR	Rhythm or the speed of pen tracing out the signature]
9.	RMSvx	Root mean square of velocity in X direction
10.	RMSvy	Root mean square of velocity in Y direction
11.	RMSax	Root mean square of acceleration in X direction
12.	RMSay	Root mean square of acceleration in Y direction
13.	MaxAx	Maximum acceleration in X direction
14.	MaxAy	Maximum acceleration in Y direction
15.	MaxVx	Maximum velocity in X direction
16.	MaxVy	Maximum velocity in Y direction
17.	R	Correlation co-efficient
18.	Zvx	Sign changes within velocity in X direction]
19.	Zvy	Sign changes within velocity in Y direction
20.	Zax	Sign changes within acceleration in X direction]
21.	Zay	Sign changes within acceleration in Y direction]
22.	xAz	Number of zeroes in acceleration in X direction

Table 1. (*Continued*)

Feature	Description
23. yAz	Number of zeroes in acceleration in Y direction
24. Savxy	Root mean square of (x,y) coordinates
25. Npoints	Number of x,y within signature
26. Sdvx	Standard deviation of velocity in X direction
27. Sdvy	Standard deviation of velocity in Y direction
28. Sdax	Standard deviation of acceleration in X direction
29. Sday	Standard deviation of acceleration in Y direction
30. Dx	Sum of changes between each consecutive points within X-coordinate (signature path horizontal length: total displacement in the X direction)
31. Dy	Sum of changes between each consecutive points within Y-coordinate (signature path vertical length: total displacement in the Y direction)

3.3 Rough Set Based Classification Integrated Approach

In this section, we will discuss in details the proposed rough set algorithm to analyze signature identification dataset. Our algorithm proposed here consists of set of stages. These stages leading towards the final goal of making identification through classification from information or decision system of the signature dataset. Rosetta rough set package [20] was used in the implementation of rough set techniques through all the experimental. The main steps of the rough set based classification integrated approach are provided below.

Rough Set Based Classification Integrated Approach

Input: A database of 2160 signatures from 108 subjects signature.
Output: A confusion matrix represents the classification accuracy.

1- **Data Discretization:** Data discretiziation (or as in machine learning also referred to as discretization) is a procedure that takes a data set and converts all continuous attributes to categorical. In other words, it discretizes the continuous attributes. We adopt the rough sets with Boolean reasoning (RSBR) algorithm proposed by Zhong et al. [17] for the discretiziation of continuous-valued attributes.

2- Attribute Reduction: The reduction technique used is the dynamic reduct. The process of computing dynamic reduct can be seen as a combining normal reduct computation with re-sampling techniques [6].

3- Rule Generation: The generated reducts are used to generate decision rules. The decision rule, at its left side, is a combination of values of attributes such that the set of (almost) all objects matching this combination have the decision value given at the rule's rough side [14].

4- **Classification**: The rule derived from reducts can be used to classify the data. The set of rules is referred to as a classifier and can be used to classify new and unseen data [16].

4 Database and Experiments

4.1 Database

We started by building our own database to form the nucleus of a local database. It contains 2160 signatures gathered from108 different volunteer subjects. Among those subjects, 60 are females and two are left-handed. Each subject was asked to contribute 20 signatures collected in two sessions that were held two to four weeks apart. Ten signatures were collected from each subject during each session.

There were no constraints on how to sign, so the subjects signed in their most natural way; in an arbitrary orientation. Therefore there was a significant intra-class deformation and variation among signatures that belong to the same subject.

4.2 Experiments

Six experiments were conducted to evaluate the classifiers as well as the features. For all of the following experiments 65% split were used for training and the remaining 35% for testing.

The first four experiments were conducted using the 20 signatures from each subject for a total of 2160 signatures.

- **Experiment 1**: In this experiment, all 31 features shown in Table 3 were used with the Naïve Bayes classifier. The correct classification rate achieved was **97.1%**.
- **Experiment 2**: One of the main objectives of this paper is to find the most effective set of dynamic features to be used in describing signatures.. Principal Component Analysis (PCA) [7] was used for feature reduction. Using the Naïve Bayes classifier with the 15 coefficients resulting from PCA resulted in **94%** classification accuracy.
- **Experiment 3**: In this experiment, the rough set approach was used to find the minimal reduct set of features. This has resulted in the following 9 feature numbers: {1,2,3,4,5,8,10,11,30} from Table 3. They correspond to the following features: {SNx, SNy, sMax, SVx, SVy, SR, RMSVy, RMSAx, ,Dx }. Using this features set with the Naïve Bayes classifier resulted in classification accuracy of 96.3**.** The following Table 2 shows some statistics of the above minimal reduct set.

Table 2. Statistics of minimal reduct set

Feature	Mean	Standard Deviation	Correlation
1	0.52	0.077	0.174
2	0.512	0.086	0.117
4	4.176	2.67	0.088
5	-0.0004	00.000327	-0.224
6	0.000297	-0.00035	-0.095
9	0.719	0.386	-0.008
11	0.011	0.004	0.181
12	0.002	0.001	-0.087
30	2520.437	1621.87	0.044

- **Experiment 4:** In this experiment, the rough set classifier was used with the minimal reduct set containing the 9 features. The classification accuracy achieved was **98.5%**.

It was mentioned in section 4.1 that the signatures were collected over 2 sessions two to four weeks apart. In each session, 10 signatures were collected from each of the 108 subjects. In the following 2 experiments we test each session separately.

- **Experiment 5:** Only the 10 signatures taken from each subject during the first signing session is considered. This results in a total of 1080 signatures. Using the Rough set classifier with minimal reduct set of 9 features resulted in **100%** classification accuracy.
- **Experiment 6:** Only the 10 signatures taken from each subject during the second signing session is considered. This results in a total of 1080 signatures. Using the Rough set classifier with minimal reduct set of 9 features resulted in **99%** classification accuracy.

The difference in results between the two signing sessions as shown in Experiments 5 and 6 was expected. The subjects themselves were less enthusiastic in completing the second signing session when they were approached few weeks later. This has resulted in higher intra-variations in the signatures than those of the first session.

Table 3 shows the summary of the results of the six experiments carried above. It clearly demonstrates the suitability and superiority of using the proposed Rough Set approach for both feature reduction and classification in online signature identification.

Table 3. Summary of Results

EXP#	Total number of Signature	Number of Signatures per subject	Feature Reduction Technique	Classifier	Accuracy
1	2160	20	No	Naïve Bayes	97.1
2	2160	20	PCA	Naïve Bayes	94
3	2160	20	Rough	Naïve Bayes	96.3
4	2160	20	Rough	Rough	98.5
5	1080	10	Rough	Rough	100
6	1080	10	Rough	Rough	99

5 Conclusion

The research presented here has demonstrated the success of using the proposed Rough set approach in feature reduction and classification of online signatures. A local database of 2160 signatures from 108 subjects was built. Thirty-one global features were identified and extracted. Different feature reduction methods such as PCA and Rough sets were tested. This resulted in a minimal set of nine features.

Classification using Naïve Bayes and Rough set classifiers was performed. The reported results from several experiments confirmed the effectiveness of the proposed method.

Acknowledgment. This research was supported by The Public Authority for Applied Education and Training of Kuwait under grant number BS-06-13.

References

1. Baansal, A., Gupta, B., Khandelwal, G., Chakraverty, S.: Offline Signature Verification Using Critical Region Matching. International Journal of Signal, Image Processing and Pattern 2(1) (2009)
2. Dessimoz, R.D., Champod, J.C., Drygajlo, A.: Multimodal Biometrics for Identity Documents (MBioID): State-of-the-Art (Version 2.0). Research Report PFS 341-08.05 (2006)
3. Fierrez-Aguilar, J., Ortega-Garcia, J., Gonzalez-Rodriguez, J.: Target dependent score normalization techniques and their application to signature verification. In: Zhang, D., Jain, A.K. (eds.) ICBA 2004. LNCS, vol. 3072, pp. 498–504. Springer, Heidelberg (2004)
4. Fiérrez-Aguilar, J., Nanni, L., Lopez-Peñalba, J., Ortega-Garcia, J., Maltoni, D.: An On-Line Signature Verification System Based on Fusion of Local and Global Information. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 523–532. Springer, Heidelberg (2005)
5. Ghandali, S., Ebrahimi, M.: Off-Line Persian Identification and Verification Based On Image Registration And Fusion. Journal of Multimedia 3(3) (2009)
6. Hassanien, A.E., Own, H.: Rough Sets for Prostate Patient Analysis. In: Proceedings of International Conference on Modeling and Simulation (MS 2006), Malaysia (2006)
7. Jain, A.K., Friederike, D.G., Scott, D.C.: Online signature verification. Pattern Recognition 35(12), 2963–2972 (2002)
8. Khan, M.A.U., Niazi, M.K.K., Khan, M.A.: Velocity-Image Model for Online Signature Verification. IEEE Transactions on Image Processing 15(11), 3540–3549 (2006)
9. Kiran, G.V., Kunte, R.S.R., Samuel, S.: On Line Signature Verification System Using Probabilistic Feature Modelling. In: International Symposium on Signal Processing and its Application (ISSPA), Kuala Lumpur, Malaysia, pp. 355–358 (2001)
10. Krawczyk, S.: User authentication using on-line signature and speech. Master's thesis, Michigan State University, Dep. of Computer Science and Engineering (2005)
11. Lei, H., Govindaraju, V.: A Study on the Consistency of Features for On-Line Signature Verification. In: SSPR/SPR, vol. 444 (2004)
12. Nanni, L., Lumini, A.: A Novel Local On-Line Signature Verification System. Pattern Recognition Letters 29(5), 559–568 (2008)
13. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
14. Pawlak, Z.: Rough Sets- Theoretical aspect of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
15. Pawlak, Z., Grzymala-Busse, Z., Slowinski, J.R., Ziarko, W.: Rough sets. Communications of the ACM 38(11), 89–95 (1995)

16. Yanikoglu, B., Kholmatov, A.: An improved decision criterion for genuine forgery classification in on-line signature verification. In: Kaynak, O., Alpaydın, E., Oja, E., Xu, L. (eds.) ICANN 2003 and ICONIP 2003. LNCS, vol. 2714. Springer, Heidelberg (2003)
17. Zhong, N., Skowron, A.: Rough sets based Knowledge Discovery Process. Int. J. Appl. Math. Comput. Sci. 11(3), 603–619 (2001)
18. Xiang, W., Desai, B., Wen, P., Wang, Y., Peng, T.: A Prototype Biometric Security Authentication System Based upon Fingerprint Recognition. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS, vol. 5589, pp. 264–272. Springer, Heidelberg (2009)
19. http://www.medicalnewstoday/articles/51455.php
20. http://www.lcb.uu.se/tools/rosetta/

Classification of Facial Photograph Sorting Performance Based on Verbal Descriptions

Daryl H. Hepting[1], Richard Spring[1], Timothy Maciag[1], Katherine Arbuthnott[2], and Dominik Ślęzak[3,4]

[1] Department of Computer Science, University of Regina
3737 Wascana Parkway, Regina, SK, S4S 0A2 Canada
{dhh,spring1r,maciagt}@cs.uregina.ca
[2] Campion College, University of Regina
3737 Wascana Parkway, Regina, SK, S4S 0A2 Canada
katherine.arbuthnott@uregina.ca
[3] Institute of Mathematics, University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
[4] Infobright Inc.
Krzywickiego 34 pok. 219, 02-078 Warsaw, Poland
slezak@infobright.com

Abstract. Eyewitness identification remains an important element in judicial proceedings. It is very convincing, yet it is not very accurate. To better understand eyewitness identification, we began by examining how people understand similarity. This paper reports on analysis of study that examined how people made similarity judgements amongst a variety of facial photographs: participants were presented with a randomly ordered set of photos, with equal numbers of Caucasian (C) and First Nations (F), which they sorted based on their individual assessment of similarity. The number of piles made by the participants was not restricted. After sorting was complete, each participant was then asked to label each pile with a description of the pile's contents. Following the results of an earlier study, we hypothesize that individuals may be using different strategies to assess similarity between photos. In this analysis, we attempt to use the descriptive pile labels (in particular, related to lips and ears) as a means to uncover differences in strategies for which a classifier can be built, using the rough set attribute reduction methodology. In particular, we aim to identify those pairs of photographs that may be the key for verifying an individual's abilities and strategies when recognizing faces. The paper describes the method for data processing that enabled the comparisons based on labels. Continued success with the same technique as previously reported to filter pairs before performing the rough sets analysis, lends credibility to its use as a general method. The rough set techniques enable the identification of the sets of photograph pairs that are key to the divisions based on various strategies. This may lead to a practical test for people's abilities, as well as to inferring what discriminations people use in face recognition.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 570–579, 2010.

1 Introduction

Eyewitness identification holds a prominent role in many judicial settings, yet it is generally not accurate. Verbal overshadowing [1] is an effect that can obscure a witness's recollection of face when he is asked to describe the face to create a composite sketch. Alternatively, if the witness is asked to examine a large collection of photos, her memory may become saturated and she may mistakenly judge the current face similar to another she has examined (i.e., inaccurate source monitoring) and not to the one she is trying to recall [2]. We hypothesize that if the presentation of images can be personalized, the eyewitness may have to deal with fewer images, minimizing both of the negative effects discussed. This research takes more steps along that path.

This paper discusses an analysis of data from a sorting study, which avoided verbalization completely while sorting. Each participant was asked to group a stack of 356 photos according to perceived similarity. One half of the photos ($n = 178$) depicted Caucasian males, taken in the southern United States of America. The other half of the photos depicted First Nations males, taken at different locales in the Canadian province of Saskatchewan. 'First Nations' is the term which has replaced 'Indian' in most cases. In Saskatchewan, there are 72 First Nations [1] governments or bands. As a participant encountered a photo, she could only place that photo and not disturb any existing piles. Indirectly, each participant made 63,190 pairwise similarity judgements. Once sorting was complete, each participant was asked to verbally label each pile according to the similarity used to create that pile. In this paper, we examine whether the occurrence of a label may be a good indicator of sorting performance.

We discuss the extension of previously published methods [3] by allowing the classification of facial photograph sorting performance based on verbal descriptions. The earlier work examined whether race (Caucasian/First Nations) had any impact on facial photograph sorting performance, which is also of interest because of the existence of a "cross-race" effect [4] which may make identifications of faces more difficult if those faces are not of the same race as the viewer.

Furthermore, we present several more successful examples of the filtering technique used to substantially reduce the processing time and effort needed to build a completely accurate classifier, if one exists.

Section 2 describes the method of making piles based on the presence or absence of a label. Section 3 describes the filtering technique developed to reduce the number of photo pairs needed as input to the attribute reduction methodology, and the results obtained for "ears/not-ears" and "lips/not-lips" label decision classes. Section 4 shows how to combine results from the previous study with those first reported here, to make a more complete test of participant performance. Section 5 presents conclusions and avenues for future work.

[1] source: http://fsin.com

Table 1. Labels for facial parts, listed from the top of head downwards, followed by general characteristic labels. Notice that many labels are used for every picture. This analysis only looked at the presence or absence of a label, so "ears" and "lips" were chosen (44%). We sought labels that were used for approximately 50% of the photos, because we required an equal number for which the label was not used. In this case, we randomly selected 157, of the 199, for which the labels were not used in order to perform our analysis. For the parts that were identified in all photos, such as "hair", "eyes", "head/face shape", or "skin/complexion", we might be able to use them to distinguish photos (as in "big head" compared to "small head"), but we did not record the data in this way.

Label	Photos	Percentage
hair	356	100
forehead	65	18
eyebrows	217	61
ears	157	44
nose	259	73
eyes	356	100
cheeks	25	7
lips	157	44
teeth	14	4
jaw/chin	318	89
neck	25	7
head/face shape	356	100
head/face size	125	35
skin/complexion	356	100
facial hair	243	68

2 Analysis of Verbal Descriptions

After sorting all 356 photos, all participants were asked to describe with a label the similarity embodied in each pile that they had created. The label or labels attached to each pile were then assigned to each photo with the pile. This process was repeated for all 25 participants. Table 1 shows the unique occurrences of various labels with photos.

The two labels occurring with approximately 50% of the photos ("ears" and "lips") were chosen for further analysis, following the procedure outlined in Hepting et al. [3]:

1. choose $N = 157$ of the photos to which the label was not attached (from 199 possible). Therefore, the label and not-label sets each have 157 photos
2. for each participant, exclude from the pile data all 42 of the photos not in the label/not-label sets
3. analyze the make-up of each pile, in terms of label (L) and not-label (NL) photos (only these photos remain in the pile). Our null hypothesis (H_0) is that each pile comprises the same proportion of L and NL photos. Using the CHITEST function in Microsoft Excel, we test the independence of the

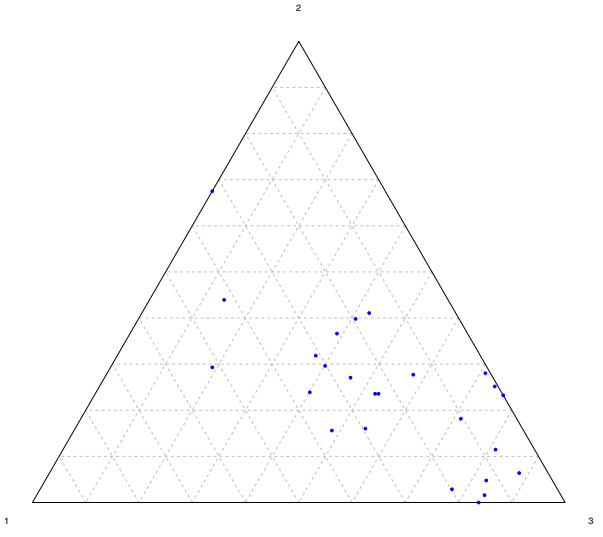


Fig. 1. Each point reflects the mix of photos classified by a participant. A point in the center of the triangle represents an equal mix of photos classified L (label), NL (not-label), and U (undecided). This figure shows the distribution of participants based on their classification of photos with respect to the "ears" label. Many points are located between Vertex 1 and Vertex 2, representing approximately equal numbers of photos classified as "ears" and "not-ears". If a point is near Vertex 3, that participant classified most of the photos as "Undecided". For the "ears" label, we constructed 2 decision classes, "uses-ears" and "uses-not-ears", based on the percentage of "undecided" photos. Participants with more than 60% "undecided" photos were put into "uses-not-ears" decision class and the others were put into "uses-ears".

observed ratio of L to NL (as a percentage) against an expected equal ratio (50%:50%). If p (returned from CHITEST) < 0.05, we rejected H_0 and either classified the pile as L, if L >NL or as NL if NL > L. The pile was classified as U (for undecided) if $p \geq 0.05$ (and we could not reject H_0). All pictures in that pile were then labelled as L, NL, or U. The total number of photos classified as L, NL, and U was expressed as a percentage (see Figure 1).

Figure 1 shows all participants plotted according to their percentages of photos classified as L, NL, and U for the "ears" label. Vertex 3 represents undecided (U) and points near this vertex represent participants who classified most photos as U. A threshold of 60% was set for the percentage of U and two groups were formed. We hypothesize that these groups correspond to different strategies for facial recognition, which we have labelled as "uses-ears" (U $< 60\%$, $n = 15$) and "uses-not-ears" (U $\geq 60\%$, $n = 10$). In other words, we hypothesize that "ears" is being used by former group but not by the latter. In the same way for "lips", a threshold of 60% was set for the percentage of U and two groups were

formed. We hypothesize that these groups correspond to different strategies for facial recognition, which we have labelled as "uses-lips" (U $< 60\%$, $n = 9$) and "uses-not-lips" (U $\geq 60\%$, $n = 16$). In other words, we hypothesize that "lips" is being used by former group but not by the latter.

We seek to find a simple way to classify participants according to these groups, which will allow for personalization of the eyewitness identification process. The strategy (uses-ears or uses-not-ears, uses-lips or uses-not-lips) then becomes the decision variable as we begin to apply the rough set attribute reduction methodology [5]. The objective is to reduce the number of pairs required as input to discriminate between the two strategies, as the original number of pairs is impractical.

3 Pair Filtering

For each participant, a decision is made (directly or indirectly) about whether a pair of photos is similar (same pile) or not (different piles). 63,190 pairs can be formed from the 356 photos used in this study, which is a very large input to the analysis stage. Thus, we have pursued a method to reduce the number of input pairs to the analysis stage, based on the following hypothesis (also discussed in Hepting et al. [3]): the pairs most useful in constructing reducts and rules will be those which are rated most differently between the decision classes, similar to the feature extraction/selection phase in knowledge discovery and data mining.

We used the following method to test the hypothesis: we compute the total distance for a pair within each decision class by normalizing the sum of all participant ratings. If all participants in the same decision class rate the pair as similar, the distance is 0. If all participants in the same decision class rate the pair as different, the distance is 1. In general, the distance is computed as the sum of similarity ratings (each one is either 0 or 1) divided by the number of participants in the decision class. We first look at the minimum of these two distances, $d = min(D_1, D_2)$, in order to find a pair that is rated as very similar by participants in one of the decision classes. If a pair is rated as very similar by participants in both decision classes (both D_1 and D_2 are small), that pair will not help to discriminate between the decision classes. Therefore, we also look at the gap between the two distances, $\Delta = |D_1 - D_2|$. The pairs which have a small d and a large Δ are those which meet the criterion of being rated most differently between the decision classes. Table 2 shows the collection of these values for ears and lips. The row values indicate the minimum distance (d) and the column values indicate the gap (Δ).

We used the Rough Set Exploration System (RSES) [6] to analyse the sets indicated by this filtering. We proceed through each table in Table 2 row by row from the top left to the bottom right, until a classifier with 100% accuracy and 100% coverage is found. Our procedure is outlined in the following:

Table 2. The values in the table indicate the number of pairs selected by each combination of minimum distance (row) and gap (column). The set of pairs used for further processing is indicated in bold. On the left, results of filtering for uses-ears/ uses-not-ears. On the right, results of filtering for uses-lips / uses-not-lips. One pair of photos from each set is illustrated Figure 2. The input to RSES (Rough Set Exploration System) [6] for uses-lips/uses-not-lips is shown in Table 3.

	Gap (Δ)				
d	≥ 0.9	≥ 0.8	≥ 0.7	≥ 0.6	≥ 0.5
≤ 0.1	2	**9**	31	59	108
≤ 0.2	2	22	77	250	398
≤ 0.3	2	22	159	498	1246
≤ 0.4	2	22	159	675	1883
≤ 0.5	2	22	159	675	2314

	Gap (Δ)				
d	≥ 0.9	≥ 0.8	≥ 0.7	≥ 0.6	≥ 0.5
≤ 0.1	0	0	0	2	2
≤ 0.2	0	3	14	**16**	46
≤ 0.3	0	3	24	78	179
≤ 0.4	0	3	24	166	762
≤ 0.5	0	3	24	166	1356

Fig. 2. On the left, one of the pairs of photos important in the classification of participants according to uses-ears/uses-not-ears. On the right, one of the pairs of photos important in the classification of participants according to uses-lips/uses-not-lips.

1. Split: Split input file (50/50): Each file in the analysis was split with 50% of participants in a training set (data from 12 participants) and 50% of participant's data (data from 13 participants) in a testing set. The files comprised objects each representing a pairwise comparison of facial photographs (0 if similar, 1 if dissimilar). The decision class was the strategy (either uses-ears/uses-not-ears (illustrated in Figure 1) or uses-lips/uses-not-lips).
2. Train: Calculate the reducts in training file using genetic algorithms in RSES. The genetic algorithms procedure calculates the top N reducts possible for a given analysis. For the purposes of our analysis, we chose N = 10 in order to pick the top 10 reducts possible (if indeed 10 top reducts could be found). Generate rules from these reducts.
3. Classify: Classify the 25 participants according to the generated rules, and observe the accuracy and coverage of the classifier.

We conducted k-fold cross-validation [7], with k = 10. If a classifier with 100% accuracy and 100% coverage is not found within 10 tries, it may still exist. Choosing 13 of 25 participants for training leads to a possible $\binom{25}{13} = 5,200,300$ combinations and classifiers.

Table 3. 16 pairs selected as input to RSES (Rough Set Exploration System) for classification based on uses-lips/uses-not-lips ($d = 0.2$, $\Delta = 0.6$)

	Photograph Pairs																
object	1969a-2094a	4211a-5893a	2660a-8127a	1296a-2811a	3722a-4158a	2094a-6682a	058-149	083-117	1716a-7001a	040-108	1032a-1867a	0011a-7453a	1296a-6682a	5241a-8164a	0079a-6524a	1032a-8831a	class
O:1	1	0	0	1	0	1	0	1	1	0	0	1	0	0	1	0	0
O:2	0	0	0	0	0	1	0	1	1	0	0	0	1	0	1	0	1
O:3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O:4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O:5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
O:6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
O:7	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
O:8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
O:9	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0
O:10	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
O:11	0	1	1	1	1	1	0	0	0	0	0	0	1	1	1	0	0
O:12	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
O:13	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
O:14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
O:15	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1
O:16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
O:17	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
O:18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O:19	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0
O:20	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
O:21	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
O:22	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0	0
O:23	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0
O:24	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
O:25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 4. The distribution of the 25 participants between groups identified by combinations of strategies based on the apparent use of race, ears, and lips in their decision making

Group	Members
uses-not-race, uses-not-ears, uses-not-lips	4
uses-not-race, uses-not-ears, uses-lips	1
uses-not-race, uses-ears, uses-not-lips	2
uses-not-race, uses-ears, uses-lips	4
uses-race, uses-not-ears, uses-not-lips	4
uses-race, uses-not-ears, uses-lips	1
uses-race, uses-ears, uses-not-lips	6
uses-race, uses-ears, uses-lips	3

Table 5. The values in the table indicate the number of pairs selected by each combination of minimum distance (row) and gap (column). The set of pairs used for further processing is indicated in bold. On the left, results of filtering for uses-race,uses-ears, uses-not-lips / not(uses-race,uses-ears, uses-not-lips). On the right, results of filtering for race/not-race. The 7 pairs chosen for race here are different than those used in Hepting et al. [3], but they were selected according to the method outlined here.

	Gap (Δ)				
Dist.	≥ 0.9	≥ 0.8	≥ 0.7	≥ 0.6	≥ 0.5
≤ 0.1	0	**10**	49	107	149
≤ 0.2	0	12	90	389	852
≤ 0.3	0	12	90	389	853
≤ 0.4	0	12	90	478	1486
≤ 0.5	0	12	90	478	1590

	Gap (Δ)				
Dist.	≥ 0.9	≥ 0.8	≥ 0.7	≥ 0.6	≥ 0.5
≤ 0.1	0	0	0	1	2
≤ 0.2	0	0	0	**7**	11
≤ 0.3	0	0	17	82	197
≤ 0.4	0	0	17	130	401
≤ 0.5	0	0	17	130	798

Table 6. Accuracy (A) and Coverage (C) for each classifier over 10 trials (with mean and standard deviation following each). FA/FC indicates the number out of the 10 trials that had 100% accuracy and 100% coverage. This is followed by the pairs used to classify according to each strategy. Photos ending in 'a' are Caucasian, others are First Nations. None of the pairs is mixed. No pair repeats, though some individual photos are included with more than 1 pair. 4 First Nations and 3 Caucasian photos are used as input for the race strategy classification. For the ears strategy classification, almost all are First Nations photos, whereas for the lips and the combined strategy classifications, almost all the photos are Caucasian.

Race	Ears	Lips	Combined
A(92.38, SD : 4.38) C(99.60, 1.26) FA/FC: 1	A(99.20, SD : 1.69) C(100, SD : 0) FA/FC: 8	A(94.40, SD : 3.86) C(100, SD : 0) FA/FC: 3	A(97.20, SD : 2.70) C(97.60, 7.59) FA/FC: 3
004-050	033-121	0011a-7453a	0576a-8530a
039-125	037-176	0079a-6524a	062-178
050-176	038-068	040-108	1338a-6553a
0662a-4919a	058-157	058-149	1513a-1859a
087-142	095-106	083-117	1907a-9929a
2325a-8650a	111-121	1032a-1867a	4099a-4459a
6281a-9265a	146-172	1032a-8831a	4099a-6553a
	152-153	1296a-2811a	4488a-6553a
	4833a-9948a	1296a-6682a	6838a-8922a
		1716a-7001a	7297a-9860a
		1969a-2094a	
		2094a-6682a	
		2660a-8127a	
		3722a-4158a	
		4211a-5893a	
		5241a-8164a	

4 Combination

We made groups based on strategies: uses-race/uses-not-race [3], uses-ears/uses-not-ears, and uses-lips/uses-not-lips. We found that the largest of these groups was uses-race, uses-ears, uses-not-lips. The distribution of the 25 participants between groups is shown in Table 4. Table 5 shows the relationship between the minimum distance and gap for this largest group in Table 4.

Table 6 presents a comparison of the classifiers discussed, based separately on different strategies identified (uses-race/uses-not-race, uses-ears/users-not-ears, and uses-lips/uses-not-lips) and on a combined strategy uses-race AND uses-ears AND uses-not-lips/NOT(uses-race AND uses-ears AND uses-not-lips) as identified in Table 4. The uses-race/uses-not-race classifier has been recomputed from the earlier paper [3], according to the algorithm described here. It is interesting to note that the average accuracy seems to be related to the first non-zero entry in the table of filtered pairs. Table 6 shows that in order of most to least accurate (with the position of the first non-zero entry, from the top-left in Tables 2 and 5, in parentheses), we have: ears (1), combined (2), lips (4), and race(4).

5 Conclusions and Future Work

Cross-race identification of faces is an important topic of ongoing research [8], and our sorting study seeks to contribute to this body of work. We have focused on the labelling of similarity judgements as a way to understand the way people perceive structure in the stimuli set.

Through this effort, we have found succinct tests to classify people into different strategy groups (ears/not-ears, lips/not-lips). Namely, we demonstrated that rough sets can help in accuracy and clarity of the results. It is interesting that the decision table for "ears" comprises almost exclusively First Nations pairs, and the decision table for "lips" comprises almost exclusively Caucasian pairs. Neither has any mixed pairs. Therefore, we hope that these results will help in our efforts to better understand the cross-race effect [4].

This work lends support to our filtering technique as a broadly applicable method. In general, all the classifiers have performed well, but the one based on the "ears" label is clearly the best among them. We still need to understand what strategy might be at work in these cases, but the accuracy of the classifier indicates a clear difference between the decision classes. Although we have not done any sort of exhaustive testing of all pairs to verify our selection criteria for filtering, that we have been able to generate consistently accurate classifiers from very small fractions of the total pairs is a very encouraging sign.

We have a test to classify participants. Further work will be devoted to validating it against the performance on eyewitness identification tasks, and to using it to help clarify the strategies being employed by participants.

Acknowledgements. The first four authors were supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada. The fifth

author was supported by the grants N N516 368334 and N N516 077837 from the Ministry of Science and Higher Education of the Republic of Poland.

References

1. Schooler, J.W., Ohlsson, S., Brooks, K.: Thoughts beyond words: when language overshadows insight. Journal of Experimental Psychology: General 122, 166–183 (1993)
2. Dysart, J., Lindsay, R., Hammond, R., Dupuis, P.: Mug shot exposure prior to lineup identification: Interference, transference, and commitment effects. Journal of Applied Psychology 86(6), 1280–1284 (2002)
3. Hepting, D.H., Maciag, T., Spring, R., Arbuthnott, K., Ślęzak, D.: A rough sets approach for personalized support of face recognition. In: Sakai, H., Chakraborty, M.K., Hassanien, A.E., Ślęzak, D., Zhu, W. (eds.) RSFDGrC 2009. LNCS, vol. 5908, pp. 201–208. Springer, Heidelberg (2009)
4. Jackiw, L.B., Arbuthnott, K.D., Pfeifer, J.E., Marcon, J.L., Meissner, C.A.: Examining the cross-race effect in lineup identification using Caucasian and First Nations samples. Canadian Journal of Behavioural Science/Revue canadienne des sciences du comportement 40(1), 52–57 (2008)
5. Pawlak, Z.: Rough set approach to knowledge-based decision support. European Journal of Operational Research 99, 48–57 (1997)
6. Bazan, J.G., 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)
7. Maciag, T., Hepting, D.H., Hilderman, R.J., Ślęzak, D.: Evaluation of a dominance-based rough set approach to interface design. In: Proceedings of the FBIT 2007 International Conference, pp. 409–416 (2007)
8. Platz, S., Hosch, H.: Cross-racial/ethnic eyewitness identification: A field study 1. Journal of Applied Social Psychology 18(11), 972–984 (1988)

Random Musical Bands Playing in Random Forests

Miron B. Kursa[1], Elżbieta Kubera[2,3],
Witold R. Rudnicki[1], and Alicja A. Wieczorkowska[3]

[1] Interdisciplinary Centre for Mathematical and Computational Modelling,
University of Warsaw, Pawińskiego 5A, 02-106 Warsaw, Poland
{M.Kursa,W.Rudnicki}@icm.edu.pl
[2] University of Life Sciences in Lublin, Akademicka 13, 20-950 Lublin, Poland
elzbieta.kubera@up.lublin.pl
[3] Polish-Japanese Institute of Information Technology,
Koszykowa 86, 02-008 Warsaw, Poland
alicja@poljap.edu.pl

Abstract. In this paper we investigate the problem of recognizing the full set of instruments playing in a sound mix. Random mixes of 2-5 instruments (out of 14) were created and parameterized to obtain experimental data. Sound samples were taken from 3 audio data sets. For classification purposes, we used a battery of one-instrument sensitive random forest classifiers, and obtained quite good results.

1 Introduction

Automatic recognition of instruments playing in a given musical recording is an interesting and ambitious problem, which has been studied intensively using various methods. In the past, our team has contributed several studies to this area. In particular, we have developed a methodology for recognition of a predominant sound in musical mixes [12,20]. In the current paper we are interested in much more difficult problem – identification of all instruments playing in mixes generated using up to five instruments. We apply an extension of the methodology developed for the earlier problem [12]. The outcomes of this results can be applied in automatic labeling and content-based searching of audio data (in order to find pieces with given instruments playing), as well as in aiding automatic music transcription – notes identified through pitch tracking can be then attributed to particular instruments, thus helping to extract the score [11].

Research on automatic identification of instruments in audio data was first performed on isolated monophonic (monotimbral) sounds, with successful application of k-nearest neighbors, artificial neural networks, rough-set based classifiers [22], support vector machines (SVM – see e.g. [6,7]). Also, research was next performed on polyphonic (polytimbral) data, when more than one instrument were playing at the same time [5,10]. In this case, separation of these sounds from the audio source can be attempted [5]. Interested reader can find results of the research on polytimbral instrumental data in [2,9,14,19,20,23].

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 580–589, 2010.

Various scientists utilized different data sets: of different number of classes (instruments and/or articulation), different number of objects/sounds in each class, and basically different feature sets, so the results are quite difficult to compare. Still, the recognition of instruments in monophonic recordings can reach 100% for a small number of classes, more than 90% if the instrument or articulation family is identified, or about 70% or less for recognition of an instrument when there are more classes to recognize. The identification of instruments in polytimbral mixes is usually lower, even below 50% for same-pitch sounds; more details can be found in the paper describing our previous work [21]. Generally, recognition for monotimbral data is much easier, in particular for isolated sounds, than for polytimbral data.

Random Forest (RF) is a classifier which comprises of a set of weak, weakly correlated and non-biased classifiers – decision trees. It has been shown that in many cases RF performs equally well or better than other methods on a diverse set of problems [1]. In the previous study [12] we have shown that RF is much better suited for recognition of the musical instruments than SVM which is considered as the state-of-the-art method for machine learning applications.

2 Material and Methods

The audio data used in our research originate from 3 repositories, commonly applied in similar works: MUMS [16], the University of IOWA Musical Instrument Samples [18], and RWC [4]. We chose the following instruments: B-flat clarinet, cello, trumpet (C trumpet in MUMS), English horn, flute, French horn, marimba, oboe, piano, tenor trombone, tubular bells, vibraphone, viola, and violin. Octave no. 4 (MIDI notation) of these instruments was used in our experiments (so that the octave number was not discerning in classification). Thus, we limited the amount of data to process, yet the experiments can be further expanded in the future to full musical scale. We choose sustained articulation if possible, e.g. vibrato for bowed strings. Isolated sounds of the instruments were then mixed to create both training and testing data; we used mixes (as preparation to future research on real recordings) to allow automatic labeling. The data were digitally represented using 44.1 kHz sampling rate and 16-bit resolution, stereo; the left channel was (arbitrarily) chosen for further works.

The music samples (mixes) were generated through the following procedure:

1. A number of instruments M in a sample is randomly chosen from 2-5 range.
2. For each instrument, a random representative sound file is selected – randomly among different notes and recordings from 3 different databases.
3. M random numbers are drawn with uniform probability distribution and normalized to have a sum of 1; those will become the weights of volumes of instruments' sounds in the mix.
4. All selected representative sound files are normalized to a globally common RMS (Root Mean Square) level of the audio signal and mixed, with corresponding weights. The resulting file is also normalized and converted to a vector of descriptors stored along with the generated weights.

In this way we obtained a database of mixes, represented by descriptors and annotated with the contribution of each instrument to the total sound varying between 0 and 1. Because we choose up to 5 out of 14 instruments, the average probability of including the instrument in the mix is equal to $\text{mean}(2,3,4,5)/14 = 0.25$, and the distribution of weights has a high peak at 0. Therefore, we present this distribution limited to $(0,1]$, as shown in Fig. 1.

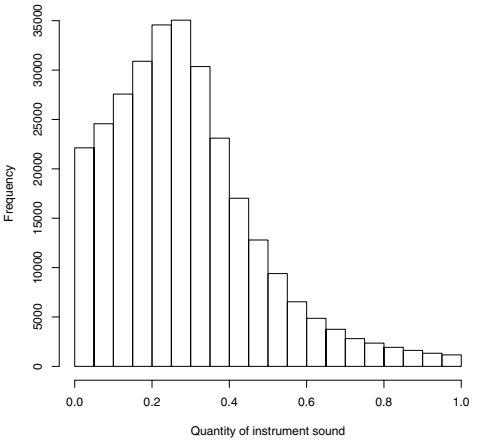


Fig. 1. Histogram of instrument weights that are greater than 0

2.1 Construction of the Descriptors

Since audio data (i.e. sequences of numbers representing quantized amplitude of a waveform) are not well suited to use as input for RF classifiers, sound parametrization was applied first to our audio data. There is no standard feature set used in such research, still, MPEG-7 descriptors are often applied [8], as well as other features describing time domain of sound, spectrum, time-frequency features – based on Fourier or wavelet analysis etc. Other parameters that can be applied include MFCC (Mel-Frequency Cepstral Coefficients), Multidimensional Scaling analysis trajectories of various sound features, statistical properties of spectrum etc., see e.g. [6].

The feature vector we used here is based on a parameterization applied in our previous research [21]. However, relying upon feature importance measures obtained through comparing RF classification on the original and randomized feature values [12], we decided to limit the initial set of 219 features to only 54 on which the accuracy is kept. Our feature vector consists of [12]:

– MPEG-7 based descriptors [8]: *AudioSpectrumSpread*; *AudioSpectrumFlatness* for 25 out of 32 frequency bands; *AudioSpectrumCentroid*; *HarmonicSpectralCentroid*, *HarmonicSpectralSpread*, *HarmonicSpectralVariation*, *HarmonicSpectralDeviation*, *LogAttackTime*, *TemporalCentroid*;

– other features: *Energy*; *MFCC* – min, max, mean, distance (sum of dissimilarity, i.e. absolute difference of values, of every pair of coordinates in the vector), and standard deviation of MFCC vector; *ZeroCrossingRate*; *RollOff*; *Flux*; *FundamentalFrequency*; $r_1, \ldots, r_{11}$ - various ratios of harmonic partials in spectrum: r_1 – energy of the fundamental to the total energy of all harmonics, r_2: amplitude difference [dB] between 1^{st} and 2^{nd} partial, r_3: ratio of the sum of partials 3-4 to all harmonics, r_4: partials 5-7 to all, r_5: partials 8-10 to all, r_6: remaining partials to all, r_7: brightness – gravity center of spectrum, r_8, r_9: contents of even/odd harmonics in spectrum.

Calculation of the parameters was performed using fast Fourier transform, with 120 ms analyzing frame and Hamming window (hop size 40 ms), which allows analysis of the low-pitch sounds in the future (requiring long analysis frame because of long period), even for the lowest audible fundamental frequencies. Also, experiments show that such a long analysis frame yields good results [3,10,21], yet we realize that shorter frame could be used here [15,21].

Most of the features represent average value of frame-based attributes, calculated for consecutive frames of a parameterized sound using sliding analysis window as described above, moved through the entire sound file.

2.2 Random Forest Method

RF is an ensemble of classification trees, constructed using procedure minimizing bias and correlations between individual trees. Each tree is built using different N-element bootstrap sample of the training N-element set; the elements of the sample are drawn with replacement from the original set, so roughly 1/3 of the training data are not used in the bootstrap sample for any given tree.

Let us assume that objects are described by a vector of P attributes (features). At each stage of tree building, i.e. for each node of any particular tree in RF, p attributes out of all P attributes are randomly selected ($p \ll P$, often $p = \sqrt{P}$). The best split on these p attributes is used to split the data in the node. Each tree is grown to the largest extent possible (no pruning). By repeating this randomized procedure M times one obtains a collection of M trees – a random forest. Classification of each object is made by simple voting of all trees.

2.3 Classification

The classification methods were modified in comparison with our previous study [12]. Formerly, a single 14-class classifier was built using RF method. The decision was simple – find the loudest instrument playing in the sample. In the current study the task is more difficult. The sample may contain between 2 and 5 instruments and we want to identify all of them, employing the following strategy.

We create a battery of 14 RF binary classifiers, each specialized in classification of one instrument (vs. others); to train such a classifier we select 2000 samples in which its instrument weight is equal to 0 as negative training objects

and 2000 samples in which its instrument weight is greater than w as positive ones. We have created 6 classifier batteries by applying this procedure for w=0, 0.1, 0.2, 0.3, 0.4 and 0.5. The unused samples from the database formed the test set (about 80,000 objects). When classifying a new object, each RF yielded *yes* or *no* when recognizing or not the corresponding instrument. For each w, the corresponding battery was tested. The result of a battery classification of a sample was a vector of instruments for which the corresponding classifiers from the battery yielded *yes* for this sample. In ideal case, only the classifiers trained to recognize instruments which were present in the sample should yield *yes*, but in reality only some classifiers will recognize instruments correctly; some will fail to recognize the instrument when its indeed present in the sample, and some will yield *yes* while the instrument is absent.

We performed experiments for the six sets as described above, in order to find for which w the best classification is obtained. In this work we used R – an environment for statistical computing [17]. The R package randomForest [13] served as a RF implementation.

3 Results and Discussion

In our previous research, we conducted experiments aiming at the recognition of the predominant sound of a single instrument, mixed with a background composed of all other instruments' sounds (with equal contributions) from the data set. Various levels of background were tested, and the classification error in the worst case (background at 50% level of the main sound) was equal to 10%.

In this research, our goal is to recognize more than one instrument in the recording, even if it is much softer than the other sounds in the mix. The outcomes of this research are presented in this section.

3.1 Classification Results

The recall as the function of the sound intensity for various classifiers is presented in Fig. 2. Obtained results show that for a high intensity of instrument sound (more than 50% of contribution to the total intensity) it is relatively easy to recognize a given instrument. One can also observe that the sensitivity of the classifier is a monotone function of the instrument weight in samples. Moreover, adding training examples of low sound level (lowering w) increases the accuracy. Also, we can see that the false positive rate decreases with increasing w. Finally, it is easily seen that sensitivity of the classifiers is very low for mixes containing sounds with intensity lower than w, whereas it is very similar for all classifiers on the samples with high sound intensity (see bottom right panel of Fig. 2). Detailed recall plots for classifiers developed for all instruments, using samples containing all sound levels for each of our 14 instruments, are shown in Fig. 3.

The balance of the false positive (FP) rate and false negative (FN) rate, as well as overall accuracy of classifiers for all instruments is shown in Table 1 (obtained for the full test-set, so the expected FP rate of a random classifier

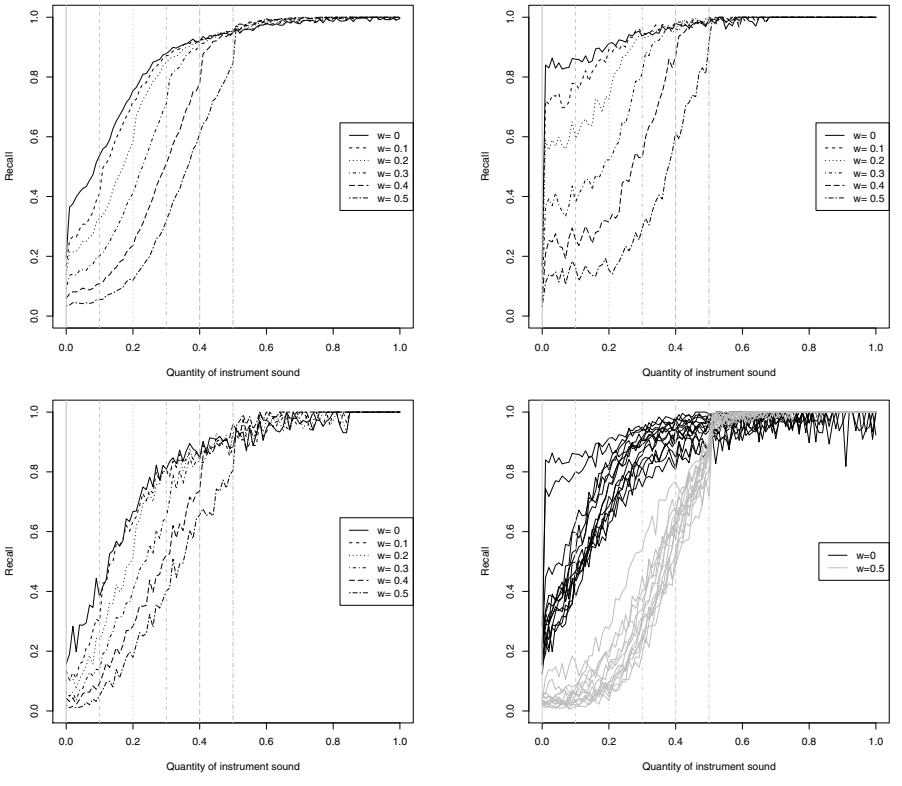


Fig. 2. Recall of binary classifiers trained with various w as a function of the weight of a given instrument in the total mix. *Top left:* the average for the whole battery. *Top right:* the most sensitive classifier (marimba). *Bottom left:* the least sensitive classifier (flute). *Bottom right:* all classifiers for $w = 0$ (black) and $w = 0.5$ (gray). Recall for 0 represents false positive rate (misclassifications).

is 25%, as mentioned before). As we can see, FP rate decreases and FN rate increases with increasing w. The highest accuracy was obtained for the classifier trained on all samples with the sound intensity higher than 0.3 of the total.

We also checked the accuracy on the subsets of test samples in which the target instrument weight is about $1/n$, which corresponds to this instrument playing in an equal $n-$tet, for $n = 2$ (duo), $n = 3$ (trio), $n = 4$ (quartet), and $n = 5$ (quintet). The results are presented in Table 2. As expected, the identification of sounds in duos is much easier than identification of instruments playing in quintets. The other result is a very high difference in performance between classifiers trained on the full set ($w = 0$) and those trained on the subset containing high intensity sounds ($w = 0.5$). Generally, classification quality drops significantly when classifier is used for recognition of sounds weaker than those

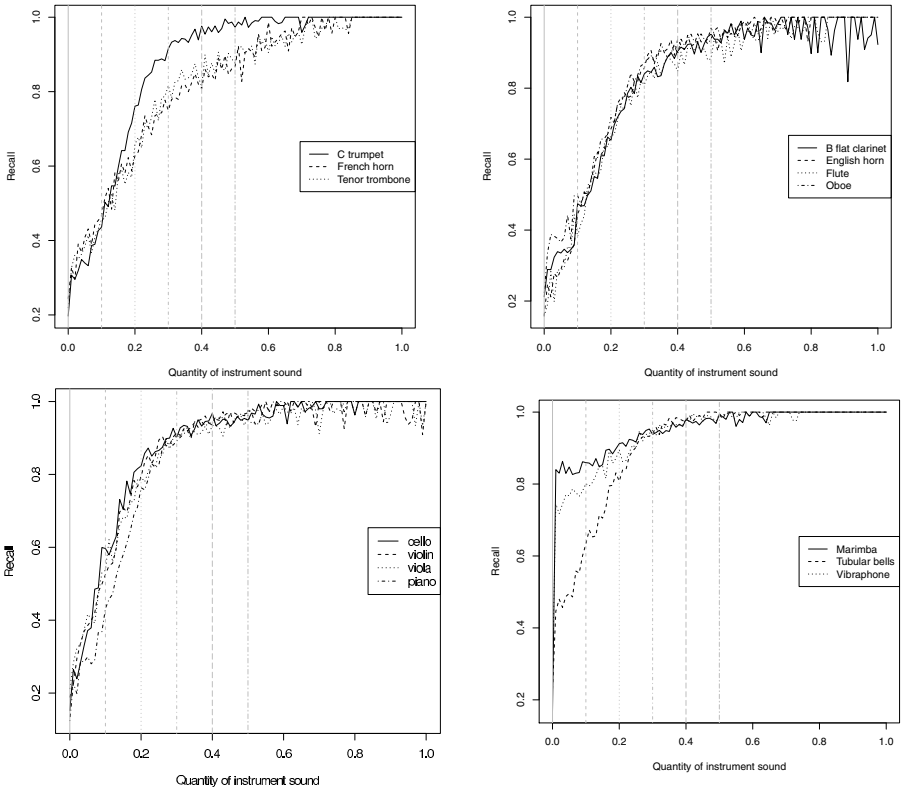


Fig. 3. Recall of binary classifiers trained with $w = 0$ as a function of the weight of a given instrument in the total mix. *Top left:* brass. *Top right:* woodwinds. *Bottom left:* chordophones. *Bottom right:* idiophones.

used for its training. The best results are obtained for the classifier trained on all possible combinations of sound intensities, i.e. $w = 0$. This classifier is able to recognize correctly instruments playing in quintet in more than 75% of cases. We can observe that the selection of the best classifier depends strongly on the task at hand – whether we can afford high false negative or rather false positive rate. For example, Kitahara et al. in [10] ascertain that high precision is more important that high recall in case of the research they performed.

3.2 Instrument Similarity

Since the sounds used in mixes were randomly chosen, we expect that the results of instrument recognition would not be correlated, unless the correlation is caused by misclassification due to e.g. similarity of sounds of particular instruments. In Fig. 4 we may see the result of such analysis – instruments of similar

Table 1. Classification results for the instruments in mixes: 1. B-flat clarinet, 2. cello, 3. trumpet, 4. English horn, 5. flute, 6. French horn, 7. marimba, 8. oboe, 9. piano, 10. tenor trombone, 11. tubular bells, 12. vibraphone, 13. viola, and 14. violin

	Instrument														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	all
$w = 0$															
Accuracy [%]	77.0	83.7	79.6	81.3	80.8	74.1	85.4	76.9	84.7	76.0	78.7	76.3	78.7	80.8	79.6
FP rate [%]	15.9	11.4	14.8	12.0	11.9	18.3	12.6	16.8	9.3	16.5	17.3	21.3	15.9	14.2	14.9
FN rate [%]	7.1	4.9	5.6	6.8	7.3	7.6	2.0	6.3	6.0	7.5	3.9	2.4	5.4	5.1	5.6
$w = 0.3$															
Accuracy [%]	82.4	86.8	83.0	83.3	84.4	77.4	86.7	80.9	85.4	79.9	82.8	81.5	82.4	83.9	82.9
FP rate [%]	5.6	4.9	6.3	4.9	4.6	11.0	5.8	7.8	4.4	8.2	7.3	11.0	8.4	7.3	7.0
FN rate [%]	12.0	8.3	10.7	11.7	10.9	11.6	7.5	11.3	10.2	11.8	10.0	7.5	9.1	8.8	10.1
$w = 0.5$															
Accuracy [%]	80.4	84.9	81.7	81.0	83.6	78.4	81.8	79.8	82.8	79.2	80.7	80.6	81.7	82.3	81.3
FP rate [%]	2.7	2.0	1.9	1.8	1.4	4.5	2.4	2.9	1.2	3.6	1.9	3.9	3.4	2.5	2.6
FN rate [%]	16.9	13.1	16.5	17.2	15.0	17.1	15.8	17.2	16.0	17.2	17.4	15.5	14.9	15.2	16.1

timbre are close in this plot. Marimba, vibraphone, and piano constitute a group in this figure. Indeed, these instruments sound similar, and their sounds have similar temporal envelope: sharp attack, no sustained part, and long offset. Also, violin, viola, and cello can be seen as a group in this graph; these instruments represent a group of bowed strings. Tenor trombone is close to French horn – these instruments represent brass group, so their timbre can also be considered similar. Therefore, we can conclude about similarity between instrument sounds on the basis of observations drawn from results of RF classification.

Table 2. Classification accuracy of recognition of target instruments playing with sound intensity of the target instrument equal to 1/5, 1/4, 1/3, and 1/2 of the total, i.e. quintet, quartet, trio, and duo respectively, for the classifiers for various w

	$w =$					
	0	0.1	0.2	0.3	0.4	0.5
quintet	76.7%	73.1%	64.6%	44.3%	26.4%	13.8%
quartet	84.4%	82.4%	78.7%	60.6%	40.8%	23.5%
trio	90.7%	90.2%	88.8%	84.6%	66.5%	46.9%
duo	94.8%	94.8%	95.2%	95.3%	94.4%	83.9%

From musical point of view, some mistakes in classification can be also caused by similarity of mixed instrument sounds to other instrument sounds – this is often done in arrangement of instruments, since 2 or more instruments playing together create a new sound, sometimes resembling other instrument. Therefore, mixes can be easily misclassified for other instruments because they actually are similar to them from timbral point of view.

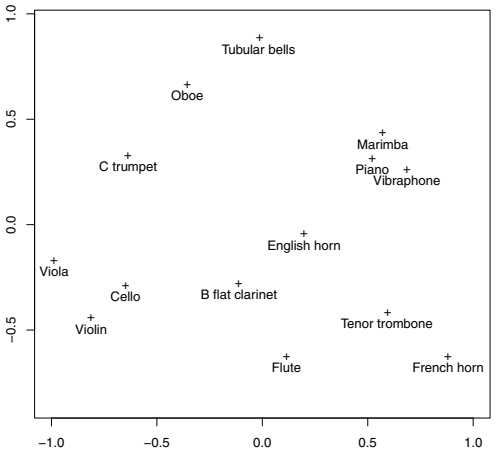


Fig. 4. 2D projection of instrument misclassification rate (i.e. of the 14×14 contingency table). Nearby instruments are more often confused, so can be considered more similar.

4 Summary and Conclusions

In the current study we have applied the RF methodology for the analysis of musical samples obtained by randomly mixing sounds of two up to five instruments. The overall accuracy of sound recognition varies between 80% and 83% in total and between 74% and 87% for individual instruments, depending on the classifier and selection of the test set. The instrument most difficult to recognize correctly was French horn, whereas marimba was easiest to recognize.

The best classification results were obtained for classifiers trained on the entire data set. Apparently, assigning the decision 'instrument present in recording' even if the contribution to the total sound is small improves sensitivity of the classifier more than degrades it.

The most sensitive classifiers (of highest recall) can be used with reasonable accuracy to identify instruments playing in small randomly generated bands; the performance on real recordings is to be tested in the future.

Acknowledgements. This project was partially supported by ICM grant 501-64-13-BST1345, and the Research Center of PJIIT, supported by the Polish National Committee for Scientific Research (KBN). Computations were performed at ICM, grant G34-5.

References

1. Breiman, L.: Random Forests. Machine Learning 45, 5–32 (2001), http://www.stat.berkeley.edu/~breiman/RandomForests/cc_papers.htm
2. Dziubinski, M., Dalka, P., Kostek, B.: Estimation of musical sound separation algorithm effectiveness employing neural networks. J. Intel. Inf. Syst. 24(2-3), 133–157 (2005)

3. Essid, S., Leveau, P., Richard, G., Daudet, L., David, B.: On the usefulness of differentiated transient/steady-state processing in machine recognition of musical instruments. In: AES 118th Convention (May 2005)
4. Goto, M., Hashiguchi, H., Nishimura, T., Oka, R.: RWC Music Database: Music Genre Database and Musical Instrument Sound Database. In: ISMIR, pp. 229–230 (2003)
5. Heittola, T., Klapuri, A., Virtanen, T.: Musical instrument recognition in polyphonic audio using source-filter model for sound separation. In: 10th ISMIR, pp. 327–332 (2009)
6. Herrera, P., Amatriain, X., Batlle, E., Serra, X.: Towards instrument segmentation for music content description: a critical review of instrument classification techniques. In: International Symposium on Music Information Retrieval, ISMIR (2000)
7. Hsu, C.-W., Chang, C.-C., Lin, C.-J.: A Practical Guide to Support Vector Classification, http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf
8. ISO: MPEG-7 Overview, http://www.chiariglione.org/mpeg/
9. Itoyama, K., Goto, M., Komatani, K., Ogata, T., Okuno, H.G.: Instrument Equalizer for Query-By-Example Retrieval: Improving Sound Source Separation Based on Integrated Harmonic and Inharmonic Models. In: 9th ISMIR (2008)
10. Kitahara, T., Goto, M., Komatani, K., Ogata, T., Okuno, H.: Instrogram: Probablilistic Representation of Instrument Existence for Polyphonic Music. IPSJ Journal 48(1), 214–226 (2007)
11. Klapuri, A.: Signal processing methods for the automatic transcription of music. Ph.D. thesis, Tampere University of Technology, Finland (2004)
12. Kursa, M., Rudnicki, W., Wieczorkowska, A., Kubera, E., Kubik-Komar, A.: Musical Instruments in Random Forest. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) ISMIS 2009. LNCS (LNAI), vol. 5722, pp. 281–290. Springer, Heidelberg (2009)
13. Liaw, A., Wiener, M.: Classification and Regression by randomForest. R News 2(3), 18–22 (2002)
14. Little, D., Pardo, B.: Learning Musical Instruments from Mixtures of Audio with Weak Labels. In: 9th ISMIR (2008)
15. Meng, A.: Temporal Feature Integration for Music Organisation. Ph.D. thesis, Lyngby, Denmark (2006)
16. Opolko, F., Wapnick, J.: MUMS – McGill University Master Samples. CD's (1987)
17. R Development Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2009)
18. The University of IOWA Electronic Music Studios: Musical Instrument Samples, http://theremin.music.uiowa.edu/MIS.html
19. Viste, H., Evangelista, G.: Separation of Harmonic Instruments with Overlapping Partials in Multi-Channel Mixtures. In: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2003, New Paltz, NY (2003)
20. Wieczorkowska, A., Kubera, E., Kubik-Komar, A.: Analysis of Recognition of a Musical Instrument in Sound Mixes Using Support Vector Machines. In: Nguyen, H.S., Huynh, V.-N. (eds.) SCKT 2008 Hanoi, Vietnam (PRICAI), pp. 110–121 (2008)
21. Wieczorkowska, A.A., Kubera, E.: Identification of a dominating instrument in polytimbral same-pitch mixes using SVM classifiers with non-linear kernel. J. Intell. Inf. Syst. (2009), doi:10.1007/s10844-009-0098-3
22. Wieczorkowska, A.: Rough Sets as a Tool for Audio Signal Classification. In: Raś, Z.W., Skowron, A. (eds.) ISMIS 1999. LNCS (LNAI), vol. 1609, pp. 367–375. Springer, Heidelberg (1999)
23. Zhang, X.: Cooperative Music Retrieval Based on Automatic Indexing of Music by Instruments and Their Types. Ph.D thesis, Univ. North Carolina, Charlotte (2007)

An Empirical Comparison of Rule Sets Induced by LERS and Probabilistic Rough Classification

Jerzy W. Grzymala-Busse[1,2], Shantan R. Marepally[1], and Yiyu Yao[3]

[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,shantan}@ku.edu
[3] Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada S4S 0A2
yyao@cs.uregina.ca

Abstract. In this paper we present results of an experimental comparison (in terms of an error rate) of rule sets induced by the LERS data mining system with rule sets induced using the probabilistic rough classification (PRC). As follows from our experiments, the performance of LERS (possible rules) is significantly better than the best rule sets induced by PRC with any threshold (two-tailed test, 5% significance level). Additionally, the LERS possible rule approach to rule induction is significantly better than the LERS certain rule approach (two-tailed test, 5% significance level).

1 Introduction

In this paper we present results of an experimental comparison (in terms of an error rate) of rule sets induced using standard rough set theory with rule sets induced by probabilistic rough classification (PRC), a probabilistic extension of Pawlak rough classification [1]. The standard rough set theory approach to rule induction is exemplified by the LERS (Learning from Examples based on Rough Sets) data mining system. In this approach, lower and upper approximations, basic ideas of the standard rough set theory, are first computed. For any concept C, concept approximations are later used for rule induction, e.g., by the MLEM2 (Modified Learning from Examples Module, version 2) rule induction algorithm. Rules induced from lower approximations are called *certain* [2] since they certainly describe the concept C. On the other hand, rules induced from upper approximations are called *possible* since they only possibly describe the concept C. During classification of unseen cases, rule sets induced by MLEM2 are used by the LERS classification system with three parameters [3,4]. There are other systems for rule induction based on rough set theory, see, e.g., [5].

Probabilistic rough set approximations are different from the standard rough set approximations. Two additional parameters are necessary, usually denoted

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 590–599, 2010.

by α and β, which are interpreted as the limits for probabilities, hence the corresponding approximations are called *parameterized*. The parameter α is larger than the parameter β. For any elementary set $[x]$ a conditional probability of the concept C given $[x]$ is computed [6]. The lower parameterized approximation is the union of all elementary sets $[x]$ with the conditional probability larger than α, while the upper parameterized approximation is the union of all elementary sets with the conditional probability greater than β. From these definitions, it is obvious that the only difference between lower and upper parameterized approximations is the choice of the threshold. The two thresholds can be calculated based on the minimization of the overall risk/loss of classification based on the well-known Bayesian decision theory in the decision -theoretic rough set (DTRS) model [7]. The can also be provided by experts as suggested in the variable precision rough set (VPRS) model [8].

As follows from [9], an important feature of the probabilistic rough classification rules is a pair of marginal and conditional probabilities associated with any rule. There is no standard approach to rule induction in probabilistic rough sets. In our experiments, for probabilistic rough classification, we simply used the same rule sets that we used for testing MLEM2, by selecting rules with conditional probabilities satisfying given PRC threshold. Additionally, in the PRC (again, there is no standard approach to classification in probabilistic rough sets) we used the LERS classification system with the partial matching factor set to one since this factor is not only heuristic but it cannot be explained in terms of probability theory.

2 Decision Tables

A decision table (information table) represents input data. An example of such a table is presented in Table 1. Rows of the decision table represent *cases*, columns (except *Case* column) represent *attributes* and a *decision*. The set of all cases is denoted by U. The decision is denoted by d. The set of all attributes is denoted by A. The value for a case x and an attribute a will be denoted by $a(x)$. A *block* of an attribute-value pair (a, v), where $a \in A$ and $v = a(x)$ for some $x \in U$, denoted by $[(a, v)]$ is a set of all cases from U that for a have value v. The set of all attribute values will be denoted by V. Note that Table 1 has 50% of conflicting cases (since four cases: 5, 6, 7 and 8 are involved in conflicts).

3 Basic and Elementary Formulas

In the standard view of concepts, a concept is jointly defined by a pair of an intension and an extension. The extension is a set of cases that are instances of the concept, and the intension is a set of properties, or a formula of a logic language, which defines the concept. One can therefore study concept formation and rule induction based on a set-theoretic setting or a logic setting based on extension and intension of concepts [10]. In the standard rough theory, one typically defines a pair of lower and upper approximations based on extensions. We

Table 1. A decision table

	Attributes			Decision
Case	Temperature	Headache	Cough	Flu
1	normal	no	no	no
2	normal	no	no	no
3	normal	yes	no	yes
4	normal	no	yes	no
5	high	yes	yes	yes
6	high	yes	yes	no
7	high	yes	yes	yes
8	high	yes	yes	yes

will present a formulation based on intensions, which is more convenient for rule induction.

A logic decision language may be defined recursively as follows [6]:

$$\begin{aligned} \text{basic formula}: &\quad (a, v), a \in A, v \in V, \\ \text{formula}: &\quad \text{if } f \text{ and } g \text{ are formulas, so are } f \wedge g \text{ and } f \vee g. \end{aligned}$$

The two operations $\wedge$ and $\vee$ are logic conjunction and disjunction. In a particular decision table, the meanings of formulas are defined recursively as:

$$\begin{aligned} \text{basic formula}: &\quad \|(a, v)\| = \{x \in U \mid a(x) = v\} = [(a, v)], \\ \text{formula}: &\quad \|f \wedge g\| = \|f\| \cap \|g\|, \quad \|f \vee g\| = \|f\| \cup \|g\|. \end{aligned}$$

A basic formula is also commonly expressed as an attribute-value pair (a, v); its meaning $\|(a, v)\|$ is called an attribute-value block [4].

With the introduction of a decision language, it is possible to formally and precisely represent concepts in a decision table. Let us first state a relation between concepts based on the logic implication, which is known as a "more-specific-than" relation [11]. Let f and g be formulas. The formula f is a specialization of g and g is a generalization of f, written $s \rightarrow g$, if any case that satisfies f will satisfy g in any decision table with the same set of attributes of the same domains. In other words, for any decision table m, we have $\|f\|_m \subseteq \|g\|_m$. For example, we have $(a, v_1) \wedge (b, v_2) \rightarrow (a, v_1)$. For a particular decision table, the logical implication always leads to the set inclusion of the extensions of concepts. The reverse is not necessarily true. The inclusion of two sets only suggest that the logic implication may hold between their corresponding logic formulas. In this case, we use the symbol $\Rightarrow$. In rule induction, a main issue to learn a relation $\Rightarrow$ based on extensions of concepts with respect to a particular decision table.

3.1 Global Approach

Let B be a subset of A. One may define an equivalence relation on U based on the values of cases with respect to attributes in B. Two cases are equivalent or discernible if and only if they have the same values on all attributes in B. In terms of the logic language, one can form an *elementary formula* for a case in U:

$$E(x) = \bigwedge_{a \in B} (a, a(x)).$$

The meaning of an elementary formula,

$$\begin{aligned} \|E(x)\| &= \| \bigwedge_{a \in B} (a, a(x)) \| \\ &= \bigcap_{a \in B} \|(a, a(x))\| \\ &= \{y \in U \mid \forall a \in B \ a(x) = a(y)\}, \end{aligned}$$

is the equivalence class containing x. The family $\{\|E(x)\| \mid x \in U\}$ is a partition of the universe.

In rough set theory, equivalence classes $\|E(x)\|$, called *elementary sets*, are the building blocks of approximations of an arbitrary set representing a certain concept. Let E_B denote the set of all elementary formulas defined with respect to a set of attribute $B \subseteq A$. For a decision class $\|(d, u)\|$, (or [(d, u)]) where u is a value of d, we form two sets of elementary formulas as a pair of lower and upper approximations of $[(d, u)]$:

$$\begin{aligned} \underline{apr}[(d, u)] &= \{e \in E_B \mid \|e\| \subseteq \|(d, u)\|\}, \\ \underline{apr}[(d, u)] &= \{e \in E_B \mid (\|e\| \cap \|(d, u)\|) \neq \emptyset\}. \end{aligned}$$

In forming the elementary formulas for rule induction, one needs first to search for a minimal set of attributes with respect to a decision. Such a minimal set is known as a reduct relative to the decision, or simply a relative reduct. Elementary formulas represent the most specific concept in a concept space. Even with a reduct, a disadvantage with the formulation based on the elementary formulas is that one may use two specific concepts in describing a decision class, or equivalently, too small equivalence classes. A solution to this problem is condition pruning in the derived rules [6].

3.2 Probabilistic Rough Classification

Probabilistic rough sets or rough classification [7,12] is a generalization of Pawlak rough sets or rough classification [1]. One may use the conditional probability to describe degree of overlap or inclusion [13] of an equivalence class and a decision class:

$$Pr([(d, u)]|E(x)) = \frac{card([(d, u)] \cap E(x))}{card(E(x))},$$

where *card*() denote the cardinality of a set. A pair of lower and upper parameterized approximations is defined by using a pair of thresholds:

$$\underline{apr}[(d,u)] = \cup\{e \in E_A \mid Pr[(d,u)]|e) \geq \alpha\},$$
$$\underline{apr}[(d,u)] = \cup\{e \in E_A \mid Pr[(d,u)]|e) > \beta\},$$

where $\alpha > \beta$. As follows from the above two formulas, the only important difference between the lower and upper parameterized approximations is the value of the parameters α and β. Hence, in our experiments, we do not distinguish between α and β, calling them a probabilistic classification *threshold*. More detailed discussion on the threshold can be found in references [7,8,12]. For example, in addition to the requirement $\alpha > \beta$, Ziarko [8] suggests the condition $\alpha > Pr([(d,u)]) > \beta$.

3.3 Local Approach

Instead of starting with the elementary formulas, the LERS family starts with the basic formulas defined by an attribute-value pair [2,4,14,15]. Basic formulas represent the most specific concepts without using the logic operator $\wedge$.

Let CF_A denote the set of all formulas formed by attributes from A with only the logic operator $\wedge$. With respect to a decision class $[(d,u)]$, we can define the following pair of lower and upper approximations:

$$\underline{apr'}[(d,u)] = \{s \in CF_A \mid \|s\| \subseteq \|(d,u)\|, \forall s \in CF_A(s \rightarrow g \text{ implies } \|s\| \not\subseteq \|(d,u)\|)\},$$
$$\underline{apr'}[(d,u)] = \underline{apr'}(d,u) \cup \{e \in CF_A \mid (\|e\| \not\subseteq \|(d,u)\|, (\|e\| \cap \|(d,u)\|) \neq \emptyset\}.$$

Although the new lower and upper approximations $\underline{apr'}$ and $\overline{apr'}$ are given by different sets of formulas as $\underline{apr}$ and $\overline{apr}$, their extensions are the same, namely,

$$\| \bigvee \underline{apr'}[(d,u)]\| = \| \bigvee \underline{apr}[(d,u)]\|,$$
$$\| \bigvee \overline{apr'}[(d,u)]\| = \| \bigvee \overline{apr}[(d,u)]\|.$$

4 MLEM2

For our experiments we used the MLEM2 algorithm that explores the search space of attribute-value pairs. The input data set of the MLEM2 is a lower or upper approximation of a concept. The LEM2 computes a *local covering* and then converts it into a rule set [16,17,4,14].

MLEM2 processes numerical attributes differently than symbolic attributes. For numerical attributes MLEM2 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 q MLEM2 creates two blocks, the first block contains all cases for which values of the numerical attribute are smaller than q, the second block contains remaining cases, i.e., all cases for which values of the numerical attribute are larger than q. The search space of MLEM2 is the set of all blocks computed this way, together with blocks defined by symbolic attributes. Finally, MLEM2 simplifies rules by merging intervals for numerical attributes.

Table 2. Error rates, first part

Data set	Percentage of conflicting cases	PRC threshold				
		0.2	0.3	0.4	0.5	0.6
Australian credit	24.35	18.72	18.75	18.62	18.46	18.95
Breast cancer (Slovenia)	4.69	34.58	33.75	34.33	34.75	34.69
Breast cancer (Wisconsin)	5.92	29.96	29.94	29.97	29.85	29.78
Echocardiogram	64.86	34.55	33.42	34.32	33.69	34.19
German credit	29.50	31.43	31.67	31.67	31.76	31.65
Image segmentation	40.95	38.74	38.92	40.09	33.28	35.09
Iris	44.00	11.88	11.75	11.42	11.95	40.95
Primary tumor	18.29	69.29	69.35	69.01	69.12	69.27
Postoperative patient	15.56	44.51	44.33	45.00	44.77	44.33
Wine recognition	38.20	9.92	9.70	9.80	10.43	10.26

5 LERS Classification System

Rule sets, induced from data sets, are used mostly to classify new, unseen cases. A classification system used in LERS is a modification of the well-known bucket brigade algorithm [18,19]. Some classification systems are based on a rule estimate of probability. Other classification systems use a decision list, in which rules are ordered, the first rule that matches the case classifies it [5].

The decision to which concept a case belongs to is made on the basis of two factors: *strength* and *support*. The strength is defined as the total number of cases correctly classified by the rule during training. The second factor, *support*, is defined as the sum strengths for matching rules indicating the same 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

$$\sum_{partially\ matching\ rules\ r\ describing\ C} Matching\_factor(r) * Strength(r)$$

Table 3. Error rates, second part

Data set	PRC threshold				LERS	
	0.7	0.8	0.9	1.0	certain rules	possible rules
Australian credit	19.00	19.65	23.04	71.54	21.20	17.76
Breast cancer (Slovenia)	34.20	35.19	40.10	47.17	29.12	28.45
Breast cancer (Wisconsin)	29.61	29.93	30.08	76.80	20.56	20.68
Echocardiogram	35.22	38.06	68.71	63.38	37.92	31.98
German credit	32.62	35.32	53.03	73.55	29.85	29.55
Image segmentation	40.36	42.04	61.50	63.39	49.47	31.50
Iris	40.88	44.44	47.24	43.44	25.68	24.79
Primary tumor	70.76	73.36	77.22	78.33	66.21	61.15
Postoperative patient	44.74	47.29	62.14	64.00	37.26	38.07
Wine recognition	11.06	11.25	27.95	61.66	32.35	8.88

For Table 1, rules in the LERS format (every rule is equipped with three numbers: the total number of conditions, strength, and the total number of training cases matching the left-hand side of the rule) [4] are:
certain rules, induced from the lower approximations:

1, 3, 3
(Headache, no) -> (Flu, no)
2, 1, 1
(Cough, no) & (Headache, yes) -> (Flu, yes)

and *possible* rules, induced from the upper approximations:

1, 2, 5
(Cough, yes) -> (Flu, no)
1, 3, 3
(Headache, no) -> (Flu, no)
1, 4, 5
(Headache, yes) -> (Flu, yes)

It is not difficult to see that this possible rule set well classifies every case from Table 1 except case 6.

Rules induced by LERS may be easily converted into probabilistic rules defined by PRC. Certain rules are associated with the conditional probability equal to one. For possible rules, the corresponding conditional probability is determined as the ratio of the second number preceding the rule to the third number. For the above possible rules, if the threshold = 0.3, all three rules survive. If the threshold = 0.5, only the last two rules satisfy this new condition. Again, it is not difficult to see that this new rule set, with only two rules, also well classifies all cases from Table 1 except case 6.

In our experiments, for a given PRC threshold, we created new PRC rule sets with conditional probabilities smaller than or equal to the threshold by deleting appropriate rules from the possible rule set.

There is no classification system recommended for the PRC. In our experiments over PRC rule sets we used the LERS classification system without partial matching. Our rationale is that the PRC is based on probabilities, while partial matching is a heuristic idea that does not match probability theory.

6 Experiments

This paper presents the experiments conducted on ten typical data sets. All these data sets are available on the UCI ML Repository. Four of these data sets, *Breast cancer* (*Slovenia*), *Breast cancer* (*Wisconsin*), *Primary tumor* and *Postoperative patient*, were in their original form. Remaining six data sets were discretized using an agglomerative cluster analysis method of discretization [20]. During this process, the percentage of conflicting cases was set to lower levels than 100%.

Table 4. Standard deviations, first part

Data set	PRC threshold					
	0.2	0.3	0.4	0.5	0.6	
Australian credit	0.67	0.69	0.63	0.81	0.80	0.72
Breast cancer (Slovenia)	1.44	2.13	2.05	2.12	2.04	1.98
Breast cancer (Wisconsin)	1.03	1.01	1.18	0.97	1.27	0.87
Echocardiogram	1.33	2.36	1.90	1.77	1.82	1.74
German credit	0.86	0.63	0.84	0.90	1.01	1.05
Image segmentation	1.67	1.58	1.56	1.73	2.25	1.38
Iris	1.56	1.50	1.34	1.64	0.76	1.43
Primary tumor	1.68	1.46	1.56	1.25	1.62	2.05
Postoperative patient	3.24	2.83	3.73	3.87	3.10	3.22
Wine recognition	1.15	0.91	1.12	1.46	1.23	1.19

Results of our experiments, based on ten-fold cross validation repeated 30 times, are presented in Tables 2–5. We evaluated the quality of these results using two tests, first the standard statistical test about the difference between two means [21], and then the sign test [21] to the results of the first test. Using the test about the difference between two means we conclude that for all values of the PRC threshold level, all rule sets induced by PRC, except the *Iris* data set and thresholds 0.2, 0.3, 0.4 and 0.5, the performance of the LERS (possible rules) is significantly better than performance of the PRC (two-tailed test, 5% significance level). Using the sign test to the results of the test about the difference between

two means, we observe that the performance of the LERS (possible rules) is better than the best possible rule set induced by the PRC with any threshold (one-tailed test, 1% significance level).

Table 5. Standard deviations, second part

Data set	PRC threshold			LERS	
	0.8	0.9	1.0	certain rules	possible rules
Australian credit	1.07	1.02	2.03	1.08	0.81
Breast cancer (Slovenia)	1.67	2.02	1.69	1.67	1.47
Breast cancer (Wisconsin)	0.84	0.59	2.02	0.66	0.57
Echocardiogram	3.10	8.72	8.82	3.75	1.86
German credit	1.05	1.61	1.02	0.89	0.70
Image segmentation	1.48	1.34	0.77	1.32	1.26
Iris	2.27	2.45	0.54	0.45	0.98
Primary tumor	1.76	1.61	1.58	1.42	1.85
Postoperative patient	3.90	3.44	3.31	2.48	2.68
Wine recognition	1.14	1.73	2.80	1.26	1.09

The LERS possible rule approach to rule induction is either better than the LERS certain rule approach or the difference is statistically insignificant (two-tailed test about the difference between two means, 5% significance level). Again, the sign test shows that the LERS possible rule approach is significantly better than the LERS certain rule approach (one-tailed test, 1% significance level).

7 Conclusions

As follows form our experiments, the LERS approach to rule induction (using the MLEM2 algorithm and possible rule sets) is significantly better than a simple probabilistic rough classification. Only for one data set, the *Iris* data set, PRC was in some cases (for thresholds between 0.2 and 0.5) better than MLEM2. Additionally, for every data set there exists some optimal PRC threshold such that the error rate is the smallest. For example, for the *Australian credit* data set such threshold is 0.5. Finally, the LERS possible rule approach is significantly better than the LERS certain rule approach.

References

1. Pawlak, Z.: Rough classification. International Journal of Human-Computer Studies 51, 369–383 (1999)
2. Grzymala-Busse, J.W.: Knowledge acquisition under uncertainty—A rough set approach. Journal of Intelligent & Robotic Systems 1, 3–16 (1988)

3. Grzymala-Busse, J.W.: Managing uncertainty in machine learning from examples. In: Proceedings of the Third Intelligent Information Systems Workshop, pp. 70–84 (1994)
4. Grzymala-Busse, J.W.: A new version of the rule induction system LERS. Fundamenta Informaticae 31, 27–39 (1997)
5. Stefanowski, J.: Algorithms of Decision Rule Induction in Data Mining. Poznan University of Technology Press, Poznan (2001)
6. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
7. Yao, Y.Y.: Probabilistic rough set approximations. International Journal of Approximation Reasonong 49, 255–271 (2008)
8. Ziarko, W.: Probabilistic approach to rough sets. International Journal of Approximate Reasoning 49, 272–284 (2008)
9. Grzymala-Busse, J.W., Ziarko, W.: Data mining based on rough sets. In: Wang, J. (ed.) Data Mining: Opportunities and Challenges, pp. 142–173. Idea Group Publ., Hershey (2003)
10. Yao, Y.Y.: Interpreting concept learning in cognitive informatics and granular computing. IEEE Transactions on System, Man and Cybernetics B 39, 855–866 (2009)
11. Mitchell, T.M.: Generalization as search. Artificial Intelligence 18, 203–226 (1982)
12. Yao, Y.Y., Wong, S.K.M.: A decision theoretic framework for approximate concepts. International Journal of Man-Machine Studies 37, 103–119 (1996)
13. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences 46(1), 39–59 (1993)
14. 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, pp. 243–250 (2002)
15. Grzymala-Busse, J.W., Yao, Y.: A comparison of the LERS classification system and rule management in PRSM. In: Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) RSCTC 2008. LNCS (LNAI), vol. 5306, pp. 202–210. Springer, Heidelberg (2008)
16. 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)
17. Chan, C.C., Grzymala-Busse, J.W.: On the attribute redundancy and the learning programs ID3, PRISM, and LEM2. Technical report, Department of Computer Science, University of Kansas (1991)
18. 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, Boston (1990)
19. Holland, J.H., Holyoak, K.J., Nisbett, R.E.: Induction. Processes of Inference, Learning, and Discovery. MIT Press, Boston (1986)
20. Chmielewski, M.R., Grzymala-Busse, J.W.: Global discretization of continuous attributes as preprocessing for machine learning. International Journal of Approximate Reasoning 15(4), 319–331 (1996)
21. Chao, L.L.: Introduction to Statistics. Brooks Cole Publishing Co., Monterey (1980)

DRSA Decision Algorithm Analysis in Stylometric Processing of Literary Texts

Urszula Stańczyk

Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland

Abstract. When the indiscernibility relation, fundamental to Classical Rough Set Approach, is substituted with dominance relation, it results in Dominance-Based Rough Set Approach to data analysis. It enables support not only for nominal classification tasks, but also when ordinal properties on attribute values can be observed [1], making DRSA methodology well suited for stylometric processing of texts. Stylometry involves handling quantitative features of texts leading to characterisation of authors to the point of recognition of their individual writing styles. As always, selection of attributes is crucial to classification accuracy, as is the construction of a decision algorithm. When minimal cover gives unsatisfactory results, and all rules on examples algorithm returns very high number of rules, usually constraints are imposed by selection of some reduct and limiting the decision algorithm by including within it only rules with certain support. However, reducts are typically numerous and within them some of conditional attributes are used more often than others, which is also true for conditions specified by decision rules. The paper presents observations how the frequency of usage for features reflects on the performance of decision algorithms resulting from selection of rules with conditional attributes exploited most and least often.

Keywords: DRSA, Decision Algorithm, Relative Reduct, Feature Selection, Stylometry, Data Mining.

1 Introduction

Modern stylometry can be seen as a successor of the historical textual analysis that was used to prove or disprove the authenticity of documents. Yet while the latter had to rely on most striking features of texts such as specific language, which is likely to be imitated, the former, with support of computational power of computers, can exploit even most common elements and parts of speech. Since used rather subconsciously, they are more reliable textual markers, conveying individual writing styles. With aims at author characterisation, comparison and attribution stylometry belongs with information retrieval domain [2],[3].

Constantly growing corpus of texts and changing linguistics cause stylometry to require an informed selection of characteristic features, the problem that remains unsolved within stylometry itself, rather being shifted to the processing

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 600–609, 2010.

phase. Techniques applied typically come either from statistics (e.g. PCA, LDA) or artificial intelligence area (e.g. ANN, SVM). Rough set methodology, which belongs with the latter group, possesses inherent mechanisms of establishing significance of features describing the input data. This is obtained by determining relative reducts - such subsets of conditional attributes that keep intact the classification properties of the decision table [4]. With help of relative reducts there are constructed decision algorithms consisting of rules that specify conditions which must be met for each decision to be applicable.

It is quite common that the number of reducts is high. Also, short decision algorithms providing just minimal cover not necessarily give the best classification accuracy, whereas generating all rules on examples can cause the length of the decision algorithm to become of unmanageable proportions. Imposing some constraints on it, while still keeping the highest recognition ratio possible, can be obtained by careful choice of a reduct and discarding rules with support below some set minimum. However, when there are no domain-based indicators as to which features are more significant than others (stylometry does not point out any particular descriptors), this selection of the reduct becomes problematic.

In analysis the relative reducts and conditions constructing rules for the decision algorithm reveal that some conditional attributes are exploited more often than others. This observation leads to ordering of attributes accordingly to the detected frequencies, one for reduct- and one for rule-based analysis. The paper presents research on how this established ordering can next be used to reduce the set of conditional attributes and found decision rules by removing those features that are used most and least frequently and how it reflects on the performance of DRSA-based classifier within the stylometric task of authorship attribution.

2 Stylometric Analysis

Stylometric analysis yields enough information on authors and their writing styles that it is possible to characterise, compare and recognise them. Historically the task of authorship attribution has always been considered of the primary importance as it enables to answer questions on authencity of some documents or settle doubts about authorship. In the past the analysis relied on human observations of noticeable features of language, yet contemporary techniques exploit even common parts of speech which, used rather subconsciously, are less likely to be imitated and thus allow to recognise individual writing styles. The origins of modern stylometry are usually dated to XIXth century and works of Mendenhall who as the first proposed to use quantitative features of texts [5].

Textual descriptors enabling to settle the question of authorship form so-called author invariant, however, there is no consensus as to which features of a text constitute it [6]. Typically as markers for the analysis there are used statistics such as usage frequency for words or letters (lexical features), structures of sentences formed by punctuation marks (syntactic), organisation of text into constructing elements such as headings or paragraphs (structural), or words of certain meaning in the given context (content-specific). The choice of descriptors

is one of crucial issues within the analysis while the other is a technique applied to the task, with approaches either from statistical-oriented computations or the artificial intelligence domain. Within the former group there are employed for example Principal Component Analysis, or Markovian Models, while from the latter area there are used Genetic Algorithms, Artificial Neural Networks [7], or Rough Set Approach [8] that was employed in the presented research.

The frequencies of textual markers studied have continuous values, and the classical rough set approach (CRSA) deals only with discrete data, thus either there had to be applied some discretisation strategy by defining a discretisation factor, or modified indiscernibility relation applicable for continuous attributes [9], or instead of using the classical methodology there could be employed dominance-based rough set approach (DRSA), which integrates dominance relation with rough approximation, used in multicriteria decision support [10,11].

3 DRSA-Based Data Mining

The first step in the rough set-based approach, proposed by Z. Pawlak [12], is defining a decision table that contains the whole knowledge about the Universe. Columns of the table are defined by conditional C and decision D attributes while rows X specify values of these attributes for objects of the Universe.

While the indiscernibility principle of the classical rough set approach says that if two objects x and y are indiscernible with respect to considered attributes then they should be classified in the same way, that is to the same class, the *dominance* or *Pareto principle* of the dominance-based rough set approach states that if x is at least as good as y with respect to the attributes, then x should be classified at least as good as y. That is why CRSA cannot deal with preference order in the value sets of attributes and it supports classification only when it is nominal, whereas DRSA has been proposed to deal with cases when the value sets of attributes are ordered [10,13]. In classification problems condition attributes are called *criteria* and with many of them the problem becomes that of multicriteria classification or decision making [14].

It often happens that the set of decision attributes contains just a single attribute $D = \{d\}$, partitioning the Universe into a finite number of classes $\boldsymbol{Cl} = \{Cl_t\}$, with $t = 1, \ldots, n$. The classes are ordered and the increasing preference is indicated by increasing indices. Due to this preference order present in the set of classes $\boldsymbol{Cl}$, the sets to be approximated are upward or downward unions of classes, or dominance cones, respectively defined as

$$Cl_t^{\geq} = \bigcup_{s \geq t} Cl_s \qquad Cl_t^{\leq} = \bigcup_{s \leq t} Cl_s \tag{1}$$

It is also quite common that neither all attributes nor all their values are necessary for correct classification of all objects and within the rough set approach there are included dedicated tools that enable to find, if they exist, such functional dependencies between attributes that allow for decreasing their number without any loss of classification properties of DT: relative reducts [15]. Each

irreducible subset $P \subseteq C$ such that preserves the quality of approximation with the selected criteria ($\gamma_P(\boldsymbol{Cl}) = \gamma_C(\boldsymbol{Cl})$) is called a *reduct*. A decision table can have many reducts [16] and their intersection is called a *core*.

Approximations of dominance cones is the starting point for induction of decision rules and a set of rules is *complete* when every object of the decision table can be classified into one or more groups according to the rules, that is no object remains unclassified. A set of rules is *minimal* when it is complete and irredundant, that is exclusion of any rule makes the set incomplete.

The minimal set of rules does not guarantee the highest classification accuracy. It includes only necessary rules created for training samples and they hardly can cover all points of the multidimensional input space. Testing samples can so vary from learning ones that there are no rules matching. That is why there are also tried approaches generating all rules and then by some methodology an optimised classifier can be built, comprising selection of rules, basing on support, assumed weights, or even a fitness score can be calculated as in genetic algorithms basing on observed frequency of application for testing examples.

4 Input Data

In experiments as the input data there were taken literary texts of two writers, H. James and T. Hardy. Three sets of samples were constructed: learning ones being the foundation for the decision table (Table 1) based on 30 parts taken from 3 novels for a writer (180 samples), testing 1 based on 8 parts from 5 novels (80 samples), and testing 2 based on 10 parts from 3 novels (60 samples). The first testing set was used to find the constraints on required minimal support for rules to be included within a decision algorithm, while the second testing set was used for additional confirmation that algorithms with constraints imposed upon rules perform with the same merit when applied to completely new data.

The samples were created by computing frequencies of usage for lexical and syntactic textual markers - 25 conditional attributes: but, and, not, in, with, on, at, of, this, as, that, what, from, by, for, to, if, a fullstop, a comma, a question mark, an exclamation mark, a semicolon, a colon, a bracket, a hyphen. With two authors, the decision attribute had two distinguished values ("hardy" as lower and "james" higher). For all conditional attributes there was assumed an arbitrary ordering of "cost" type (the lower, the classification to the higher class).

Results for classification with minimal cover and all rules on examples algorithms are given in Table 2. Usually they come into three categories: correct recognition, incorrect recognition, and with ambiguous decision when there were several partial verdicts leading to conflicting classification or when there were no rules matching. In this table as well as in all others included in the paper results are presented for correct decisions only, disregarding these possibly correct if some voting of partial decisions was employed. This attitude shortens and simplifies the classification procedure as no additional processing is needed.

It comes as no surprise that the number of rules in the minimal cover algorithm (Table 3) is significantly outranked by the full algorithm. What is

Table 1. Decision table

	but	and	not	in	···	:	(	-	author
1	0.0046	0.0355	0.0034	0.0209		0.0005	0	0.0157	hardy
2	0.0041	0.0304	0.0078	0.0165		0	0	0.0096	hardy
3	0.0053	0.0257	0.0037	0.0148		0.0002	0.0001	0.0284	hardy
4	0.0068	0.0292	0.0057	0.0108		0.0005	0	0.0171	hardy
⋮									
177	0.0103	0.0173	0.0056	0.0137		0.0012	0	0.0427	james
178	0.01	0.0156	0.0031	0.0127		0.001	0	0.0538	james
179	0.008	0.0122	0.0046	0.0117		0.0012	0	0.0303	james
180	0.0073	0.0077	0.0028	0.0137		0.0017	0	0.0274	james

Table 2. Classification results for decision algorithms involving all attributes

	Supp.	Nr of rules in short DA	Class. Test 1	Class. Test 2
Minimal cover (61 rules)	4	19	45,00%	48,33%
All rules on examples (46191 rules)	40	90	70,00%	76,66%

troublesome, classification accuracy differs so much that in the former case is totally unacceptable and useless, while in the latter not great but satisfactory.

In the past research [7] it was shown that relative reducts applied within classical rough set approach can be successfully used in reduction of characteristic features for ANN-classifier while preserving its performance. The presented approach exploited ordering of attributes based on frequency of usage in reduct construction. Similar attitude can be tried in building modified decision algorithms by including only rules with attributes that occur most and least often.

There were 6664 relative reducts, yet the relative core turned out to be empty. On the other hand the union of all reducts was equal to the whole initial set of attributes, which was also true for conditions in the calculated rules, indicating that no feature from these studied could be disregarded without further analysis.

5 Obtained Results of Feature Reduction

With total number of constructed decision rules several times higher than that of relative reducts, frequency indicators of conditional attributes for both obviously cannot possibly be numerically the same, what is of particular interest though, the resulting ordering, specified in Table 4, is not the same, with the exception of just two attributes: the most ("of") and the least ("but") frequently employed ones, which in both cases respectively open and close the list.

Since neither stylometry nor rough set approach could precisely answer the question which textual markers could be disregarded without undermining the power of the classifier, several subsets were tried exploiting the ordering of features presented. Removing some attribute meant discarding all rules from the full algorithm that involved conditions on this attribute.

Table 3. Minimal cover algorithm with rules of support at least 4

```
Rule 1. (by>=0.006700) => hardy
Rule 2. (exclamation>=0.013600) => hardy
Rule 5. (not>=0.011400) & (from>=0.002300) => hardy
Rule 6. (of>=0.040900) => hardy
Rule 8. (bracket>=0.000500) & (exclamation>=0.006900) => hardy
Rule 9. (in>=0.021700) & (exclamation>=0.002000) => hardy
Rule 13. (not>=0.008900) & (in>=0.017200) => hardy
Rule 15. (from>=0.005000) & (fullstop>=0.059200) => hardy
Rule 19. (and>=0.033500) => hardy
Rule 21. (by>=0.004700) & (fullstop>=0.050300) & (of>=0.030300) => hardy
Rule 26. (from>=0.005900) => hardy
Rule 30. (and<=0.017200) => james
Rule 34. (by<=0.001300) => james
Rule 38. (and<=0.022600) & (fullstop<=0.046700) => james
Rule 39. (not<=0.003800) & (from<=0.002200) => james
Rule 43. (semicolon<=0.001900) => james
Rule 54. (and<=0.023900) & (for<=0.005600) => james
Rule 55. (not<=0.002900) & (of<=0.025200) => james
Rule 58. (of<=0.019700) & (not<=0.003900) => james
```

Table 4. Attribute occurrence indicators a) reduct-based, b) rule-based

a)

	of	3478
M1	.	3190
M2	on	3083
	,	2943
M3	not	2778
	;	2740
M4	in	2726
	by	2648
M5	this	2585
M6	at	2585
	to	2497
M7	:	2384
	!	2368
M8	and	2324

this	2585	L9	
at	2585		
to	2497	L8	
:	2384		
!	2368	L7	
and	2324		
from	2273	L6	
with	2161		
as	2108	L5	
-	2035		
?	1712	L4	
for	1609		
if	1584	L3	
what	1415		
(	1395	L2	
that	1343		
but	893	L1	

b)

M1	of	13310
M2	on	12921
	to	11838
M3	this	11426
	,	11176
M4	.	11004
	!	10639
M5	:	10326
	not	10305
M6	in	10240
	;	9797
M7	at	9082
	with	8646
M8	as	8471
M9	by	8450
	-	7996
M10	(	7950

-	7996	L6
(	7950	
if	7691	L5
from	7614	
?	7468	L4
for	7449	
what	6172	L3
that	6166	
and	4172	L2
but	3927	L1

Those algorithms that kept only most frequently exploited attributes are labelled with "L" indicating that those least frequent were reduced, and those that kept the least often used with disregarding the most often are labelled with

"M". The classification results of obtained decision algorithms are given in Table 5, with partial verdicts that would require voting not included in the numbers listed. Further reduction was tried yet resulted in correct classification around 50% or even lower, hence details are not presented.

Analysis of these results brings conclusion that for reduct-based reduction the acceptable recognition ratio can be kept till the number of attributes is not less than 16 the most frequently used, or even as few as 13 for the least often exploited. For rule-based attribute frequency analysis, maintaining the power of the classifier requires either 19 most or just 10 the least frequent features.

For all decision algorithms the classification for the first testing set is presented by listing the best results and indicating the minimal support that was imposed upon the rules included to arrive at it. For additional confirmation these short decision algorithms were also tested on the second testing set and as specified it generally follows the trend, yet with some cases of significant difference.

The classification accuracies for two testing sets plotted against the number of attributes kept and in relation to the number of rules in the shortened decision algorithms are shown in Fig. 1 and Fig. 2 respectively, in each case giving just the best result from all obtained for different versions of decision algorithms.

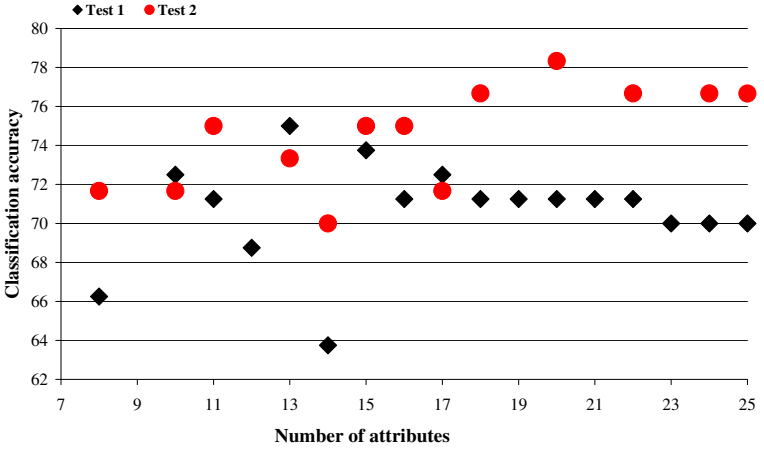


Fig. 1. Classification in relation to the number of attributes employed

Both the table with classification results specified and the graphs indicate that selection of characteristic features can be successfully performed not only by exploiting the concept of relative reducts themselves, but also basing on frequency analysis for individual attributes employed in constructed reducts and conditions for decision rules. The power of the classifier can be preserved when reducing most frequently used features as well as those used seldom.

Table 5. Classification results for reduced decision algorithms

	Nr of rules in DA	Nr of attr.	Supp.	Nr of rules in short DA	Class. Test 1	Class. Test 2
Reduct-based analysis						
L1	42263	24	40	90	70,00%	76,67%
L2	30219	22	40	68	71,25%	75,00%
L3	21834	20	40	63	71,25%	75,00%
L4	15156	18	40	60	71,25%	75,00%
L5	9673	16	40	59	71,25%	75,00%
L6	6000	14	26	122	63,75%	70,00%
L7	3463	12	18	34	68,75%	56,67%
L8	1702	10	12	83	66,25%	71,67%
L9	978	8	12	75	66,25%	71,67%
M1	25386	23	40	83	70,00%	76,67%
M2	17500	22	40	81	70,00%	76,67%
M3	9356	20	40	69	63,75%	78,33%
M4	5282	18	34	80	65,00%	76,67%
M5	2560	16	20	100	71,25%	73,33%
M6	1955	15	16	135	73,75%	71,67%
M7	930	13	14	120	75,00%	73,33%
M8	329	11	4	89	61,25%	55,00%
Rule-based analysis						
L1	42263	24	40	90	70,00%	76,67%
L2	38572	23	32	28	68,75%	81,67%
L3	28896	21	32	27	70,00%	83,33%
L4	19998	19	32	26	70,00%	81,67%
L5	12865	17	20	50	65,00%	55,00%
L6	8747	15	18	58	67,50%	58,33%
M1	32880	24	40	83	70,00%	76,67%
M2	23200	23	40	81	70,00%	76,67%
M3	12523	21	40	66	71,25%	75,00%
M4	7169	19	40	65	71,25%	75,00%
M5	3891	17	34	88	72,50%	71,67%
M6	1737	15	26	93	68,75%	75,00%
M7	1122	13	26	83	71,25%	75,00%
M8	673	11	26	74	71,25%	75,00%
M9	390	10	12	86	72,50%	66,67%
M10	246	8	12	70	66,25%	66,67%

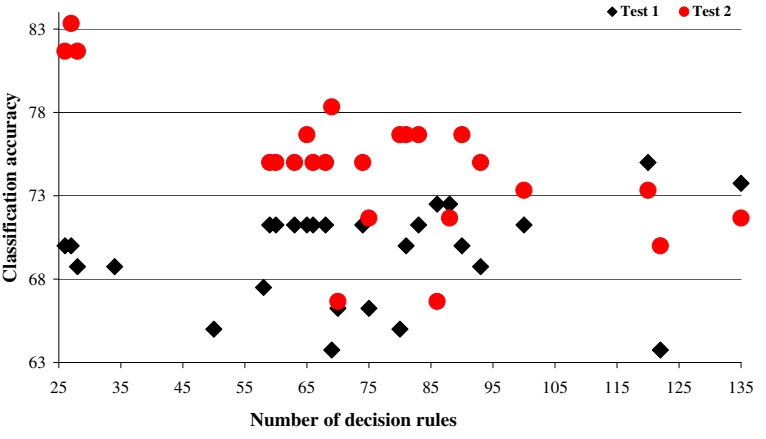


Fig. 2. Classification in relation to the number of decision rules

6 Conclusions

The paper presents the analysis of characteristic features for DRSA-classifier applied within the stylometric task of authorship attribution. Computed relative reducts and decision rules indicate ordering of attributes based on their usage frequency in reducts and rules, and this observed ordering is then employed to reduce the set of base features by leaving either the most or the least frequently used attributes. In the resulting shortened decision algorithms there are included only such rules that have no conditions on removed features. Performed tests indicate that such frequency analysis can be successfully applied for feature selection and can be considered as an alternative to selection of some relative reduct in the absence of domain knowledge about the significance of individual conditional attributes.

Acknowledgements. The software used to obtain frequencies for textual descriptors was implemented by Mr. P. Cichoń in fulfilment of requirements for M.Sc. thesis, submitted at the Faculty of Computer Science, the Silesian University of Technology, Gliwice, Poland in 2003.

4eMka Software used in search for reducts and decision rules is a system for multicriteria decision support integrating dominance relation with rough approximation [14,13]. The software is available at a website of Laboratory of Intelligent Decision Support Systems, Institute of Computing Science, Poznan University of Technology (http://www-idss.cs.put.poznan.pl/), Poland.

References

1. Greco, S., Matarazzo, B., Slowinski, R.: Rough set theory for multicriteria decision analysis. European Journal of Operational Research 129(1), 1–47 (2001)

2. Argamon, S., Karlgren, J., Shanahan, J.: Stylistic analysis of text for information access. In: Proceedings of the 28th International ACM Conference on Research and Development in Information Retrieval, Brazil (2005)
3. Peng, R., Hengartner, H.: Quantitative analysis of literary styles. The American Statistician 56(3), 15–38 (2002)
4. Shen, Q.: Rough feature selection for intelligent classifiers. In: Peters, J.F., Skowron, A., Marek, V.W., Orłowska, E., Słowiński, R., Ziarko, W.P. (eds.) Transactions on Rough Sets VII. LNCS, vol. 4400, pp. 244–255. Springer, Heidelberg (2007)
5. Peng, R.: Statistical aspects of literary style. Bachelor's Thesis, Yale University (1999)
6. Stańczyk, U.: Dominance-based rough set approach employed in search of authorial invariants. In: Kurzyński, M., Woźniak, M. (eds.) Computer Recognition Systems 3. AISC, vol. 57, pp. 315–323. Springer, Heidelberg (2009)
7. Stańczyk, U.: Relative reduct-based selection of features for ANN classifier. In: Cyran, K., et al. (eds.) Man-Machine Interactions. AISC, vol. 59, pp. 335–344. Springer, Heidelberg (2009)
8. Stańczyk, U., Cyran, K.: On employing elements of rough set theory to stylometric analysis of literary texts. International Journal on Applied Mathematics and Informatics 1(2), 159–166 (2007)
9. Cyran, K., Stanczyk, U.: Indiscernibility relation for continuous attributes: application in image recognition. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 726–735. Springer, Heidelberg (2007)
10. Greco, S., Matarazzo, B., Slowinski, R.: Dominance-based rough set approach as a proper way of handling graduality in rough set theory. In: Peters, J.F., Skowron, A., Marek, V.W., Orłowska, E., Słowiński, R., Ziarko, W.P. (eds.) Transactions on Rough Sets VII. LNCS, vol. 4400, pp. 36–52. Springer, Heidelberg (2007)
11. Słowiński, R., Greco, S., Matarazzo, B.: Dominance-based rough set approach to reasoning about ordinal data. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 5–11. Springer, Heidelberg (2007)
12. Pawlak, Z.: Rough sets and intelligent data analysis. Information Sciences 147, 1–12 (2002)
13. Greco, S., Matarazzo, B., Slowinski, R.: The use of rough sets and fuzzy sets in Multi Criteria Decision Making. In: Gal, T., Hanne, T., Stewart, T. (eds.) Advances in Multiple Criteria Decision Making, pp. 14.1–14.59. Kluwer Academic Publishers, Dordrecht (1999)
14. Greco, S., Matarazzo, B., Slowinski, R.: Handling missing values in rough set analysis of multi-attribute and multi-criteria decision problems. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) RSFDGrC 1999. LNCS (LNAI), vol. 1711, pp. 146–157. Springer, Heidelberg (1999)
15. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
16. Moshkow, M., Skowron, A., Suraj, Z.: On covering attribute sets by reducts. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 175–180. Springer, Heidelberg (2007)

Blind Music Timbre Source Isolation by Multi- resolution Comparison of Spectrum Signatures

Xin Zhang[1], Wenxin Jiang[2], Zbigniew W. Raś[2,4,5], and Rory Lewis[3]

[1] Univ. of North Carolina, Dept. of Math. and Comp. Science, Pembroke, NC 28372, USA
[2] Univ. of North Carolina, Dept. of Comp. Science, Charlotte, NC 28223, USA
[3] Univ. of Colorado, Dept. of Comp. Science, Colorado Springs, CO 80933, USA
[4] Polish-Japanese Institute of Information Technology, 02-008 Warsaw, Poland
[5] Polish Academy of Sciences, Institute of Comp. Science, 01-237 Warsaw, Poland
xin.zhang@uncp.edu, {wjiang3,ras}@uncc.edu, rlewis@eas.uccs.edu

Abstract. Automatic indexing of music instruments for multi-timbre sounds is challenging, especially when partials from different sources are overlapping with each other. Temporal features, which have been successfully applied in monophonic sound timbre identification, failed to isolate music instrument in multi-timbre objects, since the detection of the start and end position of each music segment unit is very difficult. Spectral features of MPEG7 and other popular features provide economic computation but contain limited information about timbre. Being compared to the spectral features, spectrum signature features have less information loss; therefore may identify sound sources in multi-timbre music objects with higher accuracy. However, the high dimensionality of spectrum signature feature set requires intensive computing and causes estimation efficiency problem. To overcome these problems, the authors developed a new multi-resolution system with an iterative spectrum band matching device to provide fast and accurate recognition.

Keywords: Blind Music Sound Sources Isolation, STFT (Short-Time Fourier Transform), Automatic Indexing, KNN, Spectral Features.

1 Introduction

The rapid advances in computer storage and network techniques brought the emergency of huge multimedia repositories, where fast access to individual segment piece becomes more and more important in demands while manual indexing is a non-trivial work. Automatic indexing of music instruments in the same channel is one of the important subtasks.

A piece of digital music recording in a raw format contains some header information about the file and a huge sequence of sampling data of integers to represent the air fluctuations of sounds over time, where a typical sampling data rate is 44,100 per second for compact discs.

Features, such as MPEG-7 descriptors and other popular features, which are successfully applied in identifying music timbre in monophonic sounds, fail to isolate music source in multi-timbre or polyphonic objects, where multiple music instruments

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 610–619, 2010.

are playing at the same time. More so, temporal features are difficult to be applied in multi-timbre or polyphonic objects, since the detection of the start and end position of each music segment unit is very difficult while the partials are overlapping with each other (so-called a Cocktail Party Problem [6]).

Numerous methods for blind signal separation have been explored for a wide range of business domain spanning from finance to general biomedical signal processing. Filtering Techniques ([2], [3], [20]), ICA ([4], [7], [9]) and DUET [12] require different sound sources to be stored separately in multiple channels; therefore they are not suitable in isolating blind music sources in the same channel of the recordings. Most often, Factorial Hidden Markov Models (HMM [16]) work well for sound sources separation, where fundamental frequency range is small and the variation is subtle. However, unfortunately, western orchestral musical instruments can produce a wide range of fundamental frequencies with dynamic variations. Spectral decomposition is used to efficiently decompose the spectrum into several independent subspaces [5] with smaller number of states for HMM. Commonly, Harmonic Sources Separation Algorithms have been used to estimate sound sources by detecting their harmonic peaks, decoding spectrum into several streams and re-synthesizing them separately. This type of methods relies on multi-pitch detection techniques and iterative Sinusoidal Modeling (SM) [8]; therefore they are designed to deal with only harmonic sounds. For the purpose of interpolating the breaks in the sinusoidal component trajectories, numerous mathematical models have been explored: linear models [21], non-linear models such as high degree interpolation polynomials with cubic spine approximation model [8], etc. However, it is very difficult to develop an accurate sinusoidal component model to describe the characteristics of musical sound patterns for all the western orchestral instruments. Kitahara et al. developed weights for features to minimize the influence of sound overlaps [13], which also assumes perfect fundamental frequency detection. Spectral features have been explored in peer research with traditional classifiers and proved a possible way to identify sound sources in multi-timbre music objects [11]. However, such features intuitively do not include sufficient information about sound wave behaviors along time. Also, when spectrum signatures are fed into classical classifiers, the order of frequency bins won't be taken into consideration. Therefore, the estimation accuracies of the traditional classifiers with only spectral features are normally not desirable. To overcome the problem, the authors developed a spectrum band matching device based on multi-resolution iterations to provide fast and accurate estimation based on an enlarged estimation range from the classifiers with relaxed confidence level for music instrument families.

2 Blind Music Timbre Source Isolation System

The authors developed a robust blind music sound source separation system with connection to a database of features extracted from a wide range of western music orchestral instruments, which consists of five major modules: a STFT converter with hamming window, a feature extraction engine, a K-Nearest-Neighbor classifier, an iterative sound band matching device, and an FFT subtraction mechanism for the estimated predominant sound source.

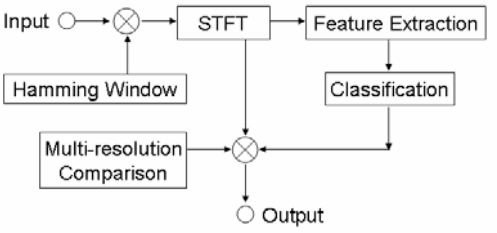


Fig. 1. System overview

The STFT converter divides a digital audio object into a sequence of frames, applies STFT transform to the mixed sample data of digital music data from time domain to frequency domain with a hamming window sliding evenly over time, and outputs NFFT (next-larger power-of-two number of samples of digital sound data from sampling window) discrete points.

The feature extraction engine calculates spectral features based on the spectrum information of the adjacent frames and stores them into a large music database used for training classifiers. In the next section, more details about those features will be presented.

The K-Nearest-Neighbor classifier takes in the flat spectrum features, constructs models, and estimates timbre categorization in terms of a series of machine understandable schemes.

An iterative sound band matching device is applied to further trim the bottom level of the tree models so that only the closely matched exemplary spectrum will remain, where each iteration rules out a certain amount of unlikely objects.

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 (Inverse discrete Fourier Transform) for each frame, and outputs resultant remaining signals into a new audio data file.

3 Feature Extraction

The authors developed a large database with spectral features and temporal attributes including popular features in this research area, such as MPEG7 spectral descriptors and Mel frequency cepstral coefficients, as well as some new temporal features.

Spectrum Centroid and Spread [1] The Audio Power Spectrum Centroid describes the center-of-gravity of a log-frequency power spectrum. Spectrum spread is defined as the Root Mean Square value of the deviation of the Log frequency power spectrum with respect to the gravity center in a frame. These two parameters economically indicate the pre-dominant frequency range.

Spectrum Flatness [1] describes the flatness property of the power spectrum within a frequency bin, which is ranged by edges. It is an array of aggregations in a set of frequency bands, where frequency band is defined by two adjacent cutting edges with a quarter octave resolution spanning eight octaves.

Spectrum Basis Functions [1] are used to reduce the dimensionality of a group of adjacent frames of the normalized spectrum envelope in a log-arithmetic scale with a quarter-octave resolution by projecting from the space of 32 dimensions of frequency bands into a space of 10 dimensions with compact salient statistical information based on singular value decomposition.

Spectrum Projection Function [1] is computed by an inner product of the resultant low dimensional spectrum vector from the spectrum basis functions and the normalized spectrum envelope in a log-arithmetic scale. It is used to represent low-dimensional features of a spectrum after projection against a reduced rank basis of 10.

Predominant Harmonic Peaks [22] is an array of power spectrum coefficients of the local harmonic peaks in a normalized log-arithmetic scale based on the predominant fundamental frequency, where the first 28 of items are considered significant and therefore chosen as features in this research.

Harmonic Spectral Centroid [1] is computed as the average over the sound segment duration in the quasi-steady state of the instantaneous harmonic spectral centroid within a frame. The instantaneous harmonic spectral centroid is computed as the amplitude in a linear scale weighted mean of the harmonic peak of the spectrum.

Harmonic Spectral Spread [1] is computed as the average over the sound segment duration in the quasi-steady state of the instantaneous harmonic spectral spread within a frame. The instantaneous harmonic spectral spread is computed as the amplitude weighted standard deviation of the harmonic peaks of the spectrum with respect to the instantaneous harmonic spectral centroid.

Harmonic Spectral Variation [1] is defined as the mean value over the sound segment duration of the instantaneous harmonic spectral variation, which is calculated as the normalized correlation between the amplitude of the harmonic peaks of the current frame and the immediate previous frame.

Harmonic Spectral Deviation [1] is computed as the average over the sound segment duration of the instantaneous Harmonic Spectral Deviation in each frame, which is computed as the spectral deviation of the log amplitude components from a global spectral envelope.

Temporal Centroid [1] is calculated as the time average over the signal envelope.

Zero crossing [17], [19] counts the number of times that the signal sample data changes signs in a frame.

Roll-off is a measure of spectral shape, which is used to distinguish between voiced and unvoiced speech [14]. The roll-off is defined as the frequency below which a proportion (empirical value: 85%) of the accumulated magnitudes of the spectrum is concentrated.

Flux is used to describe spectral rate of change [17]. It is computed by the total difference between the magnitude of FFT points in a frame and its successive frame.

Mel frequency cepstral coefficients describe the spectrum according to the human perception system in the mel scale [15]. They are computed by grouping the STFT points of each frame into a set of 40 coefficients by a set of 40 weighting curves with logarithmic transform and a discrete cosine transform (DCT). The authors used the MFCC functions from the Julius software toolkit [1].

4 Classification

Numerous types of classifiers have been explored in timbre estimation by peer researchers, while so far there is no classifier, which is supreme in identifying all types of timbres in polyphonic or multi-timbre sounds among peer types of classifiers [23]. In this research, to explore the recognition rate of popular peer spectral features, decision tree was applied; while for spectrum signature features, K-Nearest-Neighbor algorithm was chosen for its fair performance with high dimensional feature sets (over 9,600 dimensions), where each frequency bin was treated as a feature. In case that accumulated error in a high dimensional space may bias the final estimation of timbre, we relaxed the confidence level, so that a group of possible candidates were collected as the output of the KNN classifiers. Further, a multi-resolution comparison device was applied to rule out unlikely candidates.

5 Multi-resolution Comparison

Searching for the closest matched pattern through high resolution of over eight thousands of FFT points by Euclidean distance may endanger the result by accumulated error as well as by the loss of order information along the frequency dimension. Actually, it is also opposite to the human visionary perception system. For example, when one recognizes a picture of the Eiffel Tower, does he or she checks from beam to beam assuming that beam is the atomic unit in the picture? No, on the contrary, most people would rather start from the outline shape, which is an abstract of details. In this research, authors started searching through vectors of aggregation of the frequency bins by an exponent order of resolution from low to high, where each round of comparison rules out a certain percentage of unlikely spectrum patterns as shown in the Figure 2.

In each round, the spectrum signature $V(\alpha)$ is computed by the following formula:

$$V_i^k(\alpha) = \left(\sum_{n=1}^{N_k(\alpha)} 10\log_{10} \frac{\chi_{i*N_k(\alpha)+n}}{\chi_{\max} - \chi_{\min}} \right) / N_k(\alpha) \tag{1}$$

where $\alpha \in \{3,4,5\}$ is the base, $V_i^k(\alpha)$ *is the* i^{th} feature in the k^{th} resolution level, χ is a vector of the power spectrum coefficients, and $N_k(\alpha)$ is computed by

$$N_k(\alpha) = \frac{M}{\alpha^k} \tag{2}$$

where M is the total number of FFT points. To limit the total number of iterations, Table 1 is used to show what values of k are allowed for each α; in each round/level k, 1 out of α^k points is chosen; α is used to yield even distribution of each resolution.

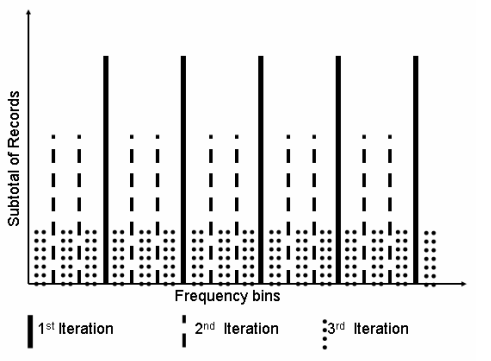


Fig. 2. Comparisons are applied based on iterative aggregation in resolution from low to high

Table 1. The relationship between α and k. Positive sign means the adoption of the combination of α and k values in our experiments.

	k=1	k=2	k=3	k=4	k=5
α=3	+	+	+	+	+
α=4	+	+	+	+	
α=5	+	+	+		

Given a dataset D with S records in total, where $S >> M$, the time complexity C_α' of the spectrum signature matching strategy based on spectrum signature feature set can be represented by comparison cost, assuming that φ is the percentage of the total FFT that remains in the k^{th} resolution level.

$$C_\alpha' = M \cdot S \cdot \varphi^{8-\alpha} \sum_{k=0}^{8-\alpha} \left(\frac{1}{\alpha \cdot \varphi} \right)^k , \quad \alpha = 3, 4, 5 \tag{3}$$

In a flat spectrum signature comparison system, the time complexity C is $O(MS)$, while the comparison of C with C_α' for each base level α is listed in the following table (φ=50% was chosen in this example for the sake of simplicity):

Table 2. Percentagewise speed comparison between a new exponential multi-resolution spectrum signature matching device and a flat matching device

	C_α'	C
α=3	8.55	100
α=4	12.11	100
α=5	20.30	100

Theoretically inferred by the above formula and data, the time complexity of spectrum signature matching strategy can be dramatically improved by the proposed multi-resolution spectrum matching system.

6 Experiments

In this research, the STFT experiments used a sampling window size of 0.12 second and a hop size of 0.04 second on music recording segments at the sampling rate of 44,100Hz, which is a typical value for compact disks. The training dataset contains 121790 spectrum signatures for the frames in the stable state of 3323 musical segment objects, which are played in the fourth octave C and originated from the MUMS (McGill University Masters Samples), assuming that similar results may be generated from music objects in other pitches. In real multimedia database, the data size of spectrum signatures will be in billions or trillions, as the musical segment objects were sampled every one second in short music sounds, of which the duration varies from around one to three seconds. Each spectrum signature contained 8192 FFT points. The training dataset included 26 music instruments: electric guitar, bassoon, oboe, b-flat clarinet, marimba, c trumpet, e-flat clarinet, tenor trombone, French horn, flute, viola, violin, English horn, vibraphone, accordion, electric bass, cello, tenor saxophone, b-flat trumpet, bass flute, double bass, alto flute, piano, Bach trumpet, tuba, and bass clarinet. The testing dataset consisted of 52 music recording pieces synthesized by Sound Forge sound editor [18], where each piece was played by two different music instruments.

The system was implemented in .NET C++ and MS SQLSERVER2005. The K Nearest Neighbor classifier package used in the experiments was from Microsoft SQLServer 2005. K= 7 was chosen empirically.

Two experiments were investigated to compare the efficiency and accuracy of the features for multi-timbre sounds: one was to check the accuracy of the popular peer features against the multi-resolution spectrum features; the other was to check the efficiency of multi-resolution spectrum signatures. In both experiments, accumulated confidence values were applied as votes for the top instrument candidates. In experiment I, we focused on the recognition rate instead of efficiency, since the peer spectral features contained much less dimensions of information than spectrum signatures; therefore the corresponding recognition results were fast and of low rate. In experiment II, linked lists were used to store the band coefficients for each tie of the resolution.

To compare the results with the traditional feature based classification strategy, five groups of spectral features (calculated for spectrum divided into 33 frequency bands) were extracted mainly from the MPEG-7 standard introduced in the previous section of Feature Extraction and fed into a set of decision tree classifiers for timbre estimation:

Group1: *Band Coefficients* = $\{b_n : 1 \le n \le 33\}$ – coefficients for Spectrum Flatness bands.

Group2: *Projections* = $\{p_n: 1 \le n \le 33\}$ – Spectrum Projection dimensions.

Group3: *MFCC* = $\{m_n: 1 \le n \le 13\}$ – Mel frequency cepstral coefficients.

Group4: *Harmonic Peaks* = $\{h_n: 1 \le n \le 28\}$ – harmonic partials of the predominant sound source.

Group5: Other Features include:

- Temporal Centroid,
- Log-arithmetic Spectral Centroid,
- Log-arithmetic Spectral Spread,
- Energy,

- Zero Crossings,
- Spectral Centroid,
- Spectral Spread,
- RollOff,
- Flux,
- Sum of the Spectrum Flatness band coefficients,
- Minimum, maximum, sum, distance, and standard deviation of the Spectrum Projection dimensions as well as of the Spectrum Basis dimensions, where distance represents a dissimilarity measure: distance of a matrix is calculated as the sum of absolute values of differences between each pair of elements on different rows and columns. Distance for a vector is calculated as the sum of dissimilarity (absolute difference of values) of every pair of coordinates in the vector.

The performance of our algorithm was measured using recognition rate R, calculated as the percentage of the correct estimations over the existing ones in the multi-timbral sound pieces.

Table 3. Music instrument recognition rate in experiment I

Experiment description	Recognition Rate (%)
Spectral features + decision tree	48.65
Flat spectrum features + KNN	82.43
α-base resolution spectrum features + KNN (α=3, 4, 5)	82.43

Table 3 shows that the multi-resolution spectrum features system with KNN classifiers had the same recognition rate as the flat one, which were both significantly better than the spectral features.

Table 4. Music instrument recognition efficiency in experiment II

Experiment description	Recognition Time (second)
Flat spectrum features + KNN	2560
α-base resolution spectrum features + KNN (α=3)	511
α-base resolution spectrum features + KNN (α=4)	524
α-base resolution spectrum features + KNN (α=5)	550

Table 4 shows that the multi-resolution spectrum features system significantly reduced the computing time to estimate the predominant music timbre in the music objects, which coincided the authors' theoretical derivation. The smaller the base, the more the iterations for the FFT points, therefore the faster the estimation. As the total number of training objects in the multimedia database grows, the difference among recognition time of different resolutions shall be further increased.

7 Conclusion

This research explored a new exponential multi-resolution spectrum signature matching device with KNN classifiers for blind music sound source isolation of multi-timbre musical objects. Temporal features were excluded in the experiments, since the detection of the start and end position of each multi-timbral music segment unit is very difficult and error prone. To compare the recognition rate, the authors developed two different training datasets: a spectral feature dataset and a spectrum signature feature dataset of multi-resolution. Traditional spectral features reduce data size for the limitation of input feature size of classic classifiers, but cause too much information loss for accurate music instrument detection. On the other hand, flat spectrum data is of high dimension and contains much more information, but does not suit most classic classifiers expect KNN. The authors designed a new algorithm with multi-resolution KNN and compared it with the peer spectral feature based algorithm. Overall, spectrum signature features were shown to provide significantly higher recognition rate for predominant music instrument than spectral features, as spectral features provided economic computation but contained not sufficient information for timbre recognition. Spectrum signature features with the multi-resolution matching device were proved same recognition rate as that with a flat matching device while the computing efficiency of the former system was much better than the latter one.

In the future, authors will explore the possibility to further improve the recognition rate of this exponential multi-resolution spectrum signature matching device with KNN classifiers by adding more carefully weighted new features, as the system can afford high dimensional dataset computing. On the other hand, feature selection algorithm may be applied to optimize the classification performance.

Acknowledgments. This work was supported by the National Science Foundation under grant IIS 0968647. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Akinobu, L., et al.: Julius software toolkit, `http://julius.sourceforge.jp/en/`
2. Balan, R.V., Rosca, J.P., Rickard, S.T.: Robustness of parametric source demixing in echoic environments. In: Proc. Int. Conf. on Independent Component Analysis and Blind Source Separation (ICA), pp. 144–148 (2001)
3. Brown, G.J., Cooke, M.P.: Computational auditory scene analysis. Computer Speech and Language 8, 297–336 (1994)
4. Cardoso, J.F.: Blind source separation: statistical principles. Proceedings of the IEEE 9(10), 2009–2025 (1998)
5. Casey, M.A., Westner, A.: Separation of mixed audio sources by independent subspace analysis. In: Proc. International Computer Music Conference (ICMC), pp. 154–161 (2000)
6. Cherry, E.C.: Some Experiments on the Recognition of Speech, with One and with Two Ears. Journal of the Acoustical Society of America 24, 975–979 (1953)
7. Davies, M.E.: Audio source separation. In: Mathematics in Signal Processing V. Oxford University Press, Oxford (2002)

8. Dziubinski, M., Dalka, P., Kostek, B.: Estimation of Musical Sound Separation Algorithm Effectiveness Employing Neural Networks. Journal of Intelligent Information Systems 24(2/3), 133–158 (2005)
9. Hyvarinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley & Sons, Chichester (2001)
10. ISO/IEC JTC1/SC29/WG11, MPEG7 Overview (2002), http://mpeg.telecomitalialab.com/standards/mpeg-7/mpeg-7.htm
11. Jiang, W., Wieczorkowska, A., Ras, Z.W.: Music Instrument Estimation in Polyphonic Sound Based on Short-Term Spectrum Match. In: Hassanien, A.-E., Abraham, A., de Carvalho, A. (eds.) Data Mining: Theoretical Foundations and Applications. Studies in Computational Intelligence. Springer, Heidelberg (2009)
12. Jourjine, A.N., Rickard, S.T., Yilmaz, O.: Blind separation of disjoint orthogonal signals: Demixing N sources from 2 mixtures. In: Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), pp. V-2985–V-2988 (2000)
13. Kitahara, T., Goto, M., Komatani, K., Ogata, T., Okuno, H.G.: Instrument Identification in Polyphonic Music: Feature Weighting to Minimize Influence of Sound Overlaps. EURASIP Journal on Advances in Signal Processing, Article ID 51979 (2007)
14. Lindsay, A.T., Herre, J.: MPEG7 and MPEG7 Audio—An Overview. J. Audio Engineering Society 49, 589–594 (2001)
15. Logan, B.: Mel Frequency Cepstral Coefficients for Music Modeling. In: Proceedings of 1st Annual International Symposium on Music Information Retrieval (2000)
16. Ozerov, A., Philippe, P., Gribonval, R., Bimbot, F.: One microphone singing voice separation using source adapted models. In: Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), pp. 90–93 (2005)
17. Scheirer, E., Slaney, M.: Construction and Evaluation of a Robust Multi-feature Speech/Music Discriminator. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (1997)
18. Sonic Foundry: Sound Forge. Software (2003)
19. Tzanetakis, G., Cook, P.: Musical Genre Classification of Audio Signals. IEEE Transactions Speech and Audio Processing 10, 293–302 (2002)
20. Vincent, E., Gribonval, R.: Construction d'estimateurs oracles pour la separation de sources. In: Proc. 20th GRETSI Symposium on Signal and Image Processing, pp. 1245–1248 (2005)
21. Virtanen, T., Klapuri, A.: Separation of Harmonic Sound Sources Using Sinusoidal Modeling. In: IEEE International Conference on Acoustics, Speech and Signal Processing, Istanbul, Turkey (2000)
22. Zhang, X., Marasek, K., Ras, Z.W.: Maximum Likelihood Study for Sound Pattern Separation and Recognition. In: Proceedings of International Conference on Multimedia and Ubiquitous Engineering, Seoul, Korea, April 26-28, pp. 807–812 (2007)
23. Zhang, X., Ras, Z.W.: Analysis of Sound Features for Music Timbre Recognition. In: Proceedings of the IEEE CS International Conference on Multimedia and Ubiquitous Engineering, Seoul, Korea, April 26-28, pp. 3–8 (2007) (invited paper)

Rough Sets for Solving Classification Problems in Computational Neuroscience

Tomasz G. Smolinski and Astrid A. Prinz

Department of Biology, Emory University, Atlanta, GA 30322, USA
{tomasz.smolinski,astrid.prinz}@emory.edu

Abstract. Understanding cellular properties of neurons is central in neuroscience. It is especially important in light of recent discoveries suggesting that similar neural activity can be produced by cells with quite disparate characteristics. Unfortunately, due to experimental constraints, analyzing how the activity of neurons depends on cellular parameters is difficult. Computational modeling of biological neurons allows for exploration of many parameter combinations, without the necessity of a large number of "wet" experiments. However, analysis and interpretation of often very extensive databases of models can be hard. Thus there is a need for efficient algorithms capable of mining such data. This article proposes a rough sets-based approach to the problem of classifying functional and non-functional neuronal models. In addition to presenting a successful application of the theory of rough sets in the field of computational neuroscience, we are hoping to foster with this paper a greater interest among the members of the rough sets community to explore the plethora of important problems in that field.

1 Introduction

Computational modeling of biological neurons plays an essential role in today's neuroscience research [1]. It allows for exploration of many parameter combinations and various types of neuronal activity, without requiring a prohibitively large number of "wet" experiments. This is especially important in light of recent discoveries suggesting that functional neuronal electrical activity can be produced on the basis of widely varying cellular parameter combinations [8].

The pyloric network in crustaceans (*e.g.*, lobster, crab) is one of the best-characterized neural networks in biology and a popular subject for studies of rhythmic activity in the central nervous system [10,18]. Rhythmic activity is crucial for any living organism as it is responsible for such critical actions as breathing, chewing, running, *etc.* The pyloric network consists of up to 14 neurons of 6 distinct types. The AB (anterior burster) neuron is one of the three neurons forming the pacemaker kernel which drives the rhythmic activity of the pyloric neural network, which is responsible for filtering of food in the animal. The AB neuron produces rhythmic bursts of electrical activity of a specific profile, even when isolated from other cells in the network. Figure 1 presents an example of bursting neural activity, along with an illustration of its attributes.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 620–629, 2010.

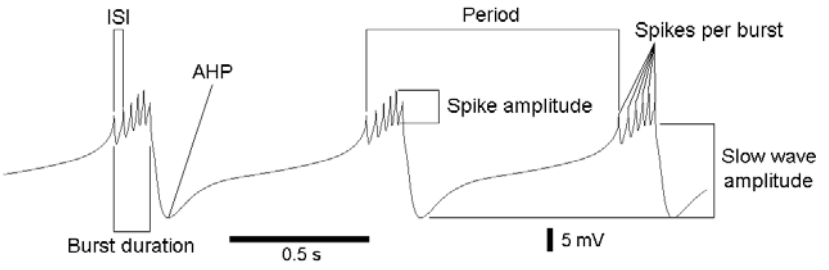


Fig. 1. An example of bursting electrical activity of the AB neuron (generated by a model). ISI stands for Inter-Spiking Interval (*i.e.*, time between spikes in a burst) and AHP stands for After-Hyperpolarization Potential (*i.e.*, trough voltage between bursts).

In a model, each part of the neuron (*e.g.*, soma–the neuron's cell body, neurites–branched projections of a neuron that conduct the electrical stimulation received from other cells, axon–the nerve fiber that conducts electrical impulses away from the cell body, *etc.*) is represented by a compartment, or a collection of compartments, each described by appropriate differential equations with a set of parameters [1]. For example, the two-compartment model shown in Fig. 2 represents the AB neuron in the conductance parameter space [22], meaning that the model is described by a set of parameters that represent the maximum membrane conductances for different ions in the neuron. The first compartment in the model represents the soma and the neurites (S/N), and the second compartment corresponds to the axon (A). The figure also shows the ionic currents determined by the membrane conductances used in the model (arrows indicate the directionality of the currents–inward vs. outward).

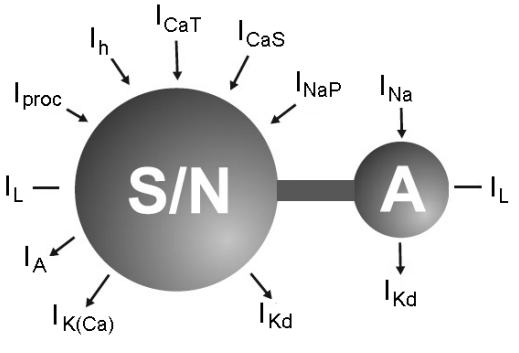


Fig. 2. A model of the AB (anterior burster) neuron of the pacemaker kernel in the crustacean pyloric neural network. S/N: Soma/Neurite compartment. A: Axon compartment. Arrows indicate inward and outward ionic currents as marked by labels (source: [22]).

To investigate the differences between functional and non-functional models of the AB cell, we created an extensive database of 21,600,000 of such models by systematically varying 12 parameters describing the model neurons (*i.e.*, maximal conductances of membrane currents, as shown in Fig. 2) from a "hand-tuned" AB model [22]. In this database, we identified those models that met all the criteria matching the observed behavior of biological AB cells, as described in the next section. The identified "good" models, as well as those which failed the test of functionality, were then subject to a rough sets-based rule-mining analysis in an attempt to explain the differences between the two groups via a set of concise and understandable IF/THEN classification rules.

We have previously applied a similar approach to the analysis of another neuron type, the PD (pyloric dilator) cell in the lobster pyloric network [19]. However, the former approach was based on the genetic algorithms-driven pseudo-association rule mining (P-ARM) methodology [11] and was concerned with generating IF/THEN rules profiling different types of neuronal activity (*e.g.*, fast spiking vs. slow spiking), rather than differentiating between functional and non-functional neurons. In addition, the P-ARM approach had one serious limitation stemming from the fact that the generated rules were based on precise values of the parameter values (*i.e.*, specific membrane conductances expressed in μS, micro-Siemens), which not only decreases the generality and applicability of the rules, but also makes them difficult to interpret biologically. The rough sets-based methodology proposed in this article overcomes this weakness.

2 Simulations and Creation of the Model Database

Our computational exploration started with a "canonical" hand-tuned model of AB, which, as reported previously, mimics the biological behavior well [22]. To investigate the extent of the variations in the parameter values that the model can withstand and still produce functional activity, we independently varied the maximal conductances of membrane currents around their canonical values. To reduce the computational time and the size of the output database, parameters were first varied one at a time to determine physiologically reasonable value ranges and step sizes for each conductance separately. The "variation matrix" of all the explored values for the parameters in the AB model is shown in Fig. 3.

Each of the 21.6 million "candidate" model neurons was simulated and classified as functional if it produced biologically realistic activity under four scenarios: spontaneous activity, spontaneous activity with neuromodulator deprivation (*i.e.*, removal of the influence of neurotransmitters descending from other parts of the nervous system), activity with external current injections, and activity with neuromodulator deprivation and current injections. Whether the activity generated by the AB models was biologically realistic was judged based on experiments performed on their biological counterparts in isolation from the rest of the

		variation in %					
soma							
g_{CaT}	4	-20	-10	0	+10		
g_{CaS}	5	-50	-25	0	+25	+50	
g_{NaP}	5	-100	-50	0	+50	+100	
g_h	3		-100	0	+150		
g_{Kd}	4		-10	0	+10	+20	
g_{KCa}	5		-50	0	+50	+100	+150
g_A	4		-100	0	+100	+200	
g_{Proc}	4	-20	-10	0	+10		
g_{leak}	3		-80	0	+80		
axon							
g_{Na}	5	-50	-25	0	+25	+50	
g_{Kd}	5		-50	0	+50	+100	+150
g_{leak}	3		-100	0	+200		

$3^3 \times 4^4 \times 5^5 = 21.6$ million combinations of AB models

Fig. 3. Explored parameter values for the AB neuron models, expressed as % deviation from the hand-tuned values (the quantity on the gray background shows the number of possible values for a given parameter)

pyloric network and under each of the four conditions [20]. There were 353,208 (1.6352% of the the entire database) models meeting all the above criteria.

Each model (*i.e.*, a particular combination of the parameter values) was coded in the database by integer numbers corresponding to the indices in the variation matrix (with 1 being the smallest possible index, and 3 always indicating the canonical value, as shown in Fig. 3). A binary classification attribute was also added to differentiate between functional and non-functional entries, thus transforming our model database into a full-fledged decision table [6].

2.1 Database Sampling

To reduce the computational complexity of our analysis, we decided to first test our approach on a sampled subset of the models. In our previous work, we investigated the impact of the sample size on the distributions of the functional and non-functional models, and determined that a 1% random sample adequately preserves the characteristics of the original dataset [21]. In addition, to deal with the problem of huge disproportion between the numbers of functional and non-functional models, we chose the following sampling protocol: first, a random 1% sample of the "good" models was selected, and then 10 random samples of the same size of the "bad" models were drawn, thus creating 10 datasets with equal distributions of the two classes, which would be subject to further analyses in parallel. This is based on a quite well known approach to balancing class distributions, especially useful in artificial neural network training [7], with existing applications in neuroscience [3].

3 Rough Sets in Classifying Functional and Non-functional Models

3.1 IF/THEN Rules and Uncertainty in the Data

One of the most natural ways to explain the differences between "good" and "bad" models could be via classification rules of the form "IF *some pattern within the parameter space*, THEN *functional model*" and "IF *some other pattern within the parameter space*, THEN *non-functional model*". The theory of rough sets (RS) lends itself naturally to this kind of analysis, especially since it is very well equipped to deal with imprecise and somewhat ambiguous data [15,9], which is a "part of life" in neuroscience. Not only can similar functional activity be produced by neurons with disparate cellular characteristics, but quite intricate interactions and relationships between the neurons' (and therefore models') parameters have been discovered [5,17]. What this means is that not only it may be difficult to identify interesting and trustworthy IF/THEN rules in our database, but also that they will most likely not be 100% accurate. In other words, even if a particular rule adequately explains the functional behavior of a subset of models, it may fail to elucidate the mechanisms governing a different subset, due to some hidden interactions characterizing that subset. The theory of rough sets by definition allows this kind of uncertainty in data, by the means of approximation of concepts via the indiscernibility relation and the equivalence classes determined upon it [6].

3.2 Discretization

As described earlier, the AB models in our database are represented by sequences of integers (*i.e.*, indices in the variation matrix), which correspond to percentages of the hand-tuned, canonical values of the maximal membrane conductances. This allows for a direct application of rough sets-based analysis, however, as discussed above, generating classification rules based on precise values of the specific membrane conductances makes for a difficult biological interpretation.

The task of discretization is to divide the domains of the attributes into a small number of discrete intervals and is commonly used in data mining [12], also in tandem with rough sets-based algorithms [13]. Discretization is usually applied to continuous data, which makes the process of analysis of such data simpler and more efficient. We propose to utilize the concept of discretization in this work, despite of the integer domain of the original data, in order to increase the biological meaningfullness of the discovered rules.

More specifically, we applied one of the simplest univariate discretization algorithms, the Equal Frequency Binning algorithm, which divides a sorted variable into k bins, where, given n instances, each bin contains m/k adjacent values [2]. We purposely set k=3 to generate bins, which we could, without a loss of too much fidelity, refer to as "low," "intermediate" (always close to the hand-tuned value), and "high" conductance, independent of the actual value in μS.

3.3 Reduction in the Number of Parameters

A "by-product" of classification rules-based analysis is the ability to identify "important" attributes in a decision table. Obviously, if a given attribute is utilized in a trustworthy rule, it must be important from the standpoint of the underlying classification problem. However, the theory of rough sets provides a more straightforward approach to the problem of selection of important features. The idea is to keep only those attributes that preserve the original indiscernibility relation and, consequently, the concept approximation. The rejected attributes are redundant since their removal cannot worsen the classification. There are usually several such subsets of attributes for a given decision table, and those that are minimal (in the sense that if we remove any of the attributes from that subset, the concept approximation accuracy will decrease) are called reducts [6].

Another very important concept related to the idea of reducts is the so-called core of reducts [6]. In basic terms, it is the set of attributes that all the discovered reducts have in common. In other words, it may be considered the smallest possible subset of attributes in a decision table that are absolutely necessary for the task of classifying objects in that table. Here, we extend the notion of the core slightly and apply it not only to reducts discovered in one particular sample of our database of models (generated as described in Sect. 2.1), but also across the samples. That way, we try to ensure that the core subset of the attributes (and thus the model parameters) is indeed important in the light of the problem of classifying functional and non-functional models, independent of the samples.

Finding reducts is not an easy task, especially in large datasets, but there exist many quite efficient algorithms that deal with this problem. In this project, we utilized the following two algorithms: 1) the well-known simple greedy Johnson's algorithm [4], which computes a single reduct only, and 2) a genetic algorithms-based implementation, which is capable of computing multiple reducts from a single dataset [23].

Not only is the idea behind searching for reducts a straightforward way to identify important attributes, but if reduction is applied *prior* to rule generation, it may also increase the clarity of discovered rules, as well as boost the computational efficiency.

4 Experiments and Results

In all the experiments performed in this project, the Rosetta system [14] along with the authors' own implementations of rough sets and genetic algorithms were employed.

As mentioned earlier, applying a reduction algorithm first may not only help determine the attributes playing an important role in a given classification problem, but significantly improve the efficiency of the rule-generating process. Therefore, we utilized both the Johnson's algorithm, as well as the genetic algorithms-based reduction across all 10 samples as the first step in our analyses. We obtained 30 reducts (20 unique ones) of the length of 10 to 11 attributes. The lack of a dramatic reduction in the number of attributes (from

the 12 attributes/parameters in the original dataset) was not surprising, as all the parameters in a conductance-based neuronal models play a role in generating the observable activity. Nevertheless, by analyzing the core of all those reducts we could try to determine which of those parameters are the most important in our classification problem. Based on the reducts computed from all 10 samples, we can state that *soma* CaT, Kd, and $Proc$, are absolutely necessary for differentiation between the functional and non-functional models (they were included in all of the reducts), the *soma* CaS, NaP, KCa, A, and *axon* Na currents are very important (utilized in over 90% of the reducts), while the leak currents (both in the soma and the axon), *soma* h, as well as the *axon* Kd current, seem to be the least important (they were used by 65%–85% of the reducts).

In the next step, we performed discretization, as described in Sect. 3.2. Since all the attributes were included in at least one reduct, and the Equal Frequency Binning algorithm is univariate (meaning that discretization is performed for each of the attributes independently of the others), we decided to discretize all the parameter values. The "low," "intermediate," and "high" conductance ranges were then used to generate a set of classification rules.

To produce a set of trustworthy rules, we employed the previously tested methodology of genetic algorithms-driven pseudo-association rule mining [19]. However, this time, since the technique was applied to discretized data, the limitation of too specific rules was dealt with. Furthermore, we only utilized those attributes that had "participated" in at least 90% of the reducts, thus significantly reducing the computational cost of the rule generation algorithm. We obtained 9 concise rules with support [s] of between 1% and 20%, and confidence [c] of at least 75% in the data. The support is the number of records with a given combination of values in the dataset, and the confidence is expressed as the ratio of the number of the records having a particular combination of values on the left-hand side of the rule *and* a given value of the classification attribute, to the total number of records with the same set of particular values on the left-hand side of the rule. In other words, the confidence expresses how sure one can be that given a set of particular values, a particular outcome will occur. Several examples of the discovered rules are shown below.

RULE 1 [s=7%, c=75%]
IF *soma* Kd is low **AND** *axon* Na is high, **THEN** functional model.

RULE 2 [s=13%, c=77%]
IF *soma* KCa is low **AND** *axon* Na is intermediate, **THEN** functional model.

RULE 3 [s=5%, c=82%]
IF *soma* CaT is intermediate **AND** *axon* Na is low, **THEN** non-functional model.

RULE 4 [s=20%, c=85%]
IF *axon* Na is low, **THEN** non-functional model.

5 Discussion

The methodology of rough sets-based classification rule mining is a useful tool for the analysis of neuronal models. It allows for an efficient exploration of the relationships between the models' parameters and their behavior. The resulting concise and comprehensible rules provide very useful insights into the problem of analysis of how the activity of neurons depends on their cellular parameters. For instance, the four rules presented above describe an intuitive dependence of the neural activity of the AB neuron on its axon's sodium (Na) current. The current is known to play a critical role in the process of spike generation, thus it makes sense that its corresponding conductance must be at least intermediate (*i.e.*, close to the hand-tuned value), as in *RULES 1 and 2*, to produce proper bursting. The *RULES 3 and 4* demonstrate the opposite situation, in which insufficient amounts of the sodium current would cause a model (or its biological counterpart) to cease being functional. Furthermore, such rules can help understand relationships between the neuronal models' parameters (and thus the cellular properties of the real cells), which is a tremendously "hot topic" in today's neuroscience. For example, *RULES 1 and 2* presents two examples in which even though the delayed-rectifier current (Kd), or the calcium-dependant potassium current (KCa) might be a little "unrepresented," the functionality will still be preserved, as long as the sodium current will "compensate" for the relative decrease in the other currents. The understanding of these kinds of phenomena is extremely important to neuroscientists.

In future work, we would like to explore other discretization and reduction algorithms in order to further improve our methodology. We would also like to apply it to other types of neurons, as well as small neural networks consisting of 2–3 cells tied together via synaptic connections. That would allow us to explore how the activity of neurons depends on their synaptic inputs, as well.

As mentioned earlier, in addition to presenting this successful application of a rough sets-based method in the area of computational neuroscience, we are sincerely hoping that with this article we will be able to foster a greater interest of the members of the rough sets (and granular computing, in general) community in the fascinating and rich field of neuroscience. There is a whole plethora of very important problems in that field that are in desperate need of efficient data mining techniques.

Acknowledgements

The authors would like to thank collaborators Farzan Nadim, Cristina Soto-Treviño, and Pascale Rabbah, who performed the physiological experiments underlying this project and provided invaluable feedback throughout the work.

References

1. Dayan, P., Abbott, L.F.: Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. MIT Press, Cambridge (2001)
2. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: Proc. of the 12th International Conference on Machine Learning, Tahoe City, CA, pp. 194–202 (1995)
3. Günay, C., Prinz, A.A.: Model calcium sensors for network homeostasis: Sensor and readout parameter analysis from a database of model neuronal networks. J. Neuroscience 30(5), 1686–1698 (2010)
4. Johnson, D.S.: Approximation algorithms for combinatorial problems. J. of Computer and System Sciences 9, 256–278 (1974)
5. Khorkova, O., Golowasch, J.: Neuromodulators, not activity, control coordinated expression of ionic currents. J. Neuroscience 27(32), 8709–8718 (2007)
6. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: A tutorial. In: Pal, S.K., Skowron, A. (eds.) Rough Fuzzy Hybridization: A New Trend in Decision-Making, pp. 3–98. Springer, Heidelberg (1999)
7. Lawrence, S., et al.: Neural network classification and prior class probabilities. In: Orr, G.B., Müller, K.-R. (eds.) NIPS-WS 1996. LNCS, vol. 1524, pp. 299–314. Springer, Heidelberg (1998)
8. Marder, E., Goaillard, J.M.: Variability, compensation and homeostasis in neuron and network function. Nature Reviews Neuroscience 7(7), 563–574 (2006)
9. Marek, W., Pawlak, Z.: Rough Sets and Information Systems. Fundamenta Matematicae 17, 105–115 (1984)
10. Miller, J.P., Selverston, A.I.: Mechanisms underlying pattern generation in lobster stomatogastric ganglion as determined by selective inactivation of identified neurons. II. Oscillatory properties of pyloric neurons. J. Neurophysiology 48(6), 1378–1391 (1982)
11. Min, H., Smolinski, T.G., Boratyn, G.M.: A genetic algorithm-based data mining approach to profiling the adopters and non-adopters of e-purchasing. In: Proc. of the 3rd International Conference on Information Reuse and Integration, Las Vegas, NV, pp. 1–6 (2001)
12. Nguyen, S.H., Nguyen, H.S.: Discretization methods in data mining. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery, pp. 451–482. Physica-Verlag, Heidelberg (1998)
13. Nguyen, S.H.: Discretization problems for rough set methods. In: Polkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS (LNAI), vol. 1424, pp. 545–552. Springer, Heidelberg (1998)
14. Øhrn, A.: ROSETTA Technical Reference Manual. Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway (2001), http://www.lcb.uu.se/tools/rosetta/materials/manual.pdf (retrieved February 26, 2010)
15. Pawlak, Z.: Rough Sets. International J. of Computer and Information Sciences 11, 341–356 (1982)
16. Prinz, A.A., Bucher, D., Marder, E.: Similar network activity from disparate circuit parameters. Nature Neuroscience (7), 1345–1352 (2004)
17. Schulz, D.J., Goaillard, J.-M., Marder, E.: Quantitative expression profiling of identified neurons reveals cell-specific constraints on highly variable levels of gene expression. PNAS 104(32), 13187–13191 (2007)

18. Selverston, A.I., Miller, J.P.: Mechanisms underlying pattern generation in lobster stomatogastric ganglion as determined by selective inactivation of identified neurons. I. Pyloric system. J. Neurophysiology 44(6), 1102–1121 (1980)
19. Smolinski, T.G., Soto-Treviño, C., Rabbah, P., Nadim, F., Prinz, A.A.: Analysis of biological neurons via modeling and rule mining. International J. of Information Technology and Intelligent Computing 1(2), 293–302 (2006)
20. Smolinski, T.G., Soto-Treviño, C., Rabbah, P., Nadim, F., Prinz, A.A.: Computational exploration of a multi-compartment model of the AB neuron in the lobster pyloric pacemaker kernel. BMC Neuroscience 9(suppl. 1), P53 (2008)
21. Smolinski, T.G., Soto-Treviño, C., Rabbah, P., Nadim, F., Prinz, A.A.: Conductance relationships in a model of the AB neuron in the lobster pyloric pacemaker kernel revealed by brute-force parameter exploration and evolutionary algorithms (in preparation)
22. Soto-Treviño, C., Rabbah, P., Marder, E., Nadim, F.: Computational model of electrically coupled, intrinsically distinct pacemaker neurons. J. Neurophysiology 94(2), 590–604 (2005)
23. Vinterbo, S., Øhrn, A.: Minimal approximate hitting sets and rule templates. International J. of Approximate Reasoning 25(2), 123–143 (2000)

Towards Approximate SQL – Infobright's Approach

Dominik Ślęzak and Marcin Kowalski

[1] Institute of Mathematics, University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
[2] Infobright Inc., Poland
Krzywickiego 34 pok. 219, 02-078 Warsaw, Poland
{slezak,mkowalski}@infobright.com

Abstract. We discuss various ideas how to implement execution of approximate SQL statements within Infobright database engine. We first outline the engine's architecture, which is designed entirely to work with standard SQL. We then discuss several possible extensions towards approximate querying and point out at some analogies with the principles of the theory of rough sets. Finally, we present the results of experiments obtained at the prototype level, both with respect to the speed of query execution and the accuracy of approximate answers.

Keywords: Analytic Database Engines, Approximate Querying, Rough Sets and Granular Computing, Infobright Community and Enterprise Editions.

1 Introduction

Infobright Community Edition[1] (ICE) and Infobright Enterprise Edition[2] (IEE) enable to run SQL statements against terabytes of data. Leveraging MySQL architecture[3] provides the users with an easy start and rich database functionality. Internal mechanisms based on data compression [24], columnar storage [12] and rough sets [15] provide performance sufficient for the data warehousing applications, with neither specialized hardware nor advanced tuning needed. The crucial aspect of ICE/IEE is partitioning data onto *packrows*, each consisting of 64K of original rows. We automatically label packrows with *rough information* about their values on data columns. We create new information systems where objects correspond to packrows and attributes correspond to various flavors of rough information. Database operations are efficiently supported within such a new framework, with the actual data accessible whenever rough information is not sufficient. Both ICE and IEE are based on a number of algorithms that apply rough information to minimize and optimize the access to compressed data [23].

Like other database vendors, Infobright stands in front of a dilemma whether standard SQL is enough. For example, in such areas of applications as, e.g., Business Intelligence[4] and Web Analytics[5], there is an ongoing debate whether the answers to SQL statements have to be always exact. The same question occurs in the case of SQL-based

[1] www.infobright.org
[2] www.infobright.com
[3] dev.mysql.com/doc/refman/6.0/en/storage-engines.html
[4] en.wikipedia.org/wiki/Business_intelligence
[5] en.wikipedia.org/wiki/Web_analytics

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 630–639, 2010.

machine learning algorithms, which are often based on heuristics, randomness and inexactness anyway [14,22]. Motivation for SQL approximations may be related also to such aspects as complexity of queries and data sources (occurring, e.g., at the edge of databases and semi-structured data analysis [3,19]), dynamically changing data with a limited access (occurring, e.g., for sensory data and data streams [6,7]), as well as huge data sets for which there is a need to monitor convergence of query execution in time, regardless of whether the final answers are to be exact or approximate [10,11].

We began to consider how to extend Infobright's framework by approximate queries in [20], pointing at some straightforward possibilities resulting from the architecture based on rough sets and granular computing [17]. In this paper, we report further discussion and experimental findings. Our main motivation is to provide faster performance at the cost of reasonably minor errors in the query answers. One of the ideas is to limit the percentage of packrows required to get accessed and to rely to a larger extent on rough information, even if it does not guarantee fully exact answers.

The paper is organized as follows: In Section 2, we provide background for our research. In Section 3, we outline the basics of Infobright. Both sections include some hints how approximate queries can be introduced into Infobright's architecture in the future. In Section 4, we consider two alternative ideas. In Section 5, we report the experimental results related to one of them. Section 6 concludes the paper.

2 Related Work

There are several categories of approximate SQL. We focus on those addressing fast inexact queries over large data. Other motivations mentioned in Section 1 are beyond the scope of this paper. The first category is based on estimating the actual answers by executing queries against data samples [5,8]. Some online forum discussions[6] show that Infobright's layer of rough information may be applied to efficient identification of collections of packrows that form statistically representative samples. Sampling may be useful also in case of standard SQL, e.g., for estimating cardinalities of intermediate results during optimization of query execution plans. Although Infobright's cardinality estimation is currently based entirely on rough information, we may extend it by additional sampling in the future, for both standard and approximate queries.

The second category is based on data synopses [1,4], particularly on histograms [6,9]. Depending on user preferences, the system may build numerous synopses for various subsets of columns and measures. Each query is appropriately translated and calculated only on synopses instead of the whole data set. The answer obtained in such a way is returned as approximation. Some of the problems with synopses are as follows: Which (subsets of) columns/measures should they describe? How to estimate answers in case several instances of synopses are applicable? How to quickly rebuild synopses in case of changing data or user preferences? One may actually interpret Infobright's rough information as a kind of data synopses, although it is designed specifically to avoid the above-mentioned problems. It is also important to note that our ideas presented in Section 4 assume the usage of both data synopses and the actual data, although the intensity of data usage is highly decreased comparing to the exact mode of SQL execution.

[6] www.infobright.org/Forums/viewthread/454/

In [11], the authors propose a framework for time-constrained SQL, wherein a user provides an upper bound for query processing time and acceptable nature of answers (partial or approximate). Similar idea was presented earlier in [10]. One can imagine an analogous framework designed for Infobright, wherein a query is executed starting with rough information and then it is gradually refined by decompressing heuristically selected pieces of data. The execution process can be then bounded by means of various parameters, such as time, acceptable errors, or percentage of data accessed.

An interesting approach to approximate SQL is introduced in [13]. The authors rely on α-*Rough-Set-Theory* [18], which is an extension of rough sets [15]. For example, for the *select* * query, the approximated answer is not bigger or not smaller than the exact one, dependent on whether lower or upper rough set approximations are in use. Also some of our ideas described in the next sections can be interpreted in the language of rough set approximations and their extensions, like those in [18] or e.g. [25]. In general, the answers to approximate SQL statements can have different syntax. For example, one can answer to a query with a description of the bounds with certainty that the actual answer is somewhere inside. One can also use a standard syntax, additionally labeled with an estimate of the answer's error. While original rough set methodology intuitively fits that former scenario, its generalizations can lead toward the latter one.

3 Infobright's Architecture

Infobright is based on grouping rows into so called *packrows*. For each packrow, it stores the values of each of columns separately, as so called *data packs* – the sets of 64K values of a single column. Data packs are compressed and described with *knowledge nodes*. Knowledge nodes form together the *Infobright's Knowledge Grid*, also referred as *rough information*, as in Section 1. Our interpretation of the concept of Knowledge Grid is different than in, e.g., Semantic Web [3], although there are some analogies in a way Infobright's Knowledge Grid *mediates* between the query engine and the data. Infobright's query optimizer implements estimation methods based on knowledge nodes instead of standard indices. It is also able to simulate the steps of query execution and approximate its final answer with no need to access data packs. One may regard it as one of the ways towards fast approximate querying, as mentioned in Section 2.

In [23], we presented various examples of using internal interface with knowledge nodes to speed up particular data operations. It is important to refer to those methods in order to better understand further sections. For illustration, let us recall how Infobright uses knowledge nodes to classify data packs into three categories:

- *Relevant (R) data packs* with all data elements relevant for further execution
- *Irrelevant (I) data packs* with no data elements relevant for further execution
- *Suspect (S) data packs* that cannot be R/I-classified based on available nodes

Inspiration to consider such three categories grew from rough sets [15], where data is split onto *positive*, *negative*, and *boundary regions* with respect to their membership to the analyzed concepts. One may say that we apply knowledge nodes to calculate *rough approximations* of data needed for resolving queries at the exact level and to assist query execution modules in accessing required data packs in an optimal way.

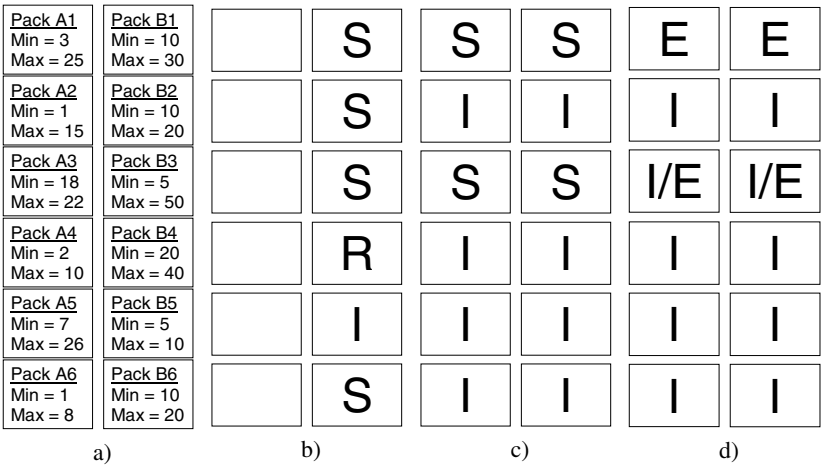


Fig. 1. Illustration for Section 3.3: (a) Min/max nodes; (b,c,d) Query execution stages. RSI denote Relevant, Suspect and Irrelevant data packs. E denotes a need of processing at the exact level.

The following case study was first introduced in [23]. Consider table T with 350,000 rows and columns A and B. We have six packrows: (A1,B1) corresponds to rows 1-65,536, (A2,B2) – to rows 65,537-131,072, etc., until (A6,B6) corresponding to rows 327,681-350,000. Consider an example of knowledge nodes that are currently used in both ICE and IEE: *min/max nodes* that contain the minimum and maximum values (interpreted specifically for different data types) for each separate data pack, as displayed in Fig. 1a. For simplicity, assume there are no nulls in T and ignore all other types of available knowledge nodes [20,21,23,24]. The query of interest is the following:

```
SELECT MAX(A) FROM T WHERE B>15;
```

According to min/max node for column B, data packs B1, B2, B3, B6 are S, B4 is R, and B5 is I (Fig. 1b). According to min/max node for column A, we obtain the following approximation: MAX(A) subject to B>15 is between 18 and 25. Thus, after re-classifying data packs and their corresponding packrows, only (A1,B1) and (A3,B3) require further investigation (Fig. 1c). The maximum in A1 is higher than in A3. Therefore, packrow (A1,B1) is the first one to be processed at the exact level. Depending on the analysis of its rows, (A3,B3) will become I or will require exact processing too (Fig. 1d).

Fig. 1 illustrates two important aspects of Infobright's technology. Firstly, knowledge nodes are used not only as a naïve technique for filtering data out – they are also applied to automatically decide in what order data should be analyzed. Secondly, one can interpret our query execution as the approximation process. In the above example, the first approximation was between 18 and 25, and the second one would be obtained after decompressing packrow (A1,B1). As mentioned in Section 2, one can extend it towards approximate query framework analogous to [10,11]. Although we do not proceed with this idea in the next sections, it is one of our future research directions.

4 Approximate Querying in Infobright

As already discussed, there are numerous ways of extending Infobright to let it deal with approximate SQL. Some of them should be already visible to a careful reader. In Subsections 4.1 and 4.2, we proceed with two more possibilities: modifying knowledge nodes and modifying the way of using knowledge nodes, respectively. Such a variety of methods – applicable both jointly and separately – shows that approximate queries can be injected into Infobright's technology whenever required in the future.

4.1 Inexact Knowledge Nodes

Consider an integer column and a data pack with minimum value 100 and maximum value 500. Imagine, however, that 99% of values in this pack are between 200 and 250. An obvious temptation is to put 200 and 250 instead of 100 and 500 into the corresponding min/max node. If we do it, the considered pack will be accessed less frequently and the query answers will be *roughly* the same. If we do it for more data packs, the average speed of queries will increase. Surely, it can be considered for other types of knowledge nodes as well. However, there are some challenges.

First of all, one needs good heuristics that create such inexact knowledge nodes (applicable optionally, exchangeably with the original ones) that minimize the frequency of decompressions but, in the same time, keep an average degree of error of query answers within a reasonable range. In Section 1, we described Infobright's approach by means of new information systems where objects correspond to packrows and attributes correspond to rough information. Thus, the task is to define such new attributes that provide more compact knowledge representation, not fully consistent with the data.

It is also crucial to estimate errors occurring at the level of data packs and to propagate them through the whole query execution process in order to provide the users with the overall expected errors. This task gets obviously more complicated along a growing complexity of analytical SQL statements and needs to be considered for all approximate query methods. With this respect, although we do not address it in this paper, one can refer query execution plans to, e.g., multi-layered approximation schemes developed within the frameworks of rough and interval computing (cf. [17]).

One can build inexact min/max nodes using a simple technique based on histograms, as illustrated by Fig. 2. An intermediate histogram is constructed for each new data pack, during data load. A specific parameter is responsible for how big fraction of values we can abandon when approximating min/max statistics, i.e., what percentage of local outliers we can cut off from a data pack's rough representation. The values are cut off from both edges of histogram to enquire the shortest interval. One can employ a straightforward greedy algorithm for choosing which edge should be cut off.

Although experimental results were quite interesting, it turns out that inexact knowledge nodes should be applied rather as a complementary technique, as they do not provide enough of query execution speed-up just by themselves. On the other hand, it is worth noting that in some applications query answers are actually more reliable when outliers are removed (cf. [7]). Thus, if we treat the process of replacing min/max with min*/max* values as a kind of metadata cleaning, we may think about a novel method for producing more robust answers with no changes to the original data.

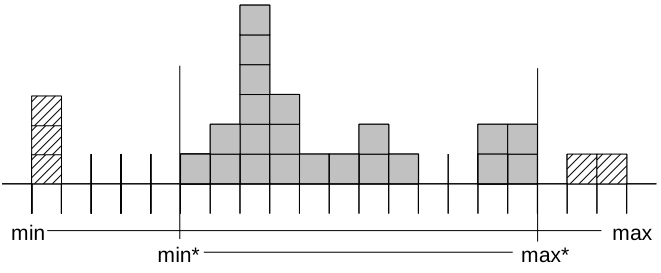


Fig. 2. Histogram representing distribution of values in a numeric data pack. Heuristically derived min*/max* values are stored in the corresponding *inexact min/max node*. For alpha-numeric values, the meaning of min* and max* can be reformulated. It is important to note that the presented histogram is only an intermediate structure, not added to Infobright's Knowledge Grid.

4.2 Randomized Classification

Consider a data pack for numeric column A. Imagine that its min and max values equal to 0 and 100, respectively. Consider SQL statement with the clause `WHERE A>1`. Intuitively, the considered data pack (and its corresponding packrow) is *almost* relevant with respect to the considered clause. Thus, one can think about approximate SQL based on so called *degrees of relevance* of data packs with respect to constraints occurring during query execution, such as SQL filters, groupings, subqueries, etc.

Degrees of relevance or, in short, rel-degrees need to be carefully defined for particular components of SQL. Their values should be derivable directly from knowledge nodes, with no data access required. Given space limitations, let us focus in detail on operators in the `WHERE` clauses. Table 1 presents two examples related to such operators. In general, we should expect that rel belongs to the unary interval. Moreover, for I- and R-classified data packs, equalities $rel = 0$ and $rel = 1$ should hold.

There are several ways of employing rel-degrees in approximate SQL. One of them is to rely entirely on knowledge nodes and rel-degrees using, e.g., fuzzy logic and related fields [16,17]. One can also follow the idea behind so called variable precision rough set model [25] and push some of *almost irrelevant/relevant* data packs into the I/R categories. An obvious motivation to do it is to limit the amount of data packs requiring decompression, which immediately results in faster query performance. This method is simple to implement at a prototype level, as it keeps the ICE/IEE internals and knowledge nodes unchanged. The only change is in R/S/I-classification.

Our framework looks as follows: We test various monotonic functions $f : [0,1] \rightarrow [0,1]$ that re-scale rel-degrees. The following equalities should hold: $f(0) = 0$, $f(1) = 1$, $f(0.5) = 0.5$. One can compare the role of f with, e.g., modifications of fuzzy memberships in [16]. We replace R/S/I with R*/S*/I*-classification, wherein R becomes R*, I becomes I*, and each data pack that was initially S-classified has a chance to change its status due to the following formula, wherein $x \in [0, 0.5]$ is random:

$$\text{S} \rightarrow \begin{cases} \text{R*} & \text{if } rel > 0.5 \text{ and } f(rel) \geq x + 0.5 \\ \text{I*} & \text{if } rel < 0.5 \text{ and } f(rel) \leq x \end{cases} \quad \text{// otherwise S} \rightarrow \text{S*} \qquad (1)$$

Table 1. Two examples of rel-degrees for operators `IS NULL` and `BETWEEN` occurring in the `WHERE` clauses. The quantities $\#NullsInPack$ and $\#ObjectsInPack$ are stored in Infobright's Knowledge Grid for each data pack. $PackRange$ is an interval based directly on min/max nodes. $ConditionRange$ refers to SQL filter based on the `BETWEEN` operator.

operator	degree of relevance $rel \in [0,1]$
`IS NULL`	$\#NullsInPack \,/\, \#ObjectsInPack$
`BETWEEN`	$\lvert PackRange \cap ConditionRange \rvert \,/\, \lvert PackRange \rvert$

Only S*-classified packs will be decompressed during query execution. The ICE/IEE-engine will omit I*-classified packs even if they contain some relevant values. It will also use knowledge nodes for R*-classified packs as if they were fully relevant.

5 Experimental Framework

We examine the prototype implemented according to the guidelines in Subsection 4.2. Given potential applications of approximate SQL, we focus on aggregate and top-k queries [2]. We use data table `fact_sales` taken from our database benchmark `car_sales` [21]. It contains 1 billion of rows. Its columns include: `sales_person` (varchar), `dealer_id` (decimal), `make_id` (decimal), `sales_commission` (decimal), and `trans_date` (date). Queries are tested with respect to the answers' errors and execution times, depending on the choice of formula for $f : [0,1] \rightarrow [0,1]$ applied in (1). One of the considered query templates looked as follows:

```
SELECT aggregate_function FROM fact_sales WHERE
trans_date between '2006-03-01' AND '2006-05-31';      (Q1)
```

The `aggregate_function` may take various forms, as illustrated by Fig. 3. It shows that some of aggregates are easier to approximate than the others. Among the easy ones, we can see `min`, `max` and `count distinct`, while the harder ones are `sum`, `avg` and `count`. Another examined query was of type top-k:

```
SELECT sales_person, SUM(sales_commission) FROM fact_sales
WHERE trans_date BETWEEN '2006-03-01' AND '2006-03-30'
AND sales_discount > 450 GROUP BY sales_person ORDER BY
SUM(sales_commission) DESC LIMIT 5;                    (Q2)
```

Here, we need to measure a *distance* between exact and approximate answers. We apply Spearman rank coefficient subject to partial knowledge about the ordering. It does not reflect errors of particular aggregate components – only their ranking. Let $top(k)$ denote the set of values in the exact answer. Let $Rank_{exact}(v)$ and $Rank_{approx}(v)$ denote the rank of value v in the exact and approximate answers. If v is not one of top-k-ranked values, we put its rank as equal to $k+1$. We used the following formula:

$$error(exact, approx) = \sqrt{\sum_{v \in top(k)} (Rank_{exact}(v) - Rank_{approx}(v))^2} \qquad (2)$$

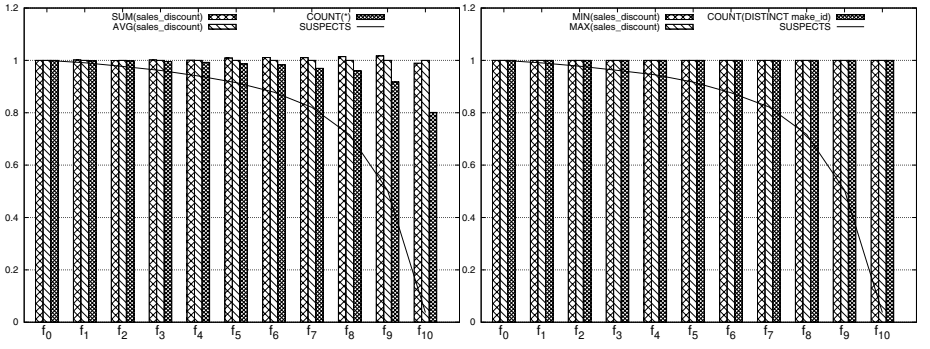


Fig. 3. Queries Q1. X-axis corresponds to the considered functions $f_0, f_1, ..., f_{10}$ that model our willingness to gain speed at the cost of precision when applying formula (1). Function f_0 yields original R/S/I-classification. Detailed definitions of f_i are omitted due to space limitations. Generally, one can see that S-classified data packs are more likely pushed to R*/I* for higher $i = 1, ..., 10$. Each query was executed 10 times for each of f_i. Y-axis reflects the average approximate query answers (normalized in order to present them all together) and the average percentages of S*-classified data packs (denoted as SUSPECTS) that require decompression.

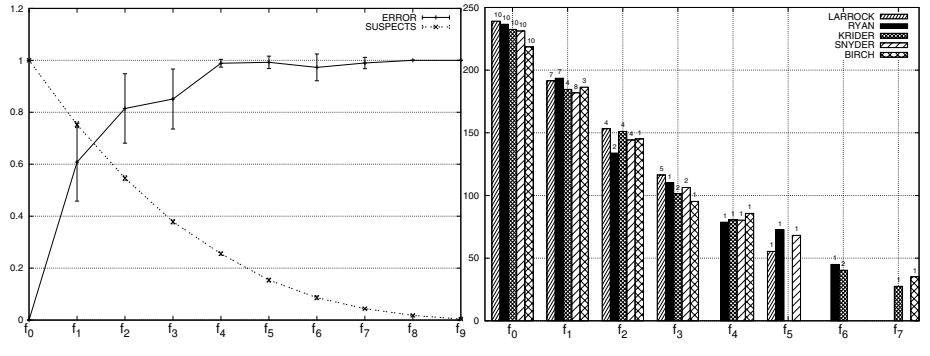


Fig. 4. Query Q2. The meaning of X-axis is the same as in Fig. 3, although here we used slightly different functions $f_0, f_1, ..., f_{10}$ (details omitted as before). At the left, Y-axis reflects the normalized quantities of $error$ defined by (2), averaged from 10 runs of the query, with the minimum and maximum obtained errors additionally marked. At the right, we report the numbers of occurrences (in 10 runs) of the actual top-5 `sales_person` values in approximate answers.

In Fig. 4, we can see that approximate answers to top-k queries are very sensitive to changes of functions $f : [0, 1] \rightarrow [0, 1]$. It may be caused by the fact that the exact aggregates computed for particular values of `sales_person` are close to each other, so even slight changes of their approximations yield quite different outcomes.

Experiments show that our approach appears to provide a good starting point for efficient SQL approximations. The achieved precision turns out satisfactory especially for simple aggregations, where the obtained answers do not differ significantly even

for functions $f : [0,1] \rightarrow [0,1]$ aiming at the highest speed-ups. On the other hand, complex queries surely need better tuning of those functions' parameters.

The method requires far more tests against real-life data sets and complex queries. One should remember that it is just a basic prototype and that it can be further improved in many ways, such as: 1) better analysis of distribution inside data packs (performed online during load), 2) applying wider range of available definitions of rel-degrees (adapted to data distributions), 3) developing a mechanism that better combines rel-degrees of multiple clauses; 4) applying similar optimizations beyond the `WHERE` clauses, etc. All these directions are on our future research roadmap.

6 Conclusion and Discussion

We proposed how to extend Infobright's architecture to handle approximate SQL. The need for such types of calculation arises in the database industry since the volumes of data have become too large for exact processing with a reasonable speed.

Given Infobright's specifics, one can address approximate querying at three levels: query execution (as summarized in the end of Section 3), rough information (Subsection 4.1), and the usage of rough information to decide which (and in what way) data packs should be processed during query execution (Subsection 4.2). Experimental results show that the approach outlined in Subsection 4.2 is quite adequate and prospective, with an interesting underlying theoretical model that adapts probabilistic generalizations of rough set principles [15,25] and extends them in a novel way by additional randomization. On the other hand, we believe that the ICE/IEE-extensions towards approximate SQL should rely on integration of all three above-listed aspects.

Among the discussed challenges, the most important one seems to relate to measuring and controlling the query answer errors. Appropriate mathematical models and further experimental tuning are required for each of the proposed approaches, when applied both separately and together. One can employ here various techniques, such as statistical analysis [8] or, e.g., appropriate extensions of interval computing [17]. It is also worth analyzing the convergence of approximations and the corresponding error estimates during the query execution process [10,11]. The good news is that most of those methods are quite naturally applicable within Infobright's framework.

Acknowledgements. This paper was partially supported by grants N N516 368334 and N N516 077837 from the Ministry of Science and Higher Education of the Republic of Poland. We would also like to thank our colleagues at Infobright Inc. for their generous help. Last but not least, we are grateful to the Infobright's online forum members for their fruitful discussions related to approximate SQL.

References

1. Beyer, K.S., Haas, P.J., Reinwald, B., Sismanis, Y., Gemulla, R.: On synopses for distinct-value estimation under multiset operations. In: SIGMOD, pp. 199–210 (2007)
2. Bruno, N., Chaudhuri, S., Gravano, L.: Top-k Selection Queries over Relational Databases: Mapping Strategies and Performance Evaluation. ACM Trans. Database Syst. 27(2) (2002)

3. Cannataro, M., Talia, D.: The knowledge grid. Commun. ACM 46(1), 89–93 (2003)
4. Chakrabarti, K., Garofalakis, M.N., Rastogi, R., Shim, K.: Approximate query processing using wavelets. In: VLDB, pp. 199–223 (2001)
5. Chaudhuri, S., Das, G., Narasayya, V.R.: Optimized stratified sampling for approximate query processing. ACM Trans. Database Syst. 32(2) (2007)
6. Cuzzocrea, A.: Top-Down Compression of Data Cubes in the Presence of Simultaneous Multiple Hierarchical Range Queries. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) ISMIS 2008. LNCS (LNAI), vol. 4994, pp. 361–374. Springer, Heidelberg (2008)
7. Deligiannakis, A., Kotidis, Y., Vassalos, V., Stoumpos, V., Delis, A.: Another outlier bites the dust: Computing meaningful aggregates in sensor networks. In: ICDE, pp. 988–999 (2009)
8. Ganti, V., Lee, M.-L., Ramakrishnan, R.: ICICLES: Self-Tuning Samples for Approximate Query Answering. In: VLDB, pp. 176–187 (2000)
9. Gibbons, P.B., Matias, Y., Poosala, V.: Fast incremental maintenance of approximate histograms. ACM Trans. Database Syst. 27(3), 261–298 (2002)
10. Hellerstein, J.M., Haas, P.J., Wang, H.J.: Online Aggregation. In: SIGMOD, pp. 171–182 (1997)
11. Hu, Y., Sundara, S., Srinivasan, J.: Supporting time-constrained SQL queries in Oracle. In: VLDB, pp. 1207–1218 (2007)
12. Kersten, M.L.: The database architecture jigsaw puzzle. In: ICDE, pp. 3–4 (2008)
13. Naouali, S., Missaoui, R.: Flexible query answering in data cubes. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2005. LNCS, vol. 3589, pp. 221–232. Springer, Heidelberg (2005)
14. Nguyen, H.S., Nguyen, S.H.: Fast split selection method and its application in decision tree construction from large databases. Int. J. Hybrid Intell. Syst. 2(2), 149–160 (2005)
15. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Inf. Sci. 177(1), 3–27 (2007)
16. Pedrycz, W.: From fuzzy sets to shadowed sets: Interpretation and computing. Int. J. Intell. Syst. 24(1), 48–61 (2009)
17. Pedrycz, W., Skowron, A., Kreinovich, V. (eds.): Handbook of Granular Computing. Wiley, Chichester (2008)
18. Quafafou, M.: alpha-RST: a generalization of rough set theory. Inf. Sci. 124(1-4), 301–316 (2000)
19. Sarawagi, S.: Information Extraction. Foundations and Trends in Databases 1(3), 261–377 (2008)
20. Ślęzak, D., Eastwood, V.: Data warehouse technology by Infobright. In: SIGMOD, pp. 841–845 (2009)
21. Ślęzak, D., Kowalski, M.: Intelligent Data Granulation on Load: Improving Infobright's Knowledge Grid. In: Lee, Y.-h., Kim, T.-h., Fang, W.-c., Ślęzak, D. (eds.) FGIT 2009. LNCS, vol. 5899, pp. 12–25. Springer, Heidelberg (2009)
22. Ślęzak, D., Sakai, H.: Automatic Extraction of Decision Rules from Non-deterministic Data Systems: Theoretical Foundations and SQL-Based Implementation. In: DTA, pp. 151–162 (2009)
23. Ślęzak, D., Wróblewski, J., Eastwood, V., Synak, P.: Brighthouse: an analytic data warehouse for ad-hoc queries. PVLDB 1(2), 1337–1345 (2008)
24. Wojnarski, M., Apanowicz, C., Eastwood, V., Ślęzak, D., Synak, P., Wojna, A., Wróblewski, J.: Method and system for data compression in a relational database. US Patent Application 2008/0071818 A1 (2008)
25. Ziarko, W.: Probabilistic approach to rough sets. Int. J. Approx. Reasoning 49(2), 272–284 (2008)

A Protein Classifier Based on SVM by Using the Voxel Based Descriptor

Georgina Mirceva, Andreja Naumoski, and Danco Davcev

Faculty of Electrical Engineering and Information Technologies,
Univ. Ss. Cyril and Methodius, Skopje, Macedonia
{georgina,andrejna,etfdav}feit.ukim.edu.mk

Abstract. The tertiary structure of a protein molecule is the main factor which determines its function. All information required for a protein to be folded in its natural structure, is coded in its amino acid sequence. The way this sequence folds in the 3D space can be used for determining its function. With the technology innovations, the number of determined protein structures increases every day, so improving the efficiency of protein structure retrieval and classification methods becomes an important research issue. In this paper, we propose a novel protein classifier. Our classifier considers the conformation of protein structure in the 3D space. Namely, our voxel based protein descriptor is used for representing the protein structures. Then, the Support Vector Machine method (SVM) is used for classifying protein structures. The results show that our classifier achieves 78.83% accuracy, while it is faster than other algorithms with comparable accuracy.

Keywords: PDB, SCOP, protein classification, voxel descriptor, Support Vector Machine (SVM).

1 Introduction

Proteins are one of the most important molecules in the living organisms, since they play a vital functional role in living organisms. All information required for a protein to be folded in its natural structure is coded in its amino acid sequence. The way this sequence folds in the 3D space is very important, in order to understand the function of the protein molecule. The knowledge of the protein function is crucial in the development of new drugs, better crops, and development of synthetic biochemical.

Since determining of the first protein structure of the myoglobin, up to now, the complexity and the variety of the protein structures has increased, as the number of the new determined macromolecules has. Therefore, a need for efficient methods for classification of proteins is obvious, which may result in a better understanding of protein structures, their functions, and the evolutionary procedures that led to their creation. Many classification schemes and databases, such as CATH [1], FSSP [2] and SCOP [3], have been developed in order to describe the similarity between proteins.

The Structural Classification of Proteins (SCOP) database [3] describes the evolutionary relationships between proteins. SCOP has been accepted as the most relevant and the most reliable classification dataset [4], due to the fact that it is based on visual

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 640–648, 2010.

observations of the protein structures made by human experts. In SCOP, proteins are classified in hierarchical manner. The main levels of the SCOP hierarchy are Domain, Family, Superfamily, Fold, and Class. Due to its manual classification methods, the number of proteins released in PDB database which have not been classified by SCOP yet, drastically increases. So, the necessity of fast, accurate and automated algorithms for protein classifications is obvious.

One way to determine protein similarity is to use sequence alignment algorithms like Needleman–Wunch [5], BLAST [6], PSI-BLAST [7] etc. Since these methods cannot recognize proteins with remote homology, we can use structure alignment methods such as CE [8], MAMMOTH [9] and DALI [10]. In general, these methods are accurate, but their speed of classification is always questioned. For example, CE takes 209 days [8] to classify 11.000 novel protein structures. Also, there are numerous methods, like SCOPmap [11] and FastSCOP [12], which combine sequence and structure alignment of the proteins.

Classification of protein structures can be done without applying alignment techniques. Namely, proteins can be mapped in a feature space, and then some classification method can be used. In [13], some local and global features are extracted from the distance matrix histograms. Classification is based on the E-predict algorithm [13]. In [14], some features of the protein sequence are extracted, and then proteins are classified by using Naive Bayes and boosted C4.5 decision trees.

In this paper, we propose a novel protein classifier. Our voxel based descriptor [15] is used to represent the protein molecules in the feature space. After proper mapping of the protein structures in the feature space, the Support Vector Machine method (SVM) [16] is used to classify the protein structures. A part of the SCOP 1.73 database was used in the evaluation of our classifier.

The rest of the paper is organized as follows: our protein classifier is presented in section 2; section 3 presents some experimental results; while section 4 concludes the paper.

2 Our Classifier

In this paper, we propose an accurate and fast system that allows to the user to classify protein structures. The information about protein structure is stored in PDB files. The PDB files are stored in the Protein Data Bank (PDB) [17], which is the primary depository of experimentally determined protein structures. They contain information about primary, secondary and tertiary structure of proteins. We have used our voxel based descriptor [15] in order to map the protein structure in the feature space. Then, SVM classifier [16] is applied in order to classify each newly protein in corresponding protein domain in the SCOP hierarchy.

Our goal is to provide a system which provides structural classification of protein structures. The phases of our classification system are illustrated on Fig. 1. In the training phase, the information about the protein structure contained in PDB file is processed and the voxel based descriptor is extracted. After generation of the voxel descriptors of all training proteins, a SVM model for each SCOP domain is generated by the SVM method. In the testing phase, the user uploads the PDB file of the query protein. The information from the PDB file is processed and the voxel based descriptor is extracted. Then, the protein is classified according to the SVM method.

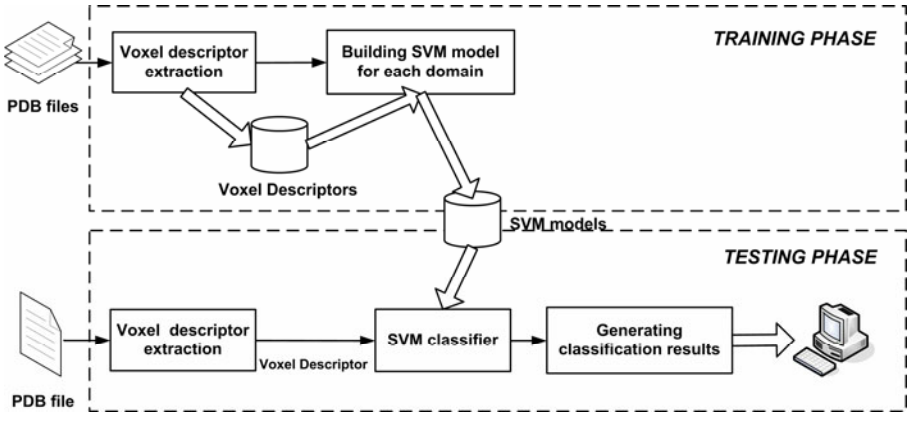


Fig. 1. The training and testing phase of our SVM based classification system

2.1 Voxel Based Descriptor

We have used the voxel-based descriptor presented in [18] to extract the geometrical features of the protein structures. Voxel based descriptor is extracted in five phases. Since the exact 3D position of each atom and its radius are known, it may be represented by a sphere. First, we perform triangulation in order to build a mesh model of the protein structure. The surface of each sphere is triangulated, thus forming the mesh model of the protein. Then, the centre of mass is calculated and the protein is translated, so the new centre of mass is at the origin. The distance d_{max} between the new origin and the most distant vertex is computed, and protein is scaled, so d_{max}=1. In this way, we provide translation and scale invariance.

After triangulation, we perform voxelization. Voxelization transforms the continuous 3D-space, into discrete 3D voxel space. The voxelization proceeds in two steps: discretization and sampling. Discretization divides the continuous 3D-space into voxels. With sampling, depending on the positions of the polygons of the 3D-mesh model, to each voxel v_{abc}, a value is attributed equal to the fraction of the total surface area S of the mesh which is inside the region μ_{abc} (1).

$$v_{abc} = \frac{area\{\mu_{abc} \cap I\}}{S}, 0 \le a,b,c \le N - 1. \quad (1)$$

Each triangle T_j of the model is subdivided into p_j^2 coincident triangles each of which has the surface area equal to $\delta = S_j / p_j^2$, where S_j is the area of T_j . If all vertices of the triangle T_j lie in the same cuboid region μ_{abc}, then we set $p_j = 1$, otherwise we use (2) to determine the value of p_j.

$$p_j = \left\lceil \sqrt{p_{\min} \frac{S_j}{S}} \right\rceil \quad (2)$$

For each newly obtained triangle, the center of gravity G is computed, and the voxel μ_{abc} is determined. Finally, the attribute v_{abc} is incremented by δ. The quality of approximation is set by p_{min}. According to [18], we have set p_{min} =32000.

The information contained in the voxel grid can be processed further to obtain both correlated information and more compact representation of the voxel attributes as a feature. We applied the 3D Discrete Fourier Transform (3D-DFT) to obtain a spectral domain feature vector, which provides rotation invariance of the descriptor. A 3D-array of complex numbers $F = [f_{abc}]$ is transformed into another 3D-array by (3).

$$f'_{pqs} = \frac{1}{\sqrt{MNP}} \sum_{a=0}^{M-1} \sum_{b=0}^{N-1} \sum_{c=0}^{P-1} f_{abc} e^{-2\pi j(ap/M + bq/N + cs/P)} \tag{3}$$

Since we apply the 3D-DFT to a voxel grid with real-valued attributes, we shift the indices so that (a; b; c) is translated into (a–M/2; b–N/2; c–P/2). Let M=N=P and we introduce the abbreviation (4).

$$v'_{a-M/2,b-N/2,c-P/2} \equiv v_{abc} \tag{4}$$

We take the magnitudes of the low-frequency coefficients as components of the vector. Since the 3D-DFT input is a real-valued array, the symmetry is present among obtained coefficients, so the feature vector is formed from all non-symmetrical coefficients which satisfy $1 \le |p| + |q| + |s| \le k \le N/2$. We form the feature vector by the scaled values of f'_{pqs} by dividing by $|f'_{000}|$. This vector presents the geometrical features of the protein structure.

Additionally, some features of the primary and secondary structure of the protein molecule are considered, as in [19]. More specifically, concerning the primary structure, the ratios of the amino acids' occurrences and hydrophobic amino acids ratio are calculated. Concerning the secondary structure, the ratios of the helix types' occurrences, the number of Helices, Sheets and Turns in the protein are also calculated. These features are incorporated in the previously extracted geometry descriptor, thus forming better integrated descriptor.

In this way, we transform the protein tertiary structures into N dimensional feature space. Then, classification process follows, where voxel based descriptors are used as representatives of protein structures.

2.2 Support Vector Machine (SVM) Method

The support vector machine is a binary classification method proposed by Vapnik and his colleagues at Bell laboratories [16], [20]. As a binary problem, it has to find the optimal hyperplane which separates the positive from negative examples, see Fig. 2. Examples are presented as data points: $\{\mathbf{x}_i, y_i\}$, i=1,...,N , $y_i \in \{-1, 1\}$, $\mathbf{x}_i \in R^d$. In our approach, $\mathbf{x}$ corresponds to the voxel descriptor of the i-th training protein. The points $\mathbf{x}$ which lie on the hyperplane satisfy $\mathbf{w}\cdot\mathbf{x} + b = 0$, where $\mathbf{w}$ is normal to the hyperplane, $|b|/\|\mathbf{w}\|$ is the distance from the hyperplane to the origin, and $\|\mathbf{w}\|$ is the Euclidean norm of $\mathbf{w}$. The "margin" of a separating hyperplane is defined as a sum of the distances from the separating hyperplane to the closest positive and negative examples. Suppose that all the training examples satisfy the constraints (5), so they can be combined as an inequality (6).

$$\begin{aligned} \mathbf{x}_i * \mathbf{w} + b \geq +1, \quad & \text{for } y_i = +1 \\ \mathbf{x}_i * \mathbf{w} + b \leq -1, \quad & \text{for } y_i = -1 \end{aligned} \tag{5}$$

$$y_i(\mathbf{x}_i * \mathbf{w} + b) - 1 \geq 0, \quad \forall i \tag{6}$$

The points which satisfy the equality (6) lie on the two hyperplanes H_1 and H_2. These hyperplanes are parallel and distinguish the positive from negative examples. So, the goal is to find a pair of hyperplanes which gives the maximum margin by minimizing $\|\mathbf{w}\|^2$, according to (5). The model will contain only examples that lie on the separating hyperplanes, named support vector machines.

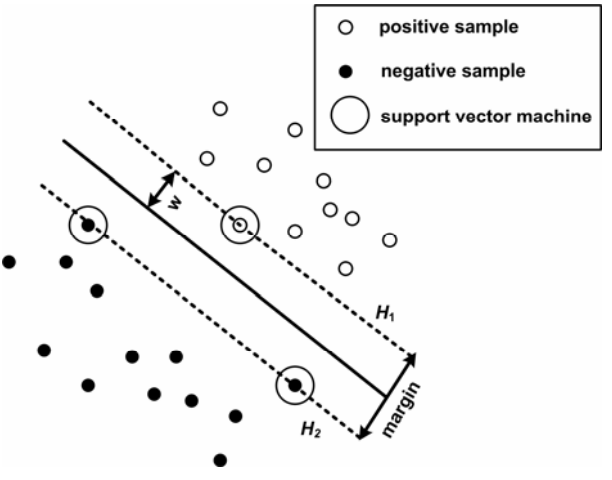


Fig. 2. A separable hyperplane for two dimensional feature space

Nonnegative Lagrange multipliers α_i are introduced for each example. In this way, primal Lagrangian gets the form (7).

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i y_i (\mathbf{x}_i * \mathbf{w} + b) + \sum_{i=1}^{N} \alpha_i \tag{7}$$

Then, we have to minimize L_P with respect to $\mathbf{w}$, b, and maximize with respect to all α_i at the same time. This is a convex quadratic programming problem, since the function is itself convex, and those points which satisfy the constraints form a convex set. This means that we can equivalently solve the following "dual" problem: maximize L_P, subject to the constraints that the gradient of L_P with respect to $\mathbf{w}$ and b vanish, and subject also to the constraints that the $\alpha_i \geq 0$. This gives the conditions (8). Then, (8) is substituted into (7), which leads to (9). L_P and L_D show the Lagrangian which arise from the same objective function, but under different constraints. In this way, the problem can be solved by minimizing L_P or by maximizing L_D.

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \,, \qquad \sum_i \alpha_i y_i = 0 \tag{8}$$

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i * \mathbf{x}_j \tag{9}$$

This algorithm is not suitable for a non-separable data, since it will find no feasible solution. So, positive slack variables e_i, i=1,…,N are introduced in (5), thus forming constraints (10).

$$\begin{aligned} &\mathbf{x}_i * \mathbf{w} + b \geq +1 - e_i, \quad \text{for } y_i = +1 \\ &\mathbf{x}_i * \mathbf{w} + b \leq -1 + e_i, \quad \text{for } y_i = -1 \end{aligned} \qquad e_i \geq 0, \ \forall i \tag{10}$$

An extra cost for an error is assigned, so the objective function to be minimized will be $\|\mathbf{w}\|^2/2 + C(\Sigma_i e_i)$ instead $\|\mathbf{w}\|^2/2$. The parameter C is defined by the user, where larger C corresponds to a higher penalty to the errors.

In order to generalize the above method to be applicable for a non-separable problem, the data should be mapped into other feature space H, by using a mapping Φ, thus getting linearly separable problem. The training algorithm would only depend on the data through dot products in H. Now, if there were a kernel function K such that $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$, we would only need to use K in the training, and would never need to explicitly know what Φ is. One example for this function is Gaussian, given by (11), where σ is the standard deviation.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 / 2\sigma^2) \tag{11}$$

In the test phase, the sign of (12) is computed, where s_i correspond to the support vectors. So, we can avoid computing $\Phi(\mathbf{x})$ explicitly, and use $K(\mathbf{s}_i, \mathbf{x}) = \Phi(\mathbf{s}_i) \cdot \Phi(\mathbf{x})$.

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \Phi(\mathbf{s}_i) * \Phi(\mathbf{x}) + b = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{s}_i, \mathbf{x}) + b \tag{12}$$

Although the SVM method is originally proposed as a binary classifier, there are many approaches that perform multi-class classification [21], but are computationally much more expensive than solving several binary problems. On the other hand, many approaches decompose the multi-class problem into several binary problems, thus leading to faster classifier.

One possible approach is one-against-all (OvA), where N separate classifiers are constructed. On the other hand, one-against-one (OvO) approach can be used by building a separate classifier for each pair of classes, thus leading to $N(N\text{-}1)/2$ classifiers. In this research, our dataset contains proteins from 150 classes, so we used OvA algorithm, thus leading to 150 classifiers, instead of 11175 classifiers in OvO case.

3 Experimental Results

We have implemented a system for protein classification based on the SVM method. Part of SCOP 1.73 database was used. Our standard of truth data contains 6979 randomly selected protein chains from the 150 most populated protein SCOP domains. 90% of the data set serves as training data and the other 10% serves as test data.

First, we examined the influence of standard deviation σ on the classification accuracy. In Table1, the classification accuracies are presented by using different values for σ. We tested the influence of σ on the classification accuracy on the training and test set.

Table 1. The influence of the standard deviation σ on the classification accuracy

Standard deviation (σ)	Classification accuracy (%) on training set	Classification accuracy (%) on test set	Standard deviation (σ)	Classification accuracy (%) on training set	Classification accuracy (%) on test set
1650	/	70.20	8000	98.48	78.01
3000	99.98	74.76	8500	98.14	78.50
4000	99.98	75.09	9000	97.69	77.69
5000	99.98	76.06	10000	96.76	78.34
6000	99.98	76.06	12000	94.67	78.18
7000	99.44	77.04	15000	91.72	76.88
7500	99.04	77.04	20000	76.68	75.73

Analysis showed that for small value of σ, the training phase lasts longer, and leads to over-fitting of the classifier. So, for small value of σ, we achieve high classification accuracy by using the training data in the test phase. On the other hand, by decreasing the standard deviation, when the test set is used in the testing phase, the classification accuracy is getting worse (70.2% classification accuracy for σ=1650). By increasing σ, the classification accuracy on the training data decreases due to the inability of the classifier to suits to the data so well. On the other hand, for higher values of σ, the classification accuracy on test data increases. Table 2 present the experimental results of more detailed analysis of the influence of the standard deviation on the classification accuracy on the test data. Further analysis can be performed in order to find the optimal value of the standard deviation.

Table 2. The influence of the standard deviation σ on the classification accuracy on the test set

Standard deviation (σ)	Classification accuracy (%)	Standard deviation (σ)	Classification accuracy (%)	Standard deviation (σ)	Classification accuracy (%)
1650	70.20	8500	78.50	103000	77.69
3000	74.76	8550	78.34	104000	78.01
4000	75.09	8600	78.34	105000	78.01
5000	76.06	8750	77.85	106000	77.85
6000	76.06	9000	77.69	107000	77.52
7000	77.04	9100	78.18	108000	77.69
7500	77.04	9200	78.18	109000	78.01
8000	78.01	9300	78.01	110000	78.18
8250	78.34	9400	78.01	120000	78.18
8300	78.01	9500	78.34	150000	76.87
8350	78.50	100000	78.34	200000	75.73
8400	78.50	101000	77.85		
8450	78.50	102000	77.85		

Further, we examined the influence of the penalty given to the errors c. The analysis is performed for the best values of σ (according to Table 2). Experimental results presented in Table 3 show that the error penalty c has minor influence on the classification accuracy.

Table 3. The influence of the error penalty c on the classification accuracy for distinct values of σ

c	10	25	35	40	45	50	55	60	75	100
σ =8350	76.71	77.69	77.69	78.18	78.50	78.83	78.34	78.01	78.18	78.50
σ =8400	76.71	77.36	77.85	78.34	78.66	78.66	78.50	78.01	78.18	78.50
σ =8450	76.71	77.36	77.85	78.18	78.50	78.66	78.66	78.18	78.01	78.50
σ =8500	76.55	77.36	77.85	78.18	78.34	78.66	78.66	78.34	78.34	78.50

As it can be seen from Table 3, our approach achieves 78.83% classification accuracy for σ =8350 and c=50. The training phase lasts several minutes, while the test phase takes several seconds. Compared to other classification algorithms with comparable accuracy, our approach has shown as much faster.

4 Conclusion

In this paper we proposed a novel approach for classifying protein tertiary structures based on the Support Vector Machine (SVM) method. Our voxel based descriptor was used as representatives of the protein structures in the feature space. After proper transformation of the protein structures into the feature space, a SVM classifier is used in order to build a separate SVM for each SCOP domain.

A part of SCOP 1.73 database was used to evaluate the proposed classification approach. We investigated the influence of the standard deviation and the error penalty on the classification accuracy. The results showed that the error penalty has minor influence on the accuracy, while the standard deviation drastically affects the adequacy of the classifier, so leading to high influence on the classification accuracy. Further analysis can be made in order to find the optimal value of the standard deviation. Also, an automatic adjustment of the standard deviation can be made, thus leading to faster training. The proposed approach achieves 78.83% classification accuracy. Compared to other classification algorithms with comparable accuracy, our approach has shown as much faster.

We have already investigated our protein ray based descriptor which has shown as an accurate, simple and fast way for representation of protein structures. Since the average precision of the voxel based descriptor is 77.8% and the average precision of the ray descriptor is 92.9%, we expect that similar SVM classifier based on the ray descriptor will achieve much higher precision. Also, due to the lower dimensionality of the ray descriptor, we expect that the ray based classifier will be faster than the proposed voxel based classifier.

References

1. Orengo, C.A., Michie, A.D., Jones, D.T., Swindells, M.B., Thornton, J.M.: CATH - A hierarchic classification of protein domain structures. Structure 5(8), 1093–1108 (1997)
2. Holm, L., Sander, C.: The FSSP Database: Fold Classification Based on Structure-Structure Alignment of Proteins. Nucleic Acids Research 24(1), 206–209 (1996)
3. Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C.: Scop: a structural classification of proteins database for the investigation of sequences and structures. Journal of Molecular Biology 247(4), 536–540 (1995)
4. Camoglu, O., Can, T., Singh, A.K., Wang, Y.F.: Decision tree based information integration for automated protein classification. Journal of Bioinformatics and Computational Biology 3(3), 717–742 (2005)
5. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 48(3), 443–453 (1970)
6. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. Journal of Molecular Biology 215(3), 403–410 (1990)
7. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research 25(17), 3389–3402 (1997)
8. Shindyalov, H.N., Bourne, P.E.: Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Engineering 11(9), 739–747 (1998)
9. Ortiz, A.R., Strauss, C.E., Olmea, O.: MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. Protein Science 11(11), 2606–2621 (2002)
10. Holm, L., Sander, C.: Protein structure comparison by alignment of distance matrices. Journal of Molecular Biology 233(1), 123–138 (1993)
11. Cheek, S., Qi, Y., Krishna, S.S., Kinch, L.N., Grishin, N.V.: SCOPmap: Automated assignment of protein structures to evolutionary superfamilies. BMC Bioinformatics 5, 197–221 (2004)
12. Tung, C.H., Yang, J.M.: fastSCOP: a fast web server for recognizing protein structural domains and SCOP superfamilies. Nucleic Acids Research 35, W438–W443 (2007)
13. Chi, P.H.: Efficient protein tertiary structure retrievals and classifications using content based comparison algorithms. Ph.D. Thesis. University of Missouri-Columbia (2007)
14. Marsolo, K., Parthasarathy, S., Ding, C.: A Multi-Level Approach to SCOP Fold Recognition. In: IEEE Symposium on Bioinformatics and Bioengineering, pp. 57–64 (2005)
15. Mirceva, G., Kalajdziski, S., Trivodaliev, K., Davcev, D.: Comparative analysis of three efficient approaches for retrieving protein 3D structures. In: 4th Cairo International Biomedical Engineering Conference (CIBEC 2008), Cairo, Egypt (2008)
16. Vapnik, V.: The Nature of Statistical Learning Theory, 2nd edn. Springer, New York (1999)
17. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The Protein Data Bank. Nucleic Acids Research 28, 235–242 (2000)
18. Vranic, D.V.: 3D Model Retrieval. Ph.D. Thesis. University of Leipzig (2004)
19. Daras, P., Zarpalas, D., Axenopoulos, A., Tzovaras, D., Strintzis, M.G.: Three-Dimensional Shape-Structure Comparison Method for Protein Classification. IEEE/ACM Transactions on Computational Biology and Bioinformatics 3(3), 193–207 (2006)
20. Burges, C.J.C.: A tutorial on support vector machine for pattern recognition. Data Mining and Knowledge Discovery 2(2), 121–167 (1998)
21. Weston, J., Watkins, C.: Multi-class support vector machines. In: European Symposium on Artificial Neural Networks, ESANN 1999 (1999)

Explicit Neural Network-Based Nonlinear Predictive Control with Low Computational Complexity

Maciej Ławryńczuk

Institute of Control and Computation Engineering, Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
Tel.: +48 22 234-76-73
M.Lawrynczuk@ia.pw.edu.pl

Abstract. This paper describes a nonlinear Model Predictive Control (MPC) algorithm based on neural models. Two neural models are used on-line: from a dynamic model the free trajectory (the influence of the past) is determined, the second neural network approximates the time-varying feedback law. In consequence, the algorithm is characterised by very low computational complexity because the control signal is calculated explicitly, without any on-line optimisation. Moreover, unlike other suboptimal MPC approaches, the necessity of model linearisation and matrix inversion is eliminated. The presented algorithm is compared with linearisation-based MPC and MPC with full nonlinear optimisation in terms of accuracy and computational complexity.

Keywords: Process control, Model Predictive Control, neural networks, optimisation, soft computing.

1 Introduction

Model Predictive Control (MPC) refers to a control strategy in which an explicit model is used to predict future behavior of the process over some horizon and to optimise the future control action [9,16]. Because the model is used for prediction and optimisation, MPC algorithms, unlike any other control technique, can take into account constraints imposed on process inputs (manipulated variables) and outputs (controlled variables), which usually decide on quality, economic efficiency and safety. Moreover, MPC can be efficiently used for multivariable processes, with many inputs and outputs. As a result, MPC algorithms have been successfully used for years in advanced industrial applications [15].

In the simplest case a linear model is used for prediction. Unfortunately, many technological processes are in fact nonlinear. In such cases MPC based on a linear model may be inefficient (slow) or inappropriate (unstable). That is why a wide variety of nonlinear MPC approaches have been developed [4,11,16]. In particular, MPC algorithms based on neural models are recommended [6,7,8,12,16,17]. It is because neural models can be efficiently used on-line in MPC since they have excellent approximation abilities, a limited number of parameters (when compared to other model types) and simple structures. Furthermore, neural models

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 649–658, 2010.

directly describe input-output relations of process variables, complicated systems of algebraic and differential equations do not have to be solved on-line as it necessary in MPC based on fundamental models.

Usually, suboptimal nonlinear MPC algorithms are implemented in practice [6,8,12,16,17]. The nonlinear model (e.g. neural) is linearised on-line. Thanks to it, the control action is calculated at each iteration from an easy to solve quadratic programming task. Recently, a few approaches have been proposed to reduce the computational complexity of nonlinear MPC. An approximate MPC technique can be used which replaces the whole algorithm, the neural network directly calculates the control signal without any optimisation [1,2,14]. An alternative is to use an explicit piecewise linear state feedback approximator which can be found off-line using multi-parametric nonlinear programming (mp-NLP) [5]. The controller is realised by binary tree search, but complexity of trees may be significant. A yet another approach is to use a neural network to solve on-line the MPC optimisation problem [13].

In this paper a computationally efficient neural network approach to nonlinear MPC is detailed. The control action is determined explicitly, without any on-line optimisation. Hence, the algorithm can be used for very fast processes or implemented on simple hardware. Two neural models are used on-line: from a dynamic model the free trajectory (the influence of the past) is determined, the second neural network approximates the time-varying feedback law. Unlike other explicit MPC approaches [7], the necessity of model linearisation and matrix inversion is eliminated.

2 Model Predictive Control Algorithms

In MPC algorithms [9,16] at each consecutive sampling instant k, $k = 0, 1, 2, \ldots$, a set of future control increments is calculated

$$\triangle \boldsymbol{u}(k) = [\triangle u(k|k) \ \triangle u(k+1|k) \ldots \triangle u(k+N_{\mathrm{u}}-1|k)]^{\mathrm{T}} \tag{1}$$

It is assumed that $\triangle u(k+p|k) = 0$ for $p \geq N_{\mathrm{u}}$, where N_{u} is the control horizon. The objective of the algorithm is to minimise differences between the reference trajectory $y^{\mathrm{ref}}(k+p|k)$ and predicted outputs values $\hat{y}(k+p|k)$ over the prediction horizon $N \geq N_{\mathrm{u}}$, i.e. for $p = 1, \ldots, N$. The cost function is usually

$$J(k) = \sum_{p=1}^{N} (y^{\mathrm{ref}}(k+p|k) - \hat{y}(k+p|k))^2 + \sum_{p=0}^{N_{\mathrm{u}}-1} \lambda_p (\triangle u(k+p|k))^2 \tag{2}$$

where $\lambda_p > 0$ are weighting coefficients. Only the first element of the determined sequence (1) is applied to the process, i.e. $u(k) = \triangle u(k|k) + u(k-1)$. At the next sampling instant, $k+1$, the prediction is shifted one step forward and the whole procedure is repeated.

Predictions $\hat{y}(k+p|k)$ are calculated from a dynamic model of the process. For this purpose different model structures can be used [16]. In particular, neural models based on MLP and RBF networks are recommended [6,7,8].

3 Explicit Neural Network-Based Nonlinear MPC

Let the dynamic process under consideration be described by the following discrete-time Nonlinear Auto Regressive with eXternal input (NARX) model

$$y(k) = f(\boldsymbol{x}(k)) = f(u(k-\tau), \ldots, u(k-n_{\mathrm{B}}), y(k-1), \ldots, y(k-n_{\mathrm{A}})) \tag{3}$$

where $f\colon \mathbb{R}^{n_{\mathrm{A}}+n_{\mathrm{B}}-\tau+1} \to \mathbb{R}$ is a nonlinear function which describes the model, integers n_{A}, n_{B}, τ define the order of dynamics, $\tau \leq n_{\mathrm{B}}$. In computationally efficient MPC approaches a linear approximation of the nonlinear model (3)

$$y(k) = \sum_{l=\tau}^{n_{\mathrm{B}}} b_l(k)u(k-l) - \sum_{l=1}^{n_{\mathrm{A}}} a_l(k)y(k-l) \tag{4}$$

is used on-line for calculation of the future control policy (1). Coefficients $a_l(k)$ and $b_l(k)$ are calculated on-line [6,7,8,12,16,17].

3.1 Control Action Calculation

The MPC cost function (2) can be expressed in a compact form

$$J(k) = \left\|\boldsymbol{y}^{\mathrm{ref}}(k) - \hat{\boldsymbol{y}}(k)\right\|^2 + \|\triangle\boldsymbol{u}(k)\|^2_{\boldsymbol{\Lambda}} \tag{5}$$

where

$$\boldsymbol{y}^{\mathrm{ref}}(k) = \left[y^{\mathrm{ref}}(k+1|k) \ldots y^{\mathrm{ref}}(k+N|k)\right]^{\mathrm{T}} \tag{6}$$

$$\hat{\boldsymbol{y}}(k) = \left[\hat{y}(k+1|k) \ldots \hat{y}(k+N|k)\right]^{\mathrm{T}} \tag{7}$$

are vectors of length N, $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_0, \ldots, \lambda_{N_{\mathrm{u}}-1})$. Hence, the MPC optimisation problem, the solution to which gives current and future control action (1), is

$$\min_{\triangle\boldsymbol{u}(k)} \left\{ J(k) = \left\|\boldsymbol{y}^{\mathrm{ref}}(k) - \hat{\boldsymbol{y}}(k)\right\|^2 + \|\triangle\boldsymbol{u}(k)\|^2_{\boldsymbol{\Lambda}} \right\} \tag{8}$$

It can be shown [6] that if the linear approximation (4) of the original nonlinear model (3) is used for prediction in MPC, the output prediction vector is

$$\hat{\boldsymbol{y}}(k) = \boldsymbol{G}(k)\triangle\boldsymbol{u}(k) + \boldsymbol{y}^0(k) \tag{9}$$

The output prediction is expressed as the sum of a forced trajectory, which depends only on the future (on future control moves $\triangle\boldsymbol{u}(k)$) and a free trajectory $\boldsymbol{y}^0(k) = \left[y^0(k+1|k) \ldots y^0(k+N|k)\right]^{\mathrm{T}}$, which depends only on the past. The dynamic matrix $\boldsymbol{G}(k)$ of dimensionality $N \times N_{\mathrm{u}}$ contains step-response coefficients of the linearised model (4). It is calculated on-line taking into account the current state of the process [6,8,12,16,17].

Taking into account the suboptimal prediction equation (9), the MPC optimisation problem (8) becomes

$$\min_{\triangle\boldsymbol{u}(k)} \left\{ J(k) = \left\|\boldsymbol{y}^{\mathrm{ref}}(k) - \boldsymbol{G}(k)\triangle\boldsymbol{u}(k) - \boldsymbol{y}^0(k)\right\|^2 + \|\triangle\boldsymbol{u}(k)\|^2_{\boldsymbol{\Lambda}} \right\} \tag{10}$$

Since the minimised cost function $J(k)$ is quadratic, the unique solution is obtained by equating its first-order derivative

$$\frac{\mathrm{d}J(k)}{\mathrm{d}\triangle \boldsymbol{u}(k)} = -2\boldsymbol{G}^{\mathrm{T}}(k)(\boldsymbol{y}^{\mathrm{ref}}(k) - \boldsymbol{G}(k)\triangle \boldsymbol{u}(k) - \boldsymbol{y}^{0}(k)) + 2\boldsymbol{\Lambda}\triangle \boldsymbol{u}(k) \tag{11}$$

to a zeros vector of length N_{u}. Optimal control moves are

$$\triangle \boldsymbol{u}(k) = \boldsymbol{K}(k)(\boldsymbol{y}^{\mathrm{ref}}(k) - \boldsymbol{y}^{0}(k)) \tag{12}$$

where

$$\boldsymbol{K}(k) = (\boldsymbol{G}^{\mathrm{T}}(k)\boldsymbol{G}(k) + \boldsymbol{\Lambda})^{-1}\boldsymbol{G}^{\mathrm{T}}(k) \tag{13}$$

is a matrix of dimensionality $N_{\mathrm{u}} \times N$. As a result, one obtains a time-varying feedback law (12) from the difference between reference and free trajectories. The control law is time-varying because the gain matrix $\boldsymbol{K}(k)$ depends on the dynamic matrix $\boldsymbol{G}(k)$, which is calculated at each sampling instant from the local linearisation of the nonlinear model. It means that matrix inverse must be calculated at each algorithm iteration on-line. For this purpose the LU (Low-Upper) decomposition with partial pivoting of the matrix $\boldsymbol{G}^{\mathrm{T}}(k)\boldsymbol{G}(k) + \boldsymbol{\Lambda}$ can be numerically efficiently used [7].

The explicit nonlinear MPC algorithm described in this paper is designed with reducing the computational complexity in mind. Because at the current sampling instant k only the first element of the vector $\triangle \boldsymbol{u}(k)$ is actually used for control, it is only calculated. Remaining $N_{\mathrm{u}} - 1$ elements are not determined. From (12) one has

$$\triangle u(k|k) = \boldsymbol{K}_1(k)(\boldsymbol{y}^{\mathrm{ref}}(k) - \boldsymbol{y}^{0}(k)) \tag{14}$$

where $\boldsymbol{K}_1(k)$ is the first row of the matrix $\boldsymbol{K}(k)$. In the explicit algorithm a neural network calculates on-line an approximation of the vector $\boldsymbol{K}_1(k)$ for the current operating point. The structure of the algorithm is depicted in Fig. 1. At each sampling instant k of the algorithm the following steps are repeated:

1. Calculate the nonlinear free trajectory $\boldsymbol{y}^{0}(k)$ using the first neural network (NN_1) – a dynamic model of the process.
2. Calculate the approximation of the vector $\boldsymbol{K}_1(k)$ using the second neural network (NN_2).
3. Find the current control increment $\triangle u(k|k)$ from (14).
4. The obtained solution is projected onto the admissible set of constraints.
5. Apply to the process the obtained solution.
6. Set $k := k + 1$, go to step 1.

In consequence, unlike other suboptimal MPC approaches [6,8,16],

- the nonlinear model is not linearised on-line,
- step-response coefficients of the linearised model and the dynamic matrix $\boldsymbol{G}(k)$ are not calculated on-line,
- the inverse matrix $(\boldsymbol{G}^{\mathrm{T}}(k)\boldsymbol{G}(k) + \boldsymbol{\Lambda})^{-1}$ is not calculated on-line.

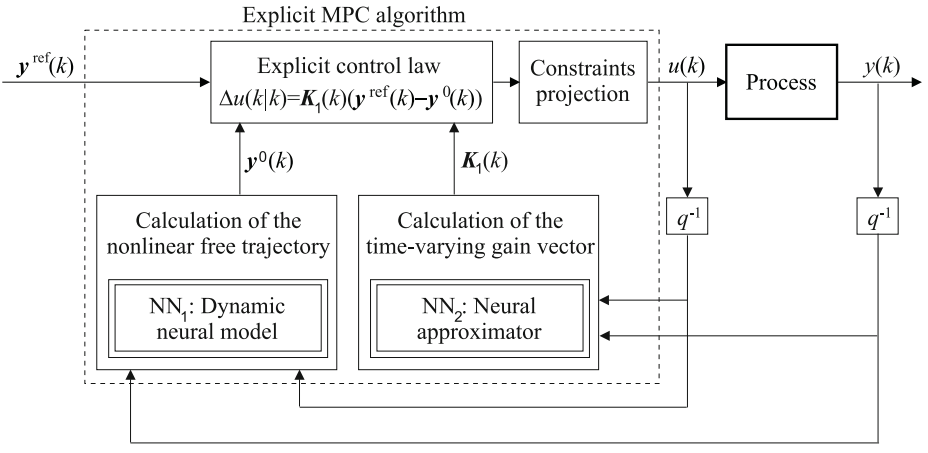


Fig. 1. The structure of the algorithm

Thanks to it, the algorithm can be used for very fast processes or implemented on simple hardware. Two neural models are used on-line: from a dynamic model the free trajectory is found, the second network approximates the vector $\boldsymbol{K}_1(k)$.

In real-life control systems constraints always exist. Usually, they result from actuators' limitations. Another constraints may be necessary to guarantee fulfilment of some technological requirements (e.g. temperature, pressure, purity). Moreover, constraints may be necessary from safety reasons. The following constraints are imposed on magnitude and increment of the manipulated variable

$$u^{\min} \le u(k|k) \le u^{\max}, \qquad -\triangle u^{\max} \le \triangle u(k|k) \le \triangle u^{\max} \tag{15}$$

The calculated current control increment $\triangle u(k|k)$ determined from (14) is projected onto the admissible set of constraints. The projection procedure is

$$\begin{aligned}
&\text{if} \quad \triangle u(k|k) < -\triangle u^{\max} \quad \triangle u(k|k) = -\triangle u^{\max} \\
&\text{if} \quad \triangle u(k|k) > \triangle u^{\max} \quad \triangle u(k|k) = \triangle u^{\max} \\
&u(k|k) = \triangle u(k|k) + u(k-1) \\
&\text{if} \quad u(k|k) < u^{\min} \quad u(k|k) = u^{\min} \\
&\text{if} \quad u(k|k) > u^{\max} \quad u(k|k) = u^{\max} \\
&u(k) = u(k|k)
\end{aligned} \tag{16}$$

3.2 Neural Models and Training

Both neural networks used in the implemented algorithm are MultiLayer Perceptron (MLP) networks with one hidden layer and linear outputs [3], but Radial Basis Functions (RBF) networks can be also used. The first one (NN_1) constitutes a dynamic model of the process, it realises the function f in (3). The nonlinear free trajectory $y^0(k+p|k)$ over the prediction horizon $(p = 1, \ldots, N)$

is calculated on-line recursively using this model. Because the free trajectory describes only the influence of the past, during calculation no changes in the control signal from the sampling instant k onwards are assumed [6,8,16].

For training a sufficiently rich data set must be recorded (e.g. responses to a series of random input steps). When experiments on the real process are not possible, data must be generated from simulations of a fundamental (first-principles) model. Available data set is divided into three sets: training, validation and test sets. Next, neural models with different input arguments and with different number of hidden nodes are trained using the first set. The model is selected using the second set. Finally, the third set is used to assess generalisation abilities of the chosen model. The dynamic neural model can be trained in the one-step ahead prediction configuration (the series-parallel model) or recurrently – in the simulation configuration (the parallel model).

The second network (NN_2) calculates on-line the approximation of the vector $\boldsymbol{K}_1(k) = [k_{1,1}(k) \ldots k_{1,N}(k)]^{\mathrm{T}}$ for the current operating point of the process. A straightforward choice is to define the current operating point by arguments of the dynamic neural model (3) (realised by the first network), i.e. by the vector $\boldsymbol{x}(k) = [u(k-\tau) \ldots u(k-n_{\mathrm{B}})\ y(k-1) \ldots y(k-n_{\mathrm{A}})]^{\mathrm{T}}$. Hence, the second network realises the function $g\colon \mathbb{R}^{n_{\mathrm{A}}+n_{\mathrm{B}}-\tau+1} \rightarrow \mathbb{R}^N$

$$\boldsymbol{K}_1(k) = g(\boldsymbol{x}(k)) = g(u(k-\tau), \ldots, u(k-n_{\mathrm{B}}), y(k-1), \ldots, y(k-n_{\mathrm{A}})) \quad (17)$$

Having obtained the dynamic neural model, a linearisation-based MPC algorithm [6,7,8,16] should be developed. Next, the algorithm is simulated for a randomly changing reference trajectory. As a result data sets for the second network training, verification and testing are recorded. Data sets consists of input and output signals which define the operating point – inputs of the second network. Additionally, time-varying elements of the vector $\boldsymbol{K}_1(k)$ are desired outputs of the model (targets). Unlike the dynamic model, the second network works as an ordinary (steady-state) approximator. Hence, it is not trained recurrently.

Alternatively, data sets necessary for training the second neural network can be generated without the necessity of simulating the MPC algorithm. The dynamic neural model is simulated open-loop (without any controller), as the excitation signal the data set used for training the dynamic neural models is used. During simulations the model is linearised and the vector $\boldsymbol{K}_1(k)$ is calculated.

4 Simulation Results

The process under consideration is a polymerisation reaction taking place in a jacketed continuous stirred tank reactor [10] 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. Polymerisation is a very important chemical process (production of plastic). The reactor

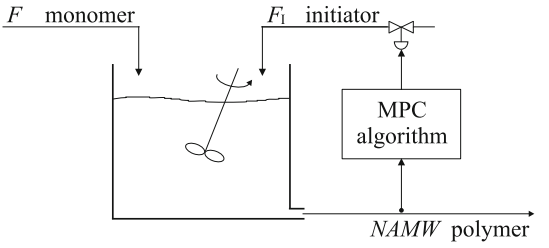


Fig. 2. The polymerisation reactor control system structure

exhibits significantly nonlinear behaviour. It cannot be controlled efficiently by classical MPC schemes based on constant linear models [6,8,10,16,17].

The fundamental model (a set of ordinary differential equations solved using the Runge-Kutta RK45 method) is used as the real process during simulations. It is simulated open-loop in order to obtain training, validation and test data sets. Each set has 2000 samples. The sampling time is 1.8 min. The output signal contains small measurement noise. The second-order dynamic model

$$y(k) = f(u(k-2), y(k-1), y(k-2)) \tag{18}$$

(i.e. $n_{\mathrm{A}} = n_{\mathrm{B}} = \tau = 2$) is chosen. The first MLP neural network has 6 hidden nodes with the hyperbolic tangent transfer function. Because input and output process variables have different orders of magnitude, they are scaled as $u = 100(F_{\mathrm{I}} - F_{\mathrm{I0}})$, $y = 0.0001(NAMW - NAMW_0)$ where $F_{\mathrm{I0}} = 0.028328$, $NAMW_0 = 20000$ correspond to the initial operating point. For training the BFGS (Broyden-Fletcher-Goldfarb-Shanno) optimisation algorithm is used. Fig. 3 depicts the test data set used for assessing accuracy of the dynamic neural model and comparison of the process vs. the model for the first 500 samples. Accuracy of the model is very high. For the training data set SSE $= 5.559159 \cdot 10^{-1}$, for the validation data set SSE $= 1.190907 \cdot 10^{0}$, for the test data set SSE $= 1.039309 \cdot 10^{0}$ (SSE – the Sum of Squared Errors).

Next, a linearisation-based MPC algorithm with Nonlinear Prediction and Linearisation (MPC-NPL) is developed [6,8,16]. For control action calculation it uses a calculated on-line linear approximation of the dynamic neural model and quadratic programming. The algorithm is simulated for a randomly changing reference trajectory ($NAMW^{\mathrm{ref}}$). Data sets for training the second neural network, for verification and for testing are generated. Each data set has 2000 samples. Fig. 4 depicts the first 500 samples (for better presentation) of the test data set used for assessing accuracy of the network.

The prediction horizon is $N = 10$. The second MLP neural network has 8 hidden nodes. The network has 3 inputs (the same as the first network) and 9 outputs, because the first element of the vector $\boldsymbol{K}_1(k)$ is always 0, it is not calculated. For training the BFGS optimisation algorithm is used. For the training data set SSE $= 1.006511 \cdot 10^{-2}$, for the validation data set SSE $= 1.311316 \cdot 10^{-2}$, for the test data set SSE $= 1.097440 \cdot 10^{-2}$.

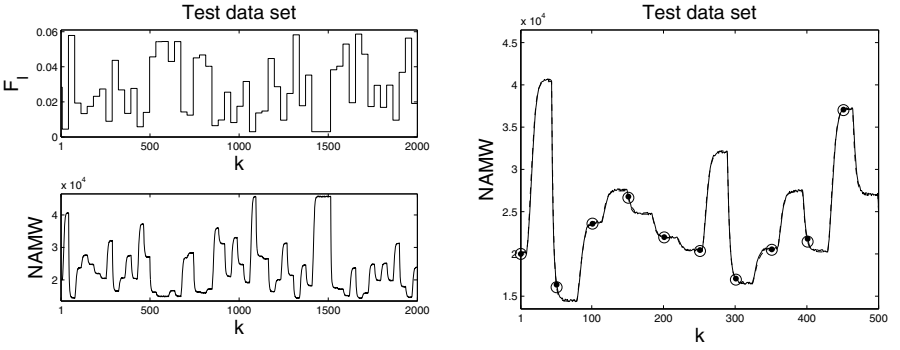


Fig. 3. *Left*: The whole test data set used for assessing accuracy of the dynamic neural model (NN_1); *right*: the process (*solid line with dots*) vs. the neural model (*dashed line with circles*) for the first 500 samples of the test data set

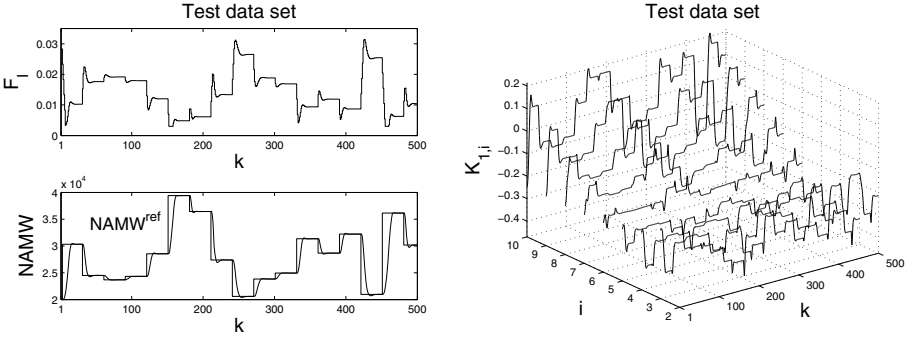


Fig. 4. The first 500 samples of the test data set used for assessing accuracy of the neural approximator (NN_2) of the vector $\boldsymbol{K}_1(k)$

In order to demonstrate accuracy and computational efficiency of the described explicit approach, the following MPC algorithms are compared:

a) the explicit MPC algorithm,
b) the MPC-NPL algorithm with on-line model linearisation and quadratic programming [8,6,16,17],
c) the MPC-NO algorithm with on-line nonlinear optimisation [8,16,17].

All three algorithms use the same dynamic neural model (NN_1), the explicit MPC algorithm also needs the second neural network (NN_2).

Parameters of all MPC algorithms are $N = 10$, $N_\mathrm{u} = 3$, $\lambda_p = 0.2$. The manipulated variable is constrained: $F_\mathrm{I}^{\min} = 0.003$, $F_\mathrm{I}^{\max} = 0.06$, $\triangle F_\mathrm{I}^{\max} = 0.005$. Fig. 5 shows trajectories obtained in the MPC-NO algorithm and in the explicit algorithm (results of the MPC-NPL algorithm are similar, they are not depicted). Table 1 shows accuracy of algorithms in terms of the SSE index.

[t]

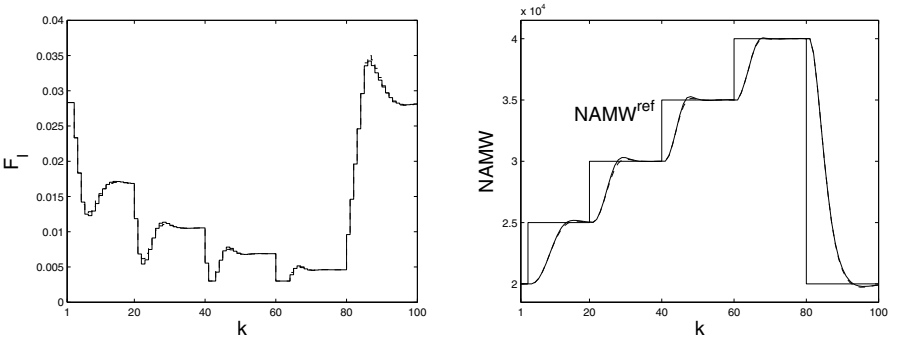


Fig. 5. Simulation results: the MPC-NO algorithm with nonlinear optimisation (*solid line*) and the explicit neural MPC algorithm (*dashed line*)

Table 1. Accuracy of compared algorithms, $N = 10$, $N_u = 3$

Algorithm	SSE
Explicit MPC	$2.211651 \cdot 10^9$
MPC-NPL with quadratic programming	$2.211703 \cdot 10^9$
MPC-NO with nonlinear optimisation	$2.210627 \cdot 10^9$

Table 2. The computational burden (in MFLOPS) of compared algorithms for different control horizons ($N_u = 2, 3, 5, 10$), $N = 10$

Algorithm	$N_u = 2$	$N_u = 3$	$N_u = 5$	$N_u = 10$
Explicit MPC	0.19	0.19	0.19	0.19
MPC-NPL with quadratic programming	0.28	0.40	0.81	3.20
MPC-NO with nonlinear optimisation	2.64	4.11	8.90	48.36

Accuracy of the linearisation-based MPC-NPL algorithm is practically the same as that of the computationally demanding MPC-NO approach. Thanks to using a neural approximator of the time-varying vector $\boldsymbol{K}_1(k)$, the explicit algorithm gives similar results, but model linearisation, calculation of the step-response and quadratic programming are not performed on-line. Hence, it is significantly less complicated than the MPC-NPL approach. Table 2 shows the computational burden (MFLOPS) of algorithms for different control horizons. The longer the horizon, the more evident computational efficiency of the explicit algorithm (it is independent of the control horizon because $\boldsymbol{K}_1(k) \in \mathbb{R}^N$).

5 Conclusions

Computational efficiency of the presented explicit MPC algorithm is twofold: quantitative and qualitative. Its computational burden is low. Furthermore, it

does not need any on-line optimisation and model linearisation. It is also not necessary to carry out a matrix decomposition task and solve a set of linear equations as it is necessary in existing explicit approaches [7]. Hence, the algorithm can be used for fast processes (short sampling time). It is not necessary to use sophisticated (and expensive) hardware necessary to implement the algorithm.

Acknowledgement. The work presented in this paper was supported by Polish national budget funds for science for years 2009-2011.

References

1. Åkesson, B.M., Toivonen, H.T.: A neural network model predictive controller. Journal of Process Control 16, 937–946 (2006)
2. Cavagnari, L., Magni, L., Scattolini, R.: Neural network implementation of nonlinear receding-horizon control. Neural Computing and Applications 8, 86–92 (1999)
3. Haykin, S.: Neural networks – a comprehensive foundation. Prentice Hall, Englewood Cliffs (1999)
4. Henson, M.A.: Nonlinear model predictive control: Current status and future directions. Computers and Chemical Engineering 23, 187–202 (1998)
5. Johansen, T.A.: Approximate explicit receding horizon control of constrained nonlinear systems. Automatica 40, 293–300 (2004)
6. Ławryńczuk, M.: Neural networks in model predictive control. In: Nguyen, N.T., Szczerbicki, E. (eds.) Intelligent Systems for Knowledge Management. Studies in Computational Intelligence, vol. 252, pp. 31–63. Springer, Heidelberg (2009)
7. Ławryńczuk, M.: Explicit nonlinear predictive control of a distillation column based on neural models. Chemical Engineering and Technology 32, 1578–1587 (2009)
8. Ławryńczuk, M.: A family of model predictive control algorithms with artificial neural networks. International Journal of Applied Mathematics and Computer Science 17, 217–232 (2007)
9. Maciejowski, J.M.: Predictive control with constraints. Prentice Hall, Harlow (2002)
10. 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)
11. Morari, M., Lee, J.H.: Model predictive control: past, present and future. Computers and Chemical Engineering 23, 667–682 (1999)
12. Nørgaard, M., Ravn, O., Poulsen, N.K., Hansen, L.K.: Neural networks for modelling and control of dynamic systems. Springer, London (2000)
13. Pan, Y., Wang, J.: Nonlinear model predictive control using a recurrent neural network. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2008), Hong Kong, China, pp. 2296–2301 (2008)
14. Parisini, T., Sanguineti, M., Zoppoli, R.: Nonlinear stabilization by receding-horizon neural regulators. International Journal of Control 70, 341–362 (1998)
15. Qin, S.J., Badgwell, T.A.: A survey of industrial model predictive control technology. Control Engineering Practice 11, 733–764 (2003)
16. Tatjewski, P.: Advanced control of industrial processes, Structures and algorithms. Springer, London (2007)
17. Tatjewski, P., Ławryńczuk, M.: Soft computing in model-based predictive control. International Journal of Applied Mathematics and Computer Science 16, 101–120 (2006)

Solution of the Inverse Heat Conduction Problem by Using the ABC Algorithm

Edyta Hetmaniok, Damian Słota, and Adam Zielonka

Institute of Mathematics,
Silesian University of Technology,
Kaszubska 23, 44-100 Gliwice, Poland
{edyta.hetmaniok,damian.slota,adam.zielonka}@polsl.pl

Abstract. In this paper, a numerical method of solving the inverse heat conduction problem based on the respectively new tool for combinational optimization, named the Artificial Bee Colony algorithm (ABC), is presented. In the first step, the direct heat conduction problem, associated to the considered inverse heat conduction problem, is solved by using the finite difference method. In the second step, the proper functional, based on the least squares method, is minimized by using the ABC algorithm, giving the solution of the considered problem. An example illustrating the precision and effectiveness of the method is also shown. The proposed approach is original and promising.

Keywords: Swarm Intelligence, Artificial Bee Colony algorithm, Inverse Heat Conduction Problem, Finite Difference Method.

1 Introduction

Many problems in technology, engineering, economy and natural sciences bring into solving optimization tasks, consisting in minimizing some functionals. There is many tools for solving such problems, but in the last few years there appear a new instrument for combinatorial optimization, called as the Artificial Bee Colony Algorithm (ABC). ABC algorithm, next to the Ant Colony Optimization algorithm (ACO), is a part of Swarm Intelligence (SI), which is a group of algorithms of artificial intelligence, based on the collective behaviour of decentralized, self-organized systems of objects. The idea was introduced by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems [1] and continued for example in [2].

Most of the present optimization algorithms need to fulfill a different number of assumptions about the properties of optimized function, its variables or its domain. It causes, that the classical algorithms (like for example the finite element method or the finite difference method) can be used only for solving a special kind of optimized problem. Much more universal, about the kinds of solved problems, are the algorithms motivated by the nature, like the genetic algorithms or algorithms inspired by the behaviour of the insects, like ABC or

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 659–668, 2010.

ACO algorithms. The only assumption needed by those algorithms is the existence of the solution. If the solution of the optimized problem exists, it will be found, with some given precision of course. It is worth to mention in this moment, that solution received by using those algorithms should be treated as the best solution in the given moment. Running the algorithm one more time can give different solution, even better. But it does not decrease the effectiveness of those algorithms.

The Artificial Bee Colony algorithm was inspired by the technique of searching for the nectar around the hive by the colony of bees. The first researcher, who by observing the behaviour of bees, described how the bees, after discovering the source of food far from the hive, can inform the other bees in the hive about the position of the food, was Karl von Frisch from the University of Munich. It is a very complicated strategy of communication, unique in the nature.

When the bee, called the scout, has localized a good source of food, it collects a sample of the nectar and flies back to the hive for informing the other bees about the available source of food. Soon after the scout returned to the hive, a lot of bees leave the hive and fly in the direction of the discovered source of nectar. The scouts stay in the hive for some time and inform the other bees about the position of the food with the aid of the special waggle dance. After that they leave the hive for searching a new source of food.

The waggle dance takes place in the special part of the hive near the exit and it consists of the moving straight and returning to the starting point. When the bee is moving straight, its body waggles and its wings vibrate very quickly. The direction of the waggle dance, distance of the moving straight and deviation of the bee's body during the vibration inform about the location, distance and the quality of the source of food. More detailed information about the natural inspiration of the ABC algorithm can be found in [3,4].

Till now, the ABC algorithm has been applied for solving a different kind of combinatorial and analytical problems, like for example the transportation problem, reaction-diffusion problem, generalized assignment problem and others [5,6,7,8]. In this paper we present the idea of using the ABC algorithm for solving the inverse heat conduction problem, which means the heat conduction problem without the complete mathematical description, consisting in the reconstruction of the state function and some of the boundary conditions [9,10].

The bibliography sacrificed to the inverse heat conduction problem is much more poor than the bibliography about the direct problems. Examples of the analytical techniques for solving the direct and inverse problems concerning steady and unsteady heat flow can be found in [9,10]. In [11] the authors determine the heat flux with the aid of the momentary measurements of temperature, by using the Green function, method of iterative regularization and Tichonov regularization. The other methods appeared for solving the inverse problems are for example: the Monte Carlo method [12], the mollification method introduced by Mourio and his co-workers [13], methods based on the vawelets theory [14] and very popular in recent time genetic algorithms and neural network [15]. In the current paper, we propose to use the ABC algorithm for minimizing some

functional, being a crucial part of the presented method of solving the inverse heat conduction problem. In [16] the authors have already applied for this purpose the ACO algorithm.

2 Artificial Bee Colony Algorithm for Finding the Global Minimum

In the proposed approach we use the following simplifications:

- We divide the bee colony into two parts: the bees-scouts, exploring the environment and the bees-viewers, waiting in the hive for the information. The numbers of scouts and viewers are equal.
- All of the bees-scouts, after the exploration of the discovered sources of nectar, come back to the hive, give the information to the bees-viewers and wait there for the next cycle. In the next cycle, they start the exploration from the positions of sources discovered in the previous cycle.
- All of the bees start the exploration in the same time. According to the scientific research, the real number of new bees, starting the exploration, is proportional to the difference between the total number of bees and the number of actually searching bees.

Let us consider the function $F(\mathbf{x})$, defined in the domain D. We do not need to make any assumptions about the function, neither its domain. Points of the domain - vectors $\mathbf{x}$ - play the role of the sources of nectar. Value of the function in the given point - number $F(\mathbf{x})$ - designates the quality of the source $\mathbf{x}$. Since we are looking for the minimum, the smaller is the value $F(\mathbf{x})$, the better is the source $\mathbf{x}$.

In the first part of the algorithm, the bees-scouts explore the domain and type some number of the points - candidates for the sources of nectar. Every scout make some control movements around the selected point, to check whether there is any better source in the neighborhood. After that, the scouts return to the hive and wait there for the next cycle.

In the second part of the algorithm, the bees-viewers select the sources, with the given probabilities, among the sources discovered by the scouts in the first part. The probability of the choice of the given source is the greater, the better is the quality of that source. After that, the viewers explore the selected points, by making some control movements around. The operation ends by choosing the best point - the best source of nectar - in the current cycle.

We will proceed according the following algorithm.

Initialization of the algorithm.

1. Initial data:
   SN - number of the explored sources of nectar (= number of the bees - scouts, = number of the bees - viewers);
   D - dimension of the source $\mathbf{x}_i$, $i = 1, ..., SN$;
   lim - number of "corrections" of the source position $\mathbf{x}_i$;
   MCN - maximal number of cycles.

2. Initial population - random selection of the initial sources localization, represented by the $D-$ dimensional vectors $\mathbf{x}_i$, $i = 1, ..., SN$.
3. Calculation of the values $F(\mathbf{x}_i)$, $i = 1, ..., SN$, for the initial population.

The main algorithm.

1. Modification of the sources localizations by the bees - scouts.
   a) Every bee - scout modifies the position $\mathbf{x}_i$ according to the formula:
   $$v_i^j = x_i^j + \phi_{ij}(x_i^j - x_k^j), \qquad j \in \{1, ..., D\},$$
   where: $\left.\begin{array}{l} k \in \{1, ..., SN\}, k \neq i, \\ \phi_{ij} \in [-1, 1]. \end{array}\right\}$ - randomly selected numbers.
   b) If $F(\mathbf{v}_i) \leq F(\mathbf{x}_i)$, then the position $\mathbf{v}_i$ replaces $\mathbf{x}_i$. Otherwise, the position $\mathbf{x}_i$ stays unchanged.
   Steps a) and b) are repeated lim times. We take: $lim = SN \cdot D$.
2. Calculation of the probabilities P_i for the positions $\mathbf{x}_i$ selected in step 1. We use the formula:
   $$P_i = \frac{fit_i}{\sum\limits_{j=1}^{SN} fit_j}, \quad i = 1, ..., SN,$$
   where: $fit_i = \begin{cases} \frac{1}{1+F(\mathbf{x}_i)} & \text{if } F(\mathbf{x}_i) \geq 0, \\ 1 + |F(\mathbf{x}_i)| & \text{if } F(\mathbf{x}_i) < 0. \end{cases}$
3. Every bee - viewer chooses one of the sources $\mathbf{x}_i$, $i = 1, ..., SN$, with the probability P_i. Of course, one source can be chosen by a group of bees.
4. Every bee - viewer explores the chosen source and modifies its position according to the procedure described in step 1.
5. Selection of the $\mathbf{x}_{best}$ for the current cycle - the best source among the sources determined by the bees - viewers. If the current $\mathbf{x}_{best}$ is better that the one from the previous cycle, we accept it as the $\mathbf{x}_{best}$ for the whole algorithm.
6. If in step 1, the bee - scout did not improve the position $\mathbf{x}_i$ ($\mathbf{x}_i$ did not change), it leaves the source $\mathbf{x}_i$ and moves to the new one, according to the formula:
   $$x_i^j = x_{min}^j + \phi_{ij}(x_{max}^j - x_{min}^j), \qquad j = 1, ..., D,$$
   where: $\phi_{ij} \in [0, 1]$.

Steps 1-6 are repeated MCN times.

3 Inverse Heat Conduction Problem

3.1 Formulation of the Problem

We consider the Fourier heat equation of the form:

$$\frac{1}{a}\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t), \qquad x \in [0,1], \;\; t \in [0,T], \tag{1}$$

where a is the thermal diffusivity, and u, t and x refer to the temperature, time and spatial location, with the following boundary condition of the first kind:

$$u(0,t) = \psi(t), \qquad t \in [0,T] \tag{2}$$

and the initial condition:

$$u(x,0) = \varphi(x), \qquad x \in [0,1]. \tag{3}$$

Symbols ψ and φ denotes the functions belonging to the proper class of functions.

We also know the numbers u_j^ε, which are the values of the temperature, measured at one point x_0, in m different moments of time τ_j, $j = 1, ..., m$. Since the values $u_j^\varepsilon = u(x_0, \tau_j)$ denote some results of measurement, they contain errors. We assume, that the amplitude of noise is bounded by ε.

The unknown elements in such determined problem are the distribution of temperature $u(x,t)$ and the form of the boundary condition for the boundary $x = 1$. We assume, that in our case the sought boundary condition is of the second kind (the heat flux) and is described by the function $q(t)$:

$$\frac{\partial u}{\partial x}(1,t) = q(t), \qquad t \in [0,T]. \tag{4}$$

3.2 Method of Solution

Since the function $q(t)$, describing the boundary condition (4), is unknown, we assume its form as the linear combination of some given base functions $\nu_i(t)$:

$$q(t) \approx \tilde{q}(t) = \sum_{i=0}^{k} b_i \nu_i(t), \tag{5}$$

where b_i, $i = 0, 1, ..., k$, are some undetermined coefficients.

First part of the proposed method consists in solving the direct heat conduction problem, described by the equations (1)-(5), by using one of the well known numerical methods. The received solution will depend on the unknown coefficients b_i, $i = 0, 1, ..., k$.

For solving the direct heat conduction problem we propose the implicit scheme of the finite difference method, because it is always stable and convergent. According to this method, we discretize the problem by using the partition of the domain $[0,1] \times [0,T]$ with a mesh Δ, of evenly placed points (x_i, t_j) with constant step h_x in space and constant step h_t in time:

$$\begin{aligned} \Delta = \big\{(x_i,t_j): \;\; & x_i = ih_x, \; h_x = \frac{1}{n}, \; i = 1, ..., n, \\ & t_j = jh_t, \; h_t = \frac{T}{m}, \; j = 0, ..., m\big\}. \end{aligned} \tag{6}$$

The points $\tilde{u}_i^j = \tilde{u}(x_i, t_j)$, $i = 1, ..., n$, $j = 0, ..., m$, represent the numerical approximation of the sought values $u(x_i, t_j)$ and they should satisfy the discretized heat equation of the form:

$$\frac{1}{a}\frac{\tilde{u}_i^{j+1}-\tilde{u}_i^j}{h_t}=\frac{\tilde{u}_{i+1}^{j+1}-2\tilde{u}_i^{j+1}+\tilde{u}_{i-1}^{j+1}}{h_x^2}. \tag{7}$$

Equation (7), together with the given initial and boundary conditions, leads to the system of linear equations. Solution of this system gives the set of points $\tilde{u}_i^j$, approximating the values of the requested function $u(x,t)$ and depending on the unknown coefficients b_i, $i=0,1,...,k$.

Second part of the method rests on determining the coefficients b_i. For this purpose we define the following functional:

$$P(b_0,b_1,...,b_k)=\sqrt{\sum_{j=1}^{m}\left(u_j^{\varepsilon}-\tilde{u}(x_0,\tau_j)\right)^2}, \tag{8}$$

where u_j^{ε} are the measurement values and $\tilde{u}(x_0,\tau_j)$ are the results received in the first part of the method, as the solution of the direct heat conduction problem, with some given values of coefficients $b_0,b_1,...,b_k$ in (5), for the nodes (x_0,τ_j), $j=1,...,m$.

We want to determine such values of coefficients b_i, for which the functional (8) is minimal, which means, the reproduced state function $\tilde{u}$ and boundary condition function $\tilde{q}$ are the best adapted to the real data. For minimizing the functional (8) we use the Artificial Bee Colony algorithm, introduced in section 2. It is important to point, that to calculate the value of the minimized functional means to solve the direct heat conduction problem for the given coefficients b_i (in the boundary condition function $\tilde{q}$) and to evaluate the value of functional (8).

4 Experimental Results and Discussion

The theoretical consideration, presented in the previous sections, will be now illustrated with an example, in which $a=1$, $T=1$ and the functions describing the boundary and initial conditions (2)-(3) are the following:

$$\psi(t)=\exp(t), \qquad t\in[0,1],$$
$$\varphi(x)=\exp(x), \qquad x\in[0,1].$$

We know the noised values u_j^{ε} of the temperature, measured with maximal noise of 0%, 1%, 2% or 5%, at one point $x_0=0.7$, for 100 different moments of time $\tau_j=j/100$, $j=1,...,100$.

The direct heat conduction problem, occurring from equations (1)-(4) for a given heat flux, is solved via the finite difference method. As a result, the temperature distribution in the domain is obtained, constituting the reference point u_j^{ε} for a comparison of results. From the distribution, temperatures u_j, simulating the temperature measurements, are obtained.

Following the procedure, described in the section 3.2, first we need to assume some form of unknown function (5), representing the boundary condition for the

boundary $x = 1$. Let us assume, that we will search the function $\tilde{q}$ as the linear combination of the base functions 1, $\exp(t)$ and $\exp(2t)$:

$$\tilde{q}(t) = b_0 + b_1 \exp(t) + b_2 \exp(2t), \tag{9}$$

where b_0, b_1 and b_2 are the undetermined coefficients. Under this assumption we solve the direct heat conduction problem, by using the implicit scheme of the finite difference method.

Since the measurement values are given in moments $\tau_j = j/100$, $j = 1, ..., 100$, it will be convenient to discretize the domain with the step exactly equal to $1/100$ in time and in space. So we introduce the following mesh:

$$\Delta = \big\{(x_i, t_j) : \quad x_i = \frac{i}{100}, \quad i = 1, ..., 100, \quad t_j = \frac{j}{100}, \quad j = 0, ..., 100\big\}.$$

The point x_0, in which the measurement values are given, is placed now in the node $x_{70} = 70/100$.

Every solution of the direct heat conduction problem, under assumption (9), leads to the functional (8), taking now the form:

$$P(b_0, b_1, b_2) = \sqrt{\sum_{j=1}^{100} \left(u_j^{\varepsilon} - \tilde{u}\Big(\frac{70}{100}, \frac{j}{100}\Big)\right)^2}, \tag{10}$$

with the unknown parameters b_0, b_1 and b_2 of the linear combination (9) as the variables. The above functional is minimized with the aid of the ABC algorithm.

The values of algorithm initial data are as below:
$SN = 25$ - number of bees (= explored sources of nectar - it means vectors (b_0, b_1, b_2));
$D = 3$ - dimension of the source;
$lim = SN \cdot D = 75$ - number of "corrections" of the source position;
$MCN = 200$ - maximal number of cycles.

After 30 runnings of the algorithm we received the following mean values of the sought parameters, received for the maximal noise of the input data $\varepsilon = 2\%$:

$$b_0 = -0.04215642, \quad b_1 = 2.8325213, \quad b_2 = -0.0592602,$$

with the values of standard deviation equals to, respectively:

$$S_0 = 0.0653689, \quad S_1 = 0.0813461, \quad S_2 = 0.0239036.$$

Thus, the requested function (9), describing the boundary condition of the second kind for the boundary $x = 1$, has the form:

$$\tilde{q}(t) = -0.04215642 + 2.8325213 \exp(t) - 0.0592602 \exp(2t).$$

The exact solution of the considered problem, with unnoised data, gives the function: $u(x, t) = \exp(x + t)$, which means, that the unknown function (4) describing the boundary condition is of the form: $q(t) = \exp(1 + t)$.

In Figure 1 the reconstructed boundary condition $\tilde{q}(t)$ is compared with the exact condition $q(t)$. The relative error distribution of this reconstruction is also displayed in this figure. The received results show, that function describing the heat flux is reconstructed very well at the beginning of the considered period of time. The reconstruction error slightly grows with the passing of time, which can be explained with the fact, that the additional initial condition is given at the initial moment of time.

Figure 2 shows the comparison between the values of the exact function, describing the distribution of temperature for the moment $t = 1$ ($u(x, 1)$) and its received approximated values, with the relative error distribution of the obtained approximation. From the figure we see, that the distribution of temperature at the end of the considered period of time is reconstructed with the error, which is in the worst case two times smaller (about 1%) than the error of the input data (2%). The reconstruction is better at the beginning of the considered region, because the boundary condition is known for the boundary $x = 0$.

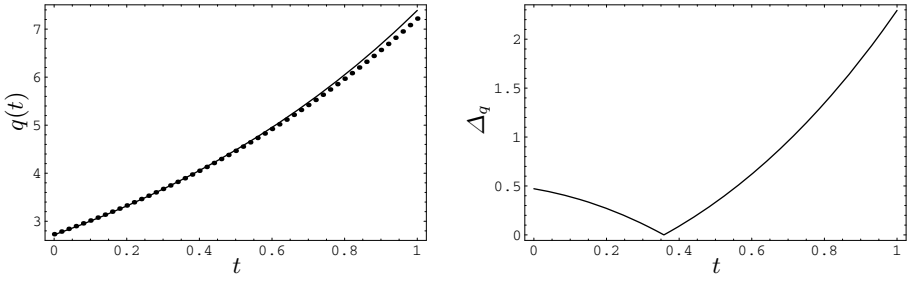


Fig. 1. Boundary condition of the second kind for the boundary $x = 1$ (left figure: solid line – exact condition, dashed line – reconstructed condition) and error distribution of this reconstruction (right figure)

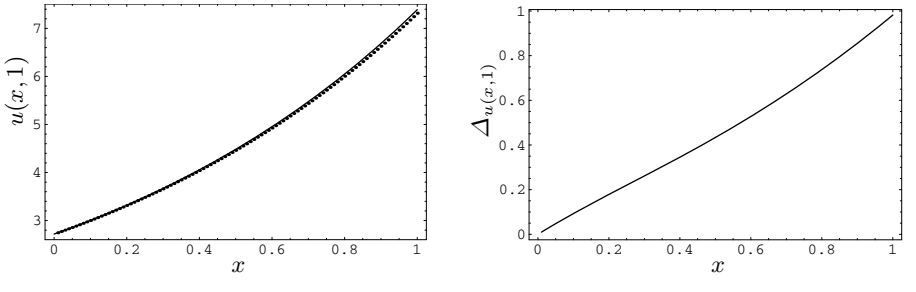


Fig. 2. Distribution of the temperature $u(x, t)$ for $t = 1$ (left figure: solid line – exact solution, dashed line – approximated values) and error of this approximation (right figure)

In Table 1 there are compiled the errors, received for different values of the maximal noise of input data: 0%, 1%, 2% and 5%. We consider the absolute and

relative errors of the reconstructed boundary condition $\tilde{q}(t)$, calculated by using the following formulas:

$$\delta_q = \sqrt{\int_0^1 \big(q(t) - \tilde{q}(t)\big)^2 dt}, \qquad \Delta_q = \frac{\delta_q}{\sqrt{\int_0^1 q^2(t)dt}} 100\%,$$

and the absolute and relative errors of the approximated values of the temperature $\tilde{u}_i^j$ in all considered domain, defined by the formulas:

$$\delta_u = \sqrt{\sum_{i=1}^{100}\sum_{j=0}^{100}\big(u(x_i,t_j) - \tilde{u}_i^j\big)^2}, \qquad \Delta_u = \frac{\delta_u}{\sqrt{\sum\limits_{i=1}^{100}\sum\limits_{j=0}^{100} u(x_i,t_j)^2}} 100\%.$$

Table 1. Values of the error in reconstruction of the boundary condition $q(t)$ and the temperature distribution $u(x,t)$

	$\varepsilon = 0\%$	$\varepsilon = 1\%$	$\varepsilon = 2\%$	$\varepsilon = 5\%$
δ_q	0.010614	0.052549	0.061622	0.071905
$\Delta_q[\%]$	0.218471	1.081600	1.268350	1.480000
δ_u	0.000848	0.006597	0.010449	0.010203
$\Delta_u[\%]$	0.026537	0.206517	0.327056	0.322029

The table shows, that the errors are insignificantly getting bigger, if the assumed input data noise is bigger, but they are always comparable with the maximal noise of the input data. One can also notice, that the growth of the result errors is slower than the growth of the input data error.

5 Conclusions

In this paper, a new and efficient method for solving the inverse heat conduction problem, based on the idea of Artificial Bee Colony algorithm, is proposed. Presented example shows, that the solution received with the aid of this method is noised with the error comparable with the error of the input data. This is especially important while considering the fact, that the control point, with the known measurement values of the unknown function $u(x,t)$, was located in $1/3$ distance of the boundary, where the boundary condition was reconstructed. Besides, the error of the boundary condition reconstruction and approximation of the temperature distribution values, received by using this method, grows slower, than the noise of the input data. The additional advantages of the approach based on the ABC algorithm are also the simplicity of implementation and respectively short time of working.

We should also emphasize the fact, that in the proposed approach one needs to use some method of solving the direct heat conduction problem, not necessarily the finite difference method, used in this paper. The future work includes an application of the proposed approach for the wider class of problems and by using some alternative methods for solving the direct problems. The comparison between the ABC algorithm, the Ant Colony Optimization, genetic and immune algorithms, used in considered approach, is also planned for the future work.

References

1. Beni, G., Wang, J.: Swarm intelligence in cellular robotic systems. In: Proceed. NATO Advanced Workshop on Robots and Biological Syst., Tuscany (1989)
2. Eberhart, R.C., Shi, Y., Kennedy, J.: Swarm Intelligence. Morgan Kaufmann, San Francisco (2001)
3. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing 8, 687–697 (2007)
4. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation 214, 108–132 (2009)
5. Teodorovič, D.: Transport modelling by multi-agent systems: a swarm intelligence approach. Transportation Planning and Technology 26, 289–312 (2003)
6. Tereshko, V.: Reaction-diffusion model of a honeybee colony's foraging behaviour. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 807–816. Springer, Heidelberg (2000)
7. Özbakir, L., Baykasoğlu, A., Tapkan, P.: Bees algorithm for generalized assignment problem. Applied Mathematics and Computation 215, 3782–3795 (2010)
8. Pham, D.T., Castellani, M.: The bees algorithm: modelling foraging behaviour to solve continuous optimization problems. In: Proc. IMechE 223 Part C, pp. 2919–2938 (2009)
9. Beck, J.V., Blackwell, B., St-Clair, C.R.: Inverse Heat Conduction: Ill Posed Problems. Wiley Intersc., New York (1985)
10. Beck, J.V., Blackwell, B.: Inverse Problems. In: Handbook of Numerical Heat Transfer. Wiley Intersc., New York (1988)
11. Beck, J.V., Cole, K.D., Haji-Sheikh, A., Litkouhi, B.: Heat Conduction Using Green's Functions. Hempisphere. Publishing Corporation, Philadelphia (1992)
12. Haji-Sheikh, A., Buckingham, F.P.: Multidimensional inverse heat conduction using the Monte Carlo method. Trans. of the ASME, Journal of Heat Transfer 115, 26–33 (1993)
13. Mourio, D.A.: The Mollification Method and the Numerical Solution of Ill-posed Problems. John Wiley and Sons Inc., New York (1993)
14. Qiu, C.Y., Fu, C.L., Zhu, Y.B.: Wavelets and relularization of the sideways heat equation. Computers and Mathematics with Applications 46, 821–829 (2003)
15. Słota, D.: Solving the inverse Stefan design problem using genetic algorithm. Inverse Probl. Sci. Eng. 16, 829–846 (2008)
16. Hetmaniok, E., Zielonka, A.: Solving the inverse heat conduction problem by using the ant colony optimization algorithm. In: CMM 2009. University of Zielona Góra Press, pp. 205–206 (2009)

Application of Fuzzy Wiener Models in Efficient MPC Algorithms

Piotr M. Marusak

Institute of Control and Computation Engineering, Warsaw University of Technology, ul. Nowowiejska 15/19, 00–665 Warszawa, Poland
P.Marusak@ia.pw.edu.pl

Abstract. Efficient Model Predictive Control (MPC) algorithms based on fuzzy Wiener models are proposed in the paper. Thanks to the form of the model the prediction of the control plant output can be easily obtained. It is done in such a way that the MPC algorithm is formulated as a numerically efficient quadratic optimization problem. Moreover, inversion of the static process model, used in other approaches, is avoided. Despite its relative simplicity the algorithm offers practically the same performance as the MPC algorithm in which control signals are generated after solving a nonlinear optimization problem and outperforms the MPC algorithm based on a linear model. The efficacy of the proposed approach is demonstrated in the control system of a nonlinear control plant.

Keywords: fuzzy systems, fuzzy control, predictive control, nonlinear control, constrained control.

1 Introduction

Model predictive control (MPC) algorithms are widely used in practice. It is because they offer very good control performance even for control plants which are difficult to control using other algorithms [4,9,15,18]. The essential feature of these algorithms is to use a control plant model to predict behavior of the control system. Thanks to such an approach, the MPC algorithms are formulated in such a way that constraints existing in the control system can be relatively easily taken into consideration. Moreover, it is possible to use all information about control system operation and on conditions in which it operates to improve prediction and, as a result, operation of an MPC algorithm.

In standard MPC algorithms linear control plant models are used for prediction. Then an algorithm can be formulated as an easy to solve, quadratic optimization problem. Moreover, in the unconstrained case, a control law can be easily obtained. Unfortunately, application of such an MPC algorithm to a nonlinear plant may bring unsatisfactory results or the results can be improved using the algorithm based on a nonlinear model. This problem is especially important if the control system should work in a wide range of set point values.

Direct application of a nonlinear process model to design the MPC algorithm does not solve all issues. It is because it leads to formulation of the algorithm as

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 669–677, 2010.

a nonlinear, and in general, non–convex optimization problem. Such a problem is hard to solve and computationally expensive. The approach which does not have these drawbacks consists in obtaining a linear approximation of the nonlinear model at each iteration of the algorithm. It can be done in an efficient way if a control plant is described using a Wiener model.

The Wiener models are composed of a linear dynamic block preceding a nonlinear static block (Fig. 1) [7]. Such a structure of the model simplifies the synthesis of the controllers based on Wiener models. Therefore, Wiener models are often used to model control plants for control purposes; see e.g. [2,10,16].

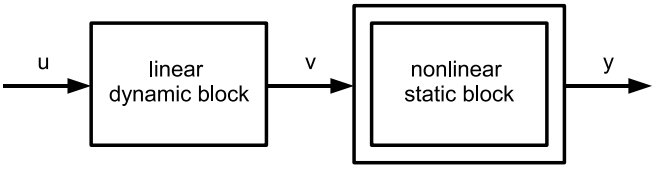


Fig. 1. Structure of the Wiener model; u – input, y – output, v – input of the nonlinear static block

The most popular method of application of Wiener models in the MPC algorithms, in such a way that computationally efficient quadratic optimization problem is solved at each iteration, is to use inverse of the static part of the model; see e.g. [1,16]. On the contrary, in the method proposed in the paper the calculation of the inverse of the static part of the model is avoided. Moreover, the prediction can be performed in a straightforward way what does not influence control performance in a negative way. It is demonstrated in the example control system of a nonlinear plant fuzzy model of which was obtained heuristically.

In the next section the idea of the MPC algorithms is described. Next, the MPC algorithms based on fuzzy Wiener models are proposed. Sect. 4 contains presentation of results obtained in the control system of the nonlinear plant, illustrating excellent performance offered by the proposed approach. The paper is summarized in the last section.

2 MPC Algorithms – Basic Information

The Model Predictive Control (MPC), during control signal generation, predict future behavior of the control plant many sampling instants ahead using a process model. The control signal is derived in such a way that the prediction fulfills assumed criteria. These criteria are, usually, formulated as the following optimization problem [4,9,15,18]:

$$\min_{\Delta \boldsymbol{u}} \left\{ J_{\mathrm{MPC}} = \sum_{i=1}^{p} \left(\overline{y}_k - y_{k+i|k} \right)^2 + \sum_{i=0}^{s-1} \lambda \left(\Delta u_{k+i|k} \right)^2 \right\} \tag{1}$$

subject to:

$$\Delta \boldsymbol{u}_{\min} \leq \Delta \boldsymbol{u} \leq \Delta \boldsymbol{u}_{\max} \ , \tag{2}$$

$$\boldsymbol{u}_{\min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\max} \ , \tag{3}$$

$$\boldsymbol{y}_{\min} \leq \boldsymbol{y} \leq \boldsymbol{y}_{\max} \ , \tag{4}$$

where $\overline{y}_k$ is a set–point value, $y_{k+i|k}$ is a value of the output for the $(k+i)^{\text{th}}$ sampling instant, predicted at the k^{th} sampling instant, $\Delta u_{k+i|k}$ are future changes of the control signal, $\lambda \geq 0$ is a tuning parameter, p and s denote prediction and control horizons, respectively; $\Delta \boldsymbol{u} = \left[\Delta u_{k+1|k}, \ldots, \Delta u_{k+s-1|k}\right]$, $\boldsymbol{u} = \left[u_{k+1|k}, \ldots, u_{k+s-1|k}\right]$, $\boldsymbol{y} = \left[y_{k+1|k}, \ldots, y_{k+p|k}\right]$; $\Delta \boldsymbol{u}_{\min}$, $\Delta \boldsymbol{u}_{\max}$, $\boldsymbol{u}_{\min}$, $\boldsymbol{u}_{\max}$, $\boldsymbol{y}_{\min}$, $\boldsymbol{y}_{\max}$ are vectors of lower and upper limits of changes and values of the control signal and of the values of the output signal, respectively. The optimization problem (1–4) is solved at each iteration of the algorithm. Its solution is the optimal vector of changes of the control signal. From this vector, the first element is applied to the control plant and then the optimization problem is solved again in the next iteration of the MPC algorithm.

The predicted output variables $y_{k+j|k}$ are derived using a dynamic control plant model. If this model is nonlinear then the optimization problem (1–4) is nonlinear and, in general, non–convex and hard to solve. Examples of this kind of algorithms utilizing fuzzy models one can find e.g. in [3,5] and those utilizing Wiener models – e.g. in [2,10].

If the model used in the MPC algorithm is linear then the optimization problem (1–4) is a standard quadratic programming problem [4,9,15,18]. It is because the superposition principle can be applied and the vector of predicted output values $\boldsymbol{y}$ is given by the following formula:

$$\boldsymbol{y} = \widetilde{\boldsymbol{y}} + \boldsymbol{A} \cdot \Delta \boldsymbol{u} \ , \tag{5}$$

where $\widetilde{\boldsymbol{y}} = \left[\widetilde{y}_{k+1|k}, \ldots, \widetilde{y}_{k+p|k}\right]$ is a free response (contains future values of the output signal calculated assuming that the control signal does not change in the prediction horizon); $\boldsymbol{A} \cdot \Delta \boldsymbol{u}$ is the forced response (depends only on future changes of the control signal (decision variables));

$$\boldsymbol{A} = \begin{bmatrix} a_1 & 0 & \ldots & 0 & 0 \\ a_2 & a_1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_p & a_{p-1} & \ldots & a_{p-s+2} & a_{p-s+1} \end{bmatrix} \tag{6}$$

is the dynamic matrix composed of coefficients of the control plant step response a_i; see e.g. [4,9,15,18].

Let us introduce the vector $\overline{\boldsymbol{y}} = [\overline{y}_k, \ldots, \overline{y}_k]$ of length p. The performance function from (1), after application of the prediction (5), can be rewritten in the matrix–vector form:

$$J_{\text{LMPC}} = (\overline{\boldsymbol{y}} - \widetilde{\boldsymbol{y}} - \boldsymbol{A} \cdot \Delta \boldsymbol{u})^T \cdot (\overline{\boldsymbol{y}} - \widetilde{\boldsymbol{y}} - \boldsymbol{A} \cdot \Delta \boldsymbol{u}) + \Delta \boldsymbol{u}^T \cdot \boldsymbol{\Lambda} \cdot \Delta \boldsymbol{u} \ , \tag{7}$$

where $\boldsymbol{\Lambda} = \lambda \cdot \boldsymbol{I}$ is the $s \times s$ matrix. The performance function (7) depends quadratically on decision variables $\Delta\boldsymbol{u}$. Thus, the optimization problem is in this case a quadratic one. Moreover, if the constraints need not be taken into consideration, the vector minimizing this performance function is given by the following formula:

$$\Delta\boldsymbol{u} = \left(\boldsymbol{A}^T \cdot \boldsymbol{A} + \boldsymbol{\Lambda}\right)^{-1} \cdot \boldsymbol{A}^T \cdot (\overline{\boldsymbol{y}} - \widetilde{\boldsymbol{y}}) \ . \tag{8}$$

The advantages offered by the quadratic optimization led to design of MPC algorithms based on linear approximations of the nonlinear process models obtained at each iteration; see e.g. [8,18]. The algorithms of this type based on fuzzy process models one can find e.g. in [11,12,13].

3 Efficient MPC Algorithms Based on Fuzzy Wiener Models

The Wiener process model (Fig. 1) with fuzzy static block is considered. It is assumed that the static part of the model is a fuzzy Takagi–Sugeno model which consists of the following rules:

Rule j: if v_k is M_j, then

$$y_k^j = g_j \cdot v_k + h_j, \tag{9}$$

where g_j, h_j are coefficients of the model, M_j are fuzzy sets, $j = 1, \ldots, l$, l is the number of fuzzy rules (local models).

The output of the static part of the model is described by the following formula:

$$\widehat{y}_k = \sum_{j=1}^{l} w_j(v_k) \cdot y_k^j \ , \tag{10}$$

where $\widehat{y}_k$ is the output of the static block (and the output of the Wiener model), v_k is the input to the static block and the output of the dynamic block, $w_j(v_k)$ are weights obtained using fuzzy reasoning (see e.g. [14,17]). Therefore, the output of the Wiener model can be described by:

$$\widehat{y}_k = \widetilde{g}_k \cdot v_k + \widetilde{h}_k \ , \tag{11}$$

where $\widetilde{g}_k = \sum_{j=1}^{l} w_j(v_k) \cdot g_j$, $\widetilde{h}_k = \sum_{j=1}^{l} w_j(v_k) \cdot h_j$ It is assumed that the dynamic part of the model is a difference equation (a model often used in linear dynamic block of the Wiener models):

$$v_k = b_1 \cdot v_{k-1} + \ldots + b_n \cdot v_{k-n} + c_1 \cdot u_{k-1} + \ldots + c_m \cdot u_{k-m} \ , \tag{12}$$

where b_j, c_j are parameters of the linear model.

Thus, the output of the Wiener model is given by the following formula:

$$\widehat{y}_k = \widetilde{g}_k \cdot \left(\sum_{j=1}^{n} b_j \cdot v_{k-j} + \sum_{j=1}^{m} c_j \cdot u_{k-j} \right) + \widetilde{h}_k \ , \tag{13}$$

In the proposed approach the fuzzy (nonlinear) Wiener model is used to obtain the free response of the plant. Thanks to the structure of the Wiener model it can be obtained in a straightforward way.

The output of the linear part of the model in the $(k+i)^{\text{th}}$ sampling instant calculated after assumption of constant control values ($u_k = u_{k+1} = \ldots = u_{k+p}$) is described by the following formula:

$$\widehat{v}_{k+i} = \sum_{j=1}^{n} b_j \cdot \widehat{v}_{k-j+i} + \sum_{j=1}^{i} c_j \cdot u_k + \sum_{j=i+1}^{m} c_j \cdot u_{k-j+i} \ , \tag{14}$$

where $\widehat{v}_{k+i}$ are values of the internal signal of the model obtained after assumption of constant control values. The free response is calculated taking into consideration also the estimated disturbances and modeling errors:

$$d_k = y_k - \widehat{y}_k \ . \tag{15}$$

The final formula describing the elements of the free response is, thus, as follows:

$$\widetilde{y}_{k+i|k} = \widetilde{g}_k \cdot \left(\sum_{j=1}^{n} b_j \cdot \widehat{v}_{k-j+i} + \sum_{j=1}^{i} c_j \cdot u_k + \sum_{j=i+1}^{m} c_j \cdot u_{k-j+i} \right) + \widetilde{h}_k + d_k \ , \tag{16}$$

where d_k is the DMC–type disturbance model, i.e. it is assumed the same on the whole prediction horizon.

Next, the dynamic matrix, needed to predict the influence of the future control changes should be derived. It can be done in a straightforward way. First, one should obtain the step response coefficients of the dynamic part of the Wiener model a_n ($n = 1, \ldots, p_d$; p_d is the dynamics horizon equal to the number of sampling instants after which the step response can be assumed as settled). Then, the proper value of gain must be derived. It can be noticed that it can be approximated by:

$$dy = \frac{\left(\left(\sum_{j=1}^{l} w_j(v_k) \cdot (g_j \cdot v_k + h_j) \right) - \left(\sum_{j=1}^{l} w_j(v_{k-}) \cdot (g_j \cdot (v_{k-}) + h_j) \right) \right)}{dv} \ , \tag{17}$$

where $v_{k-} = v_k - dv$, dv is a small number. Thus, at each iteration of the algorithm the following linear approximation of the fuzzy Wiener model (13) is used:

$$\widehat{y}_k = dy \cdot \left(\sum_{n=1}^{p_d - 1} a_n \cdot \Delta u_{k-n} + a_{p_d} \cdot u_{k-p_d} \right) \ . \tag{18}$$

The dynamic matrix will be therefore described by the following formula:

$$\boldsymbol{A}_k = dy \cdot \begin{bmatrix} a_1 & 0 & \dots & 0 & 0 \\ a_2 & a_1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_p & a_{p-1} & \dots & a_{p-s+2} & a_{p-s+1} \end{bmatrix} . \tag{19}$$

The free response (16) and the dynamic matrix (19) are used to obtain the prediction:

$$\boldsymbol{y} = \widetilde{\boldsymbol{y}} + \boldsymbol{A}_k \cdot \Delta \boldsymbol{u} \ . \tag{20}$$

After application of prediction (20) to the performance function from (1), one obtains:

$$J_{\mathrm{FMPC}} = (\overline{\boldsymbol{y}} - \widetilde{\boldsymbol{y}} - \boldsymbol{A}_k \cdot \Delta \boldsymbol{u})^T \cdot (\overline{\boldsymbol{y}} - \widetilde{\boldsymbol{y}} - \boldsymbol{A}_k \cdot \Delta \boldsymbol{u}) + \Delta \boldsymbol{u}^T \cdot \boldsymbol{\Lambda} \cdot \Delta \boldsymbol{u} \ . \tag{21}$$

Thus, as in the case of the MPC algorithm based on a linear model, a quadratic optimization problem is obtained.

4 Testing of the Proposed Approach

4.1 Control Plant – Description and Fuzzy Modeling

The control plant under consideration is a valve for control of fluid flow. It is described by the following Wiener model [1,6]:

$$v_k = 1.4138 \cdot v_{k-1} - 0.6065 \cdot v_{k-2} + 0.1044 \cdot u_{k-1} + 0.0883 \cdot u_{k-2} \ , \tag{22}$$

$$y_k = \frac{0.3163 \cdot v_k}{\sqrt{0.1 + 0.9 \cdot (v_k)^2}} \ , \tag{23}$$

where u_k is the pneumatic control signal applied to the stem, v_k is the stem position (it is the output signal of the linear dynamic block and the input signal of the nonlinear static block), y_k is flow through the valve (it is the output of the plant). The static part of the model was approximated using the fuzzy model. It was done heuristically because the nonlinear function in the control plant model (23) resembles the sigmoid function; see Fig. 2.

As a result of a few experiments the simple fuzzy model of the statics of the control plant was obtained. It consists of two rules:

Rule 1: if v_k is M_1, then

$$y_{k+1}^1 = -0.3289, \tag{24}$$

Rule 2: if v_k is M_2, then

$$y_{k+1}^2 = 0.3289. \tag{25}$$

The assumed membership functions are shown in Fig. 3. Fuzzy approximation of the static nonlinearity is presented as the dashed line in Fig. 2.

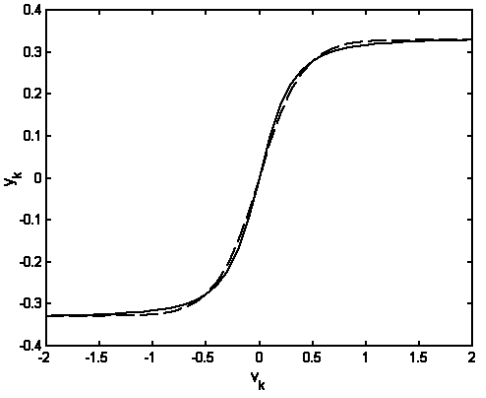


Fig. 2. Static characteristic of the valve; solid line – original model, dashed line – fuzzy approximation

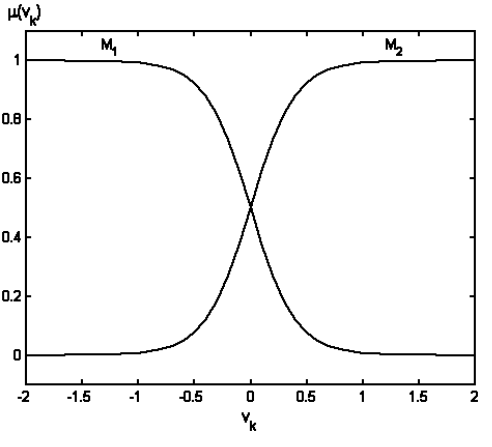


Fig. 3. Membership functions in the fuzzy model of the static characteristic of the valve

4.2 Simulation Experiments

The operation of the proposed MPC algorithm was compared with other approaches. Thus, three MPC algorithms were designed for the considered control plant:

1. LMPC – with a linear model,
2. NMPC – with nonlinear optimization,
3. FMPC – with prediction based on fuzzy Wiener model.

Tuning parameters of all three algorithms were assumed the same and: prediction horizon $p = 30$, control horizon $s = 15$, weighting coefficient $\lambda = 4$.

Performance of control systems with LMPC, NMPC and FMPC algorithms was compared. The example responses obtained after changes of the set–point value from 0 to 0.3 at the beginning of the experiment and then back from 0.3 to 0 in the half of the experiment are shown in Fig. 4. The LMPC algorithm gives the worst responses (dashed lines in Fig. 4). They are much slower than those obtained using other MPC algorithms. The responses obtained in the control systems with the FMPC (solid lines in Fig. 4) and NMPC algorithms (dotted lines in Fig. 4) are very similar. However, in the FMPC algorithm the control signal is generated much faster as a solution of the quadratic programming problem.

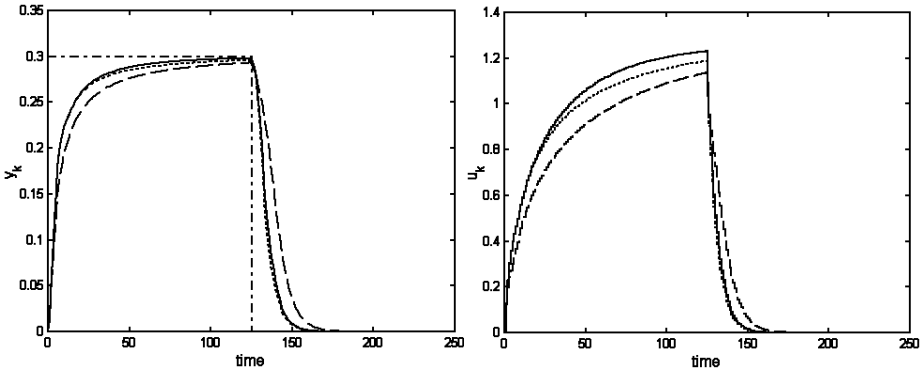


Fig. 4. Responses of the control systems to the changes of the set–point value to $\overline{y}_1 = 0.3$ and $\overline{y}_2 = 0$; FMPC (solid lines), NMPC (dotted lines), LMPC (dashed lines); dash–dotted line – set–point signal; right – output signal, left – control signal

5 Summary

The MPC algorithms proposed in the paper are based on fuzzy Wiener models. They use the nonlinear process model to derive the free response and its linear approximation to derive the forced response. Thanks to the form of the control plant model the prediction is easy to derive. The proposed algorithms are formulated as the efficient linear–quadratic optimization problems but they offer practically the same performance as the algorithms consisting in nonlinear optimization outperforming their counterparts based on linear process models.

Acknowledgment. This work was supported by the Polish national budget funds for science 2009–2011.

References

1. Al-Duwaish, H., Karim, M.N., Chandrasekar, V.: Use of multilayer feedforward neural networks in identification and control of Wiener model. IEE Proceedings on Control Theory and Applications 143, 255–258 (1996)
2. Arefi, M.M., Montazeri, A., Poshtan, J., Jahed-Motlagh, M.R.: Wiener-neural identification and predictive control of a more realistic plug-flow tubular reactor. Chemical Engineering Journal 138, 274–282 (2008)
3. Babuska, R., te Braake, H.A.B., van Can, H.J.L., Krijgsman, A.J., Verbruggen, H.B.: Comparison of intelligent control schemes for real-time pressure control. Control Engineering Practice 4, 1585–1592 (1996)
4. Camacho, E.F., Bordons, C.: Model Predictive Control. Springer, Heidelberg (1999)
5. Fink, A., Fischer, M., Nelles, O., Isermann, R.: Supervision of nonlinear adaptive controllers based on fuzzy models. Control Engineering Practice 8, 1093–1105 (2000)
6. Hsu, Y.-L., Wang, J.-S.: A Wiener-type recurrent neural network and its control strategy for nonlinear dynamic applications. Journal of Process Control 19, 942–953 (2009)
7. Janczak, A.: Identification of nonlinear systems using neural networks and polynomial models: a block-oriented approach. Springer, Heidelberg (2005)
8. Lawrynczuk, M.: A family of model predictive control algorithms with artificial neural networks. International Journal of Applied Mathematics and Computer Science 17, 217–232 (2007)
9. Maciejowski, J.M.: Predictive control with constraints. Prentice Hall, Harlow (2002)
10. Mahmoodi, S., Poshtan, J., Jahed-Motlagh, M.R., Montazeri, A.: Nonlinear model predictive control of a pH neutralization process based on Wiener-Laguerre model. Chemical Engineering Journal 146, 328–337 (2009)
11. Marusak, P., Tatjewski, P.: Effective dual-mode fuzzy DMC algorithms with on-line quadratic optimization and guaranteed stability. International Journal of Applied Mathematics and Computer Science 19, 127–141 (2009)
12. Marusak, P.: Advantages of an easy to design fuzzy predictive algorithm in control systems of nonlinear chemical reactors. Applied Soft Computing 9, 1111–1125 (2009)
13. Marusak, P.: Efficient model predictive control algorithm with fuzzy approximations of nonlinear models. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) ICANNGA 2009. LNCS, vol. 5495, pp. 448–457. Springer, Heidelberg (2009)
14. Piegat, A.: Fuzzy Modeling and Control. Physica-Verlag, Berlin (2001)
15. Rossiter, J.A.: Model-Based Predictive Control. CRC Press, Boca Raton (2003)
16. Shafiee, G., Arefi, M.M., Jahed-Motlagh, M.R., Jalali, A.A.: Nonlinear predictive control of a polymerization reactor based on piecewise linear Wiener model. Chemical Engineering Journal 143, 282–292 (2008)
17. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modeling and control. IEEE Trans. Systems, Man and Cybernetics 15, 116–132 (1985)
18. Tatjewski, P.: Advanced Control of Industrial Processes; Structures and Algorithms. Springer, London (2007)

Multicriteria Subjective Reputation Management Model

Michał Majdan[1,2] and Włodzimierz Ogryczak[1]

[1] Institute of Control and Computation Engineering, Warsaw University of Technology, Warsaw, Poland
[2] National Institute of Telecommunications, Warsaw, Poland

Abstract. The most widely used reputation models assume uniform users' preference structure. In this paper a new reputation management model is presented. It is focused on aggregation of community wide reputation in situation when agents do not share the same preference structure. The reputation is interpreted as vectors of attributes that represent several reputation evaluation criteria. Outcomes of the criteria are transformed by the utility functions and assigned subjective probabilities so that the subjective expected utility values can be obtained. Subjective expected utilities are further aggregated by the weighted ordered weighted average (WOWA) operator. The expressive power of subjective utilities concept along with the WOWA aggregation technique provides the reputation management system with a capability to model various preference structures. It is shown with an illustrative example.

1 Introduction

The problem of providing trust in virtual communities has drawn much attention ever since the widespread development of Web 2.0 applications. The purpose of trust and reputation systems is to strengthen the quality of markets and communities by providing an incentive for good quality services, and by sanctioning low quality services. One of key issues of any trust management system is how it deals with reputation. Reputation is defined as all available information about certain agent in a given community communicated by members of this community. An individual's subjective trust can be derived from a combination of received referrals and personal experience. There has been developed a number of reputation management models [7]. However, there are still valid questions that remain unanswered. Especially, if gathered reputation information can be left for an agent to process it according to his individual preferences. This is hard to accept, especially for human agents, due to the amount of reputation information (outcomes) to be analyzed. Therefore, many models ignore agents' subjective preference structures assuming that all participants of a network share the same view on available evidence and enforcing unified reputation measures.

This paper aims to present a solution to the question on how to automatically aggregate the reputation to support the decision of an agent while following its preference attitude. The proposed model allows to look at agents trust in the

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 678–687, 2010.

point of multiple evaluation criteria. It maintains the original multiattribute reputation outcome distributions for evaluation criteria and enables the trusting agents to express their preference on the outcomes when estimating trust in other agents. The model makes it possible for the agents to put importance weights on the evaluation criteria. It combines the Weighted Ordered Weighted Average (WOWA) operator with the concepts of subjective expected utility theory to build the model that can be adjusted to the users' preferential structures.

The idea of application of fuzzy aggregation operators to the problem of reputation management has appeared in various works. For example, the Ordered Weighted Average (OWA) [1] or the WOWA [2] usage were analyzed within this context. However, both these approaches do not allow to take into account individual agent preferences as the fuzzy aggregation operators are applied to unified scalar reputation scores. Lee *et al.* [8] developed a fuzzy trust model which takes into account both evaluations from multiple criteria and the recommendations from others in order to set the trust degrees on entities. In the model, the entity's preference degrees on the outcomes of the interactions are expressed in fuzzy sets and the trust degrees are determined by aggregating the satisfaction degrees with respect to evaluation criteria with Sugeno fuzzy integral. The WOWA aggregation used in our model provides, however, much simpler and more transparent agents preference modeling. Moreover, it opens a possibility to incorporate into the reputation analysis the multicriteria decision support techniques such as the Reference Point Method (RPM) [17]. The RPM interactive analysis is navigated with the commonly accepted control parameters expressing reference levels for the individual criteria and it can be based on the WOWA aggregation of appropriate achievement measures [9,10].

2 Subjective Probability

There exist a number of probability interpretations [6], however the notion of probability has basically dual understanding. The first concept of probability is based on the observation of relative frequency of outcomes in the repeated experiments. As the number of experiments increases the parameters of such empirical distribution approach the real "objective" probability distribution of the outcomes. Apart from the above "Bernoulli type" of probability the other, equally old interpretation of probability states that the probability reflects beliefs that certain outcome will obtain. This view of probability lead Ramsey [12] to the concept of "subjective probability", that was further formalized by de Finetti [3]. The general assumption that allows to use agents' subjective probabilities is that they follow probability calculus rules.

2.1 Savage's Subjective Expected Utility Theory

Savage [14] has proposed framework to deal with subjective probabilities and utilities. Savage's framework consists of the following elements:

- *states of the world* as possible scenarios of the future with only one *true state*;
- *consequences* entities, that have value to the decision maker;
- *acts* functions that associate consequences with states;
- *events* the subsets of state space.

Moreover, Savage has developed a set of 7 postulates that define the preference structure:

1. The preference relation between acts is complete (each two acts are comparable) and transitive.
2. The decision between acts is based only on the consequences in the states when the consequences are distinct.
3. The ordering of consequences is state and act independent.
4. The decision maker assigns probabilities to events with no regard to the consequences. Other words, the subjective probability of an event will not change even if the payoff's will change (preserving the ordering).
5. There exists at least one act that is preferred to some other act.
6. The state space is continuous. It is always possible to divide an event into smaller sub events and adjust probabilities accordingly.
7. The act "better", on each of the states of a certain event, then the other act is strictly preferred.

The preference relation is defined by the first four axioms, last three play rather technical role. Savage claims that the preference relation described by above seven postulates is analogous to the problem of expected utility maximization, when the utility function is defined on the set of consequences and the (subjective) probability measure is defined on the set of all events.

3 Multicriteria Trust Model Definition

Proposed reputation management model is based on the assumption of existence of subjective expected utility and subjective probabilities especially. The process of calculating reputation metric can be viewed as a process of calculating utilities on two levels. The first level of computations is done with respect to a single criterion. This involves deriving and applying utility function on the set of possible outcomes with subjective probabilities based on the reputation. The above leads us to the formulation of expected utility of the outcomes as a satisfaction level of the given criteria. The second level is the decision problem of selecting the best option (most trustworthy) among a number of offers, each described by a set of evaluation criteria. One can employ a variety of methods of multidimensional analysis to solve this problem. Interactive methods, like the reference point methods, can be used as well as the expected utility maximization approach can be applied.

3.1 Satisfaction Levels

Let us assume we have a set C of n possible criteria. For each criterion c_i ($i \in C$) there is a set of possible outcomes O^i. Each of $o \in O^i$ has a subjective probability assigned that reflects users belief on certain value to occur in the next interaction. There is a preference relation $\succeq$ on set O^i that follows usual assumptions of transitivity and completeness. The expression $o_1 \succ o_2$ means that the o_1 is more desirable then o_2 and $o_1 \sim o_2$ means that o_1 is equally desirable as o_2. If we consider the above model with respect to the language of Savage's framework then the state space can be regarded as a space of all possible transaction results. Each possible result of the transaction can be assigned one of the values from the set of consequences O^i. The assignment is done by the selection of a given option (act). Each option is comparable with the others. The ranking of options depends only on the ranking of possible transactions results where either the probability or the outcome are different. The ranking of outcomes is defined above and is independent of the transactions that yield them. The probabilities are calculated based on the reputation reports. The above defined preference structure on acts follow Savage's axioms of rationality of preference structure thus modeling it with the subjective expected utility is justified.

Set R^i is the set of evaluations (reputation) that refer to the criterion c_i Set R^i_o is a set of evaluations with outcome o of the criterion c_i The subjective probability of an outcome o corresponds to the available reputation and is defined as:

$$sp(o) = \frac{|R^i_o|}{|R^i|}$$

This is not the objective probability measure as the set of reputation valuations is not a set of independent repetitive trials of some phenomenon. Instead it reflects user's attitude towards the possible outcome of the transaction, since in the reputation system past interactions and reputation are the only source of knowledge that influence user's beliefs. Probabilities $sp(o)$ are calculated for all outcomes. If a given outcome has never occurred then the probability is assumed as equal to 0.

The degree of preference relation between outcomes has to be incorporated into the preference structure also. In order to reasonably model user's preferences we need to transform them into a common measurement scale that is going to reflect the preferential order of outcomes as well as to measure the strength of the relation. Let the measurement (utility) function be denoted by u. Utilities are normalized to sum up to one. The utility function needs to be consistent with the preference relation $\succeq$ on set the of outcomes O. The Subjective Expected Utility of a particular criterion c_i is finally calculated as

$$x_i = \frac{1}{|R^i|} \sum_{o \in O^i} |R^i_o| u(o) \tag{1}$$

3.2 Reputation Score

In the reputation management model presented in this paper the reputation scores are generated by aggregating subjective expected utilities of the criteria using the WOWA operator. The standard *Ordered Weighted Average* (OWA) operator [18] allows one to introduce preferential weights assigned to the ordered values of the aggregated vector elements rather than particular elements of the vector. Formal definition of the operator is as follows. Given vector of n values x_i for $i = 1, \ldots, n$ and preferential weights vector $w_i \geq 0$ for $i = 1, \ldots, n$ while $\sum_{i=1}^{n} w_i = 1$. The OWA operator is defined as:

$$\sum_{i=1}^{n} w_i x_{\sigma(i)}$$

where $\sigma(i)$ is a permutation ordering vector x from the largest to the smallest element:

$$x_{\sigma(1)} \geq x_{\sigma(2)} \geq x_{\sigma(3)} \geq \ldots \geq x_{\sigma(n)}.$$

The OWA operator allows one to model various aggregation functions from the maximum through the arithmetic mean to the minimum. Thus, it enables modeling various preferences from the optimistic to the pessimistic one. On the other hand, the OWA does not allow one to allocate any importance weights to specific criteria. Several attempts have been made to incorporate importance weighting into the OWA operator. Finally, Torra [15] has incorporated importance weighting into the OWA operator within the Weighted OWA (WOWA) aggregation. The WOWA averaging is based on two weighting vectors:

- preferential weights vector $\boldsymbol{w}$ ($w_i \geq 0$, $\sum_i w_i = 1$) associated with criteria satisfaction level values ordered from the highest value to the lowest.
- importance weights vector $\boldsymbol{p}$ ($p_i \geq 0$, $\sum_i p_i = 1$) associated with the aggregated criteria.

Actually, the WOWA average is a particular case of Choquet integral using a distorted probability as the measure [16].

Formal WOWA definition follows the OWA formula aggregating the values ordered from the highest to the lowest one:

$$WOWA(x_1, \ldots, x_n) = \sum_{i=1}^{n} \omega_i x_{\sigma(i)} \tag{2}$$

while weights ω are constructed by cumulation of the preferential weights $w_{\sigma(i)}$ and their decumulation according to the corresponding distribution of importance weights $p_{\sigma(i)}$, i.e.,

$$\omega_i = w^*(\sum_{j \leq i} p_{\sigma(j)}) - w^*(\sum_{j < i} p_{\sigma(j)}) \tag{3}$$

where function w^* interpolates points $(i/n, \sum_{j \leq i} w_j)$ and point $(0, 0)$. When preferential weights p_i are equal, WOWA becomes the standard OWA operator

with preferential weights w_i. When preferential weights are equal, the WOWA operator becomes the weighted average operator. The WOWA aggregation generalizes both the OWA and the weighted average.

Alternatively, the WOWA aggregation may be given by the formula [11]:

$$WOWA(x) = \sum_{k=1}^{n} w_k n \int_{(k-1)/n}^{k/n} \overline{F}_x^{(-1)}(\xi)\, d\xi \tag{4}$$

where $\overline{F}_x^{(-1)}$ is the stepwise function $\overline{F}_x^{(-1)}(\xi) = x_{\sigma(i)}$ for $\alpha_{i-1} < \xi \leq \alpha_i$ with breakpoints $\alpha_i = \sum_{k \leq i} p_{\tau(k)}$ and $\alpha_0 = 0$. It can also be mathematically formalized as the left-continuous inverse $\overline{F}_x^{(-1)}(\xi) = F_x^{(-1)}(1-\xi)$ of the cumulative distribution function

$$F_x(d) = \sum_{i=1}^{n} p_i \delta_i(d) \quad \text{where} \quad \delta_i(d) = \begin{cases} 1 & \text{if } x_i \leq d \\ 0 & \text{otherwise} \end{cases}$$

Note that $n \int_{(k-1)/n}^{k/n} \overline{F}_x^{(-1)}(\xi)\, d\xi$ represents the average within the k-th portion of $1/n$ largest outcomes, the corresponding conditional mean. Hence, formula (4) defines WOWA aggregations with preferential weights $\boldsymbol{w}$ as the corresponding OWA aggregation but applied to the conditional means calculated according to the importance weights $\boldsymbol{p}$ instead of the original outcomes.

In case of the reputation management model presented in this paper subjective expected utilities of the criteria are aggregated using the WOWA operator. Having decided about the weighting vectors, following the WOWA formula (2)–(3), weights ω are calculated and the final value of the WOWA aggregation is derived. The calculated reputation score of each alternative agent is used to rank them in decreasing order. The agent with the highest score should be the most trusted one according to the user's preference structure.[1]

The definition of weights $\boldsymbol{w}$ induces certain shape of function w^* and thereby allows us to model the user's attitude towards given decision situation [11]. Increasing weights w_i (convex function w^*) shows user's diffident approach as it amplifies the impact of low values while reducing the importance of higher values. It requires all satisfaction levels be high enough to yield high aggregation value. On the other hand, decreasing sequence of weights w_i (concave function w^*) is bound to the confident attitude since it amplifies higher values. It allows any of the criteria be highly satisfied to yield a high aggregated value. The whole range of w^* shapes can be interpreted as a variety of users preference structures what makes WOWA aggregation so well suited for this sort of applications.

4 Illustrative Example

Auction service data usually allows user to leave some kind of rating for the other party of the transaction. Different rules may govern this process. Sometimes only

[1] In general case the actual decision should not rely only on reputation. For example, one should not neglect the impact of agents own experience with evaluated partners.

buyer is entitled to leave comment, sometimes party's comment is not revealed until other agent leaves his own evaluation of the transaction. Ratings are usually limited to the set of positive, neutral or negative evaluation accompanied by the free hand opinion on what actually happened. Auction house *eBay Inc.* has extended feedback system introducing a wider seller rating system. This functionality allows buyers to asses sellers across four dimensions assigning values of one to five to each of the following criteria [4]:

1. How accurate was the item description?
2. How satisfied were you with the seller's communication?
3. How quickly did the seller ship the item?
4. How reasonable were the shipping and handling charges?

Let us further assume that the user (bidder) has to choose the most trustworthy partner among 5 alternative auction users (sellers) who participated in a different numbers of transactions. Namely:

- Alice - Very good performance at describing items but fails at communication while maintaining average performance at shipment and handling charges. Very heavy user took part in the greatest number of transactions
- Bob - Average performer at all criteria. Average user with regard to the number of transactions performed.
- Carl - Outstanding communication and description but poor at any other criteria. Average transactions number.
- David - No handling charges and quick shipment while poor description and average communication skills. Average transactions number.
- Edward - Average at all dimensions but very low number of transactions performed.

Let us assume that we have identified 4 criteria that govern the decision process of auction house user and we are able to determine outcome domain of each criterion.

Description how accurate was the item description?

1. poor (o_1^d) $u(o_1^d) = 1$
2. good (o_2^d) $u(o_2^d) = 5$

	c_1	
	o_1^d	o_2^d
Alice	1	100
Bob	25	25
Carl	2	28
David	28	3
Edward	2	2

Communication how satisfied were you with the seller's communication?

1. poor (o_1^c) $u(o_1^c) = 1$
2. medium(o_2^c) $u(o_2^c) = 2$
3. good(o_3^c) $u(o_3^c) = 3$
4. very good (o_4^c) $u(o_4^c) = 4$
5. outstanding (o_5^c) $u(o_5^c) = 5$

	c_2				
	o_1^c	o_2^c	o_3^c	o_4^c	o_5^c
Alice	40	20	15	2	1
Bob	5	8	8	7	7
Carl	0	0	0	20	10
David	6	14	9	1	0
Edward	1	2	1	1	1

Shipment how much time took the delivery?

1. poor (o_1^s) $u(o_1^s) = 1$
2. medium(o_2^s) $u(o_2^s) = 2$
3. good (o_3^s) $u(o_3^s) = 3$
4. very good (o_4^s) $u(o_4^s) = 4$

	c_3			
	o_1^s	o_2^s	o_3^s	o_4^s
Alice	2	40	40	2
Bob	4	12	12	8
Carl	19	6	1	0
David	0	0	10	20
Edward	0	2	2	1

Handling how reasonable were the shipping and handling charges?

1. poor (o_1^h) $u(o_1^h) = 1$
2. medium (o_2^h) $u(o_2^h) = 2$
3. good (o_3^h) $u(o_3^h) = 3$
4. very good (o_4^h) $u(o_4^h) = 4$
5. outstanding (o_5^h) $u(o_5^h) = 5$

	c_4				
	o_1^h	o_2^h	o_3^h	o_4^h	o_5^h
Alice	5	30	40	30	5
Bob	1	12	18	13	1
Carl	24	9	2	0	0
David	0	0	0	0	30
Edward	1	3	3	1	0

User specifies his preferences first by assigning utilities to the outcomes as it was described above. The selection of utilities leads to the following subjective expected utilities:

criterion:	c_1	c_2	c_3	c_4
Alice	0.827	0.118	0.229	0.200
Bob	0.500	0.206	0.263	0.201
Carl	0.789	0.289	0.119	0.091
David	0.231	0.144	0.394	0.333
Edward	0.500	0.189	0.273	0.167

Consider various results based on the WOWA aggregation defined with by several selections of the vector of preferential weights and importance weights.

OWA aggregation case. We suppose that user has no special preferences with regard to the criteria in question. All of them are equally weighted. Since we have only 4 criteria each element of the $\boldsymbol{p}$ is equal, so $\boldsymbol{p} = [0.25, 0.25, 0.25, 0.25]$. User though has the preferences regarding achievements of the criteria. First, we shall assume user has expressed diffident approach and requires small values to contribute more to the result of aggregation. To achieve this the $\boldsymbol{w}$ vector should be selected in a way that leads to the convex interpolation function w^*. In the case considered $\boldsymbol{w} = [0.03, 0.05, 0.28, 0.65]$.

Weighted mean. We consider case when the reputation system user expresses his indifference with respect to the satisfaction values of the criteria providing vector $\boldsymbol{w} = [0.25, 0.25, 0.25, 0.25]$ and strong preference for the "description conformity" criterion giving vector $\boldsymbol{p} = [0.67, 0.09, 0.04, 0.20]$.

General case. If we now consider the most general case we shall express both preferences with respect to the criteria as well as for the achievement levels. Suppose

the user expressed diffident approach but with the strong preference for the description component. Let us further assume he had provided following values for the weighting vectors: $\boldsymbol{p} = [0.63, 0.19, 0.06, 0.13]$ and $\boldsymbol{w} = [0.06, 0.13, 0.19, 0.63]$. Results of all three experiments are presented in the table below

	Alice	Bob	Carl	David	Edward
owa	0.164	**0.213**	0.127	0.184	0,013
wm	**0.611**	0.403	0.593	0.25	0,051
general	0.339	0.29	**0.349**	0.196	0.017

The first case would lead to the selection of *Bob* while the weighted mean aggregation would prefer *Alice*. The general WOWA aggregation case shows *Carl* as the best choice.

5 Concluding Remarks

The role of the information system based on the presented approach is to retrieve and process data according to the user's preference structure and to provide result as a single scalar rating. What is probably the most important users get instant knowledge on trustfulness of the possible transaction partners while the effort of retrieving and analyzing feedback is done by the system. When there is no extended seller's ratings much can be achieved by analyzing feedback text to retrieve opinions on the selected criteria space. Recent developments in text mining and natural language processing can be exploited in this area. Specifying both utilities of certain outcomes of the criteria as well as two weight vectors ($\boldsymbol{p}$ and $\boldsymbol{w}$) may be confusing for the average user. There is still a lot of work to do on how to retrieve user's individual preferences and transform them into utilities and WOWA weighting vectors [5,16].

The preferential weights definition can be simplified by allowing to introduce scalable preferences with weights allocated to specific portion of the worst outcomes independently from the number of criteria. Formula (4) allows us to define such a generalized WOWA aggregation [11] where the preferential weights w_k are allocated to an arbitrarily defined grid of ordered outcomes defined by m (independent of n) quantile breakpoints $\beta_0 = 0 < \beta_1 < \ldots < \beta_{m-1} < \beta_m = 1$, i.e. the aggregation defined with a piecewise linear function w^*_β interpolating points $(\beta_k, \sum_{i \le k} w_i)$ together with the point (0.0). Moreover, the WOWA aggregation opens a possibility to incorporate into the reputation analysis the multicriteria decision support techniques such as the Reference Point Method (RPM) [17]. The RPM interactive analysis is navigated with the commonly accepted control parameters expressing reference levels for the individual criteria and it can be based on the WOWA aggregation of appropriate achievement measures [9,10].

Acknowledgements

This work has been partially supported by the Polish Ministry of Science and Higher Education under the research grant N N516 4307 33.

References

1. Aringhieri, R., Damiani, E., De Capitani Di Vimercati, S., Paraboschi, S., Samarati, P.: Fuzzy techniques for trust and reputation management in anonymous peer-to-peer systems. Journal of the American Society for Information Science and Technology 57, 528–537 (2006)
2. Damiani, E., De Capitani Vimercati, S., Samarati, P., Viviani, M.: A WOWA-based Aggregation Technique on Trust Values Connected to Metadata. Electronic Notes in Theoretical Computer Science 157, 131–142 (2006)
3. De Finetti, B.: Foresight: its logical laws in subjective sources. In: Studies in Subjective Probability, pp. 93–158 (1964)
4. Ebay Inc.: http://www.ebay.com
5. Filev, D., Yager, R.R.: On the issue of obtaining OWA operator weights. Fuzzy Sets and Systems 94, 157–169 (1998)
6. Hajek, A.: Interpretations of Probability. The Stanford Encyclopedia of Philosophy (2009)
7. Josang, A., Ismail, R., Boyd, C.: A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support Systems 43, 618–644 (2007)
8. Lee, K., Hwang, K., Lee, J., Kim, H.J.: A Fuzzy Trust Model Using Multiple Evaluation Criteria. In: Wang, L., Jiao, L., Shi, G., Li, X., Liu, J. (eds.) FSKD 2006. LNCS (LNAI), vol. 4223, pp. 961–969. Springer, Heidelberg (2006)
9. Ogryczak, W.: Ordered weighted enhancement of preference modeling in the reference point method for multiple criteria optimization. Soft Computing 14, 435–450 (2010)
10. Ogryczak, W., Kozłowski, B.: Reference point method with importance weighted ordered partial achievements. TOP (2009), Springer Online First doi:10.1007/s11750-009-0121-4
11. Ogryczak, W., Śliwiński, T.: On Efficient WOWA Optimization for Decision Support under Risk. International Journal of Approximate Reasoning 50, 915–928 (2009)
12. Ramsey, F.P.: Truth and probability. The Foundations of Mathematics and other Logical Essays 7, 156–198 (1926)
13. Resnick, P., Zeckhauser, R., Friedman, E., Kuwabara, K.: Reputation Systems. Communications of the ACM 43, 45–48 (2000)
14. Savage, L.J.: The Foundations of Statistics. Wiley, New York (1954)
15. Torra, V.: The Weighted OWA Operator. International Journal of Inteligent Systems 12, 153–166 (1997)
16. Torra, V., Narukawa, Y.: Modeling Decisions: Information Fusion and Aggregation Operators. Springer, Berlin (2007)
17. Wierzbicki, A.P.: Reference point approaches. In: Gal, T., Stewart, T., Hanne, T. (eds.) Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications, pp. 9.1–9.39. Kluwer, Dordrecht (1999)
18. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decisionmaking. IEEE Transactions on Systems, Man and Cybernetics 18, 183–190 (1988)

Application of Fuzzy Preference Based Rough Set Model to Condition Monitoring

Xiaomin Zhao, Ming J. Zuo*, and Tejas Patel

Department of Mechanical Engineering, University of Alberta,
Edmonton, AB, T6G 2G8, Canada
Phone: (780) 492-4466; Fax: (780) 492-2200
ming.zuo@ualberta.ca

Abstract. Parameters that vary monotonically with fault development are useful in condition monitoring, but not easy to find especially for complex systems. A method using fuzzy preference based rough set model and principle component analysis (PCA) is proposed to generate such an indicator. The fuzzy preference based rough set model is employed to evaluate the monotonic trends of features reflecting machinery conditions. PCA is used to condense the informative features and generate an indicator which can represent the development of machine health condition. The effectiveness of the proposed method is tested for damage level detection of an impeller in a slurry pump.

Keywords: Fuzzy preference based rough set, PCA, Condition monitoring.

1 Introduction

Condition monitoring plays a key role in safe running of machinery. It is a major component of predictive maintenance, and is useful in maintenance scheduling or other actions to prevent major failures. During the condition monitoring process, a set of parameters that are observable and sensitive to the health of the machinery are monitored. Deviation from a reference value and/or trend of a parameter is detected to identify the development of a malfunction or fault.

To assess the progression of faults, parameters that are most representative of the health condition are needed. Furthermore, the parameters or indicators that have monotonic trends reflecting the health condition are the most desirable. Yesilyurt and Ozturk [1] found that the average mean frequency of the scalograms yielded a consistent trend which reflected the progression of tool wear in milling. Zhang et al [2] proposed a feature extraction method for bearing fault detection. The extracted feature had a monotonic decreasing trend as the dimension of fault increased. However, the parameters that vary monotonically with fault development are not easy to find especially for complex systems. Natke and Cempel [3] found that the singular values and singular vectors were informative parameters for fault detection and evaluation; but the relationship between singular vectors and physical faults was lacking. A criterion

* Corresponding author.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 688–697, 2010.

is, therefore, needed to evaluation the monotonic relevance between indicators and the development of faults.

Rough set has proved to be an effective tool for feature evaluation. Particularly, fuzzy rough set models have attracted significant attention as they perform well for numerical feature selection. However most of these models don't consider the preference relationship between features and decisions [4, 5]. To consider this aspect, Greco et al [6] introduced a dominance rough set model. Hu et al [7] presented a fuzzy rough set model based on fuzzy preference relations which could reflect the degree of preference quantitatively. Because of its ability to evaluate the preference degree, this model provided a helpful criterion for estimating the monotonic relevance between indicators and the development of faults in condition monitoring.

The feature space, which may be reduced through feature selection, is usually multidimensional. Each feature in the feature space contains complementary information on machine conditions. To improve the separateness, clustering ability and robustness of the features in the feature space, feature fusion should be conducted. Principal component analysis (PCA) performs well in the area of feature fusion [8].

In this paper, we propose a method that uses both fuzzy preference based rough set model and PCA. This proposed method is tested on tracking of impeller damage in a centrifugal pump.

2 Fuzzy Preference Based Rough Sets

2.1 Dominance Rough Sets

Let $IS=<U, A>$ be an information system, where U is a nonempty and finite set of samples $\{x_1, x_2, \ldots x_n\}$, and A is a set of features $\{ a_1, a_2, \ldots a_m\}$ which characterize the samples. $<U, A>$ is also called a decision table if $A=C \cup D$ where C is the set of features which describe the samples' characteristics, and D is the decision which classifies the samples' labels.

Given $\forall x, y \in U$, if y is not worse than x regarding $B \subseteq A$ for $\forall a \in B$, the relation is denoted by $y \geq_B x$. Similarly $y \leq_B x$ denotes the case that y is not better than x regarding $B \subseteq A$ for $\forall a \in B$. Furthermore, the following sets are associated:

$$[x]_B^{\geq} = \{ y \in U : y \geq_B x \}; \tag{1}$$

$$[x]_B^{\leq} = \{ y \in U : y \leq_B x \}. \tag{2}$$

The first set, $[x]_B^{\geq}$, consists of samples that are not worse than x with respect to feature subset B. The second set, $[x]_B^{\leq}$, consists of samples that are not better than x with respect to feature subset B.

Assume that $d_1 \leq d_2 \leq \ldots \leq d_p$ and let $d_i^{\geq} = \bigcup_{j \geq i} d_j$ and $d_i^{\leq} = \bigcup_{j \leq i} d_j$. The lower and upper approximations of these sets are given in [6] as follows:

- upward approximation: $\underline{B}_{d_i^{\geq}}^{\geq} = \{x : [x]_B^{\geq} \subseteq d_i^{\geq}\}$ and $\overline{B}_{d_i^{\geq}}^{\geq} = \{x : [x]_B^{\geq} \bigcap d_i^{\geq} \neq \phi\}$;
- downward approximation: $\underline{B}_{d_i^{\leq}}^{\leq} = \{x : [x]_B^{\leq} \subseteq d_i^{\leq}\}$ and $\overline{B}_{d_i^{\leq}}^{\leq} = \{x : [x]_B^{\leq} \bigcap d_i^{\leq} \neq \phi\}$.

Upward approximations reflect the degree of monotonic relevance in the sense that when the feature of sample *y* is not worse than the feature of sample *x*, the decision of *y* should not be worse than the label of *x*. Downward approximations reflect the degree of monotonic relevance in the sense that when the feature of sample *y* is not better than the feature of sample *x*, the decision of *y* should not better than the label of *x*.

2.2 Fuzzy Preference Based Rough Sets

A fuzzy preference relation *R* is a fuzzy set on the product set $U \times U$, which is characterized by a membership function $\mu_R : U \times U \to [0,1]$. If *U* is finite set, the fuzzy preference relation can also be represented by an $n \times n$ matrix $(r_{ij})_{n \times n}$, where r_{ij} is interpreted as the preference degree of x_i over x_j. We use $r_{ij}=1/2$ to indicate that there is no difference between x_i and x_j; $r_{ij}>1/2$ to indicate that x_i is preferred to x_j; $r_{ij}=1$ means x_i is absolutely preferred to x_j; and $r_{ij}<1/2$ shows x_j is preferred to x_i.

Assume that $U=\{x_1, x_2, \ldots x_n\}$, *a* is a numerical feature which describes the samples in *U,* and the feature value of sample *x* is $f(x, a)$. In [7], the upward and downward fuzzy preference relations over *U* are computed by

$$r_{ij} = \frac{1}{1+e^{-k(f(x_i,a)-f(x_j,a))}}, \text{ and } r_{ij} = \frac{1}{1+e^{k(f(x_i,a)-f(x_j,a))}}$$

where *k* is a parameter to adjust the shape of the membership function. By employing membership functions, fuzzy preference relations not only reflect the fact that sample x_i is greater (smaller) than x_j, but also reflect how much x_i is greater (smaller) than x_j.

With fuzzy preference relations $R^{>}$ and $R^{<}$ and preference decision label $d_i^{\geq}$ and $d_i^{\leq}$ given, the membership of sample *x* to the lower and upper approximations of $d_i^{\geq}$ and $d_i^{\leq}$ are defined in [7] as

- upward fuzzy lower approximation: $\underline{R}^{>}_{d_i^{\geq}(x)} = \inf_{u \notin U} \max\{1 - R^{>}(x,u), d_i^{\geq}(u)\}$
- upward fuzzy upper approximation: $\overline{R}^{>}_{d_i^{\geq}(x)} = \sup_{u \in U} \min\{R^{>}(x,u), d_i^{\geq}(u)\}$
- downward fuzzy lower approximation: $\underline{R}^{<}_{d_i^{\leq}(x)} = \inf_{u \notin U} \max\{1 - R^{<}(x,u), d_i^{\leq}(u)\}$
- downward fuzzy upper approximation: $\overline{R}^{<}_{d_i^{\leq}(x)} = \sup_{u \in U} \min\{R^{<}(x,u), d_i^{\leq}(u)\}$

Furthermore, given a decision table $<U, C, D>$, $R^{>}$ and $R^{<}$ are two fuzzy relations generated by $B \subseteq C$. The decision value domain is $D=\{d_1, d_2, \ldots d_p\}$. Assume $d_1 \leq \leq d_2 \leq \ldots \leq d_p$. The fuzzy preference approximation quality (FPAQ) of *D* with respect to *B* is then defined in [7] as

- upward FPAQ: $r_B^{>}(D) = \dfrac{\sum_i \sum_{x \in d_i} \underline{R}^{>}_{d_i^{\geq}(x)}}{\sum_i |d_i^{\geq}|}$

- downward FPAQ $r_B^{\leq}(D) = \dfrac{\sum_i \sum_{x \in d_i} \underline{R}^{\leq}_{d_i^{\leq}(x)}}{\sum_i |d_i^{\leq}|}$
- global FPAQ: $r_B(D) = \dfrac{\sum_i \sum_{x \in d_i} \underline{R}^{\leq}_{d_i^{\leq}(x)} + \sum_i \sum_{x \in d_i} \underline{R}^{>}_{d_i^{\geq}(x)}}{\sum_i |d_i^{\leq}| + \sum_i |d_i^{\geq}|}$

where $|d_i^{\geq}|$ and $|d_i^{\leq}|$ are the numbers of samples with decisions dominating and dominated by d_i, respectively. Detailed descriptions are available in [7]. The fuzzy preference approximation quality reflects the capability of B to D, and thus can be used as a criterion to select features that have better monotonic relationship with decisions.

3 A Feature Fusion Method

As stated in Section 1, in condition monitoring, selecting indicators which monotonically vary with the development of faults is important. Depending on specific objectives, many features from the time-domain, the frequency-domain, and/or the time-frequency domain could be extracted with signal processing technology, which results in a high dimension of the feature space. However, not all the features contribute useful information especially in the sense of monotonicity. The fuzzy preference based rough set model can help to select candidate features which have greater monotonic relationship with the fault development. Moreover, each of the candidate features may contain complementary information on machinery condition. To make the final indicator more robust, feature fusion technology is needed to combine the information of candidate features. PCA, as a popular and widely known feature fusion method, is hence applicable here. PCA transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The first principal component, as it accounts for the largest portion of the variability in the data, contains the most information for fault conditions and is thus can be used as the final single indicator.

To generate a better indicator, a proper number of candidate features should be included. A feature fusion method taking advantage of fuzzy preference based rough set and PCA is proposed and shown in Fig. 1. The procedure is stated as follows:

(1): Extract features that contain information of the machine conditions.
(2): Employ fuzzy preference approximation quality (FPAQ) to evaluate features.
(3): Generate the candidate feature pool based on the values obtained in step (2). The first feature in the candidate feature pool has the highest value, the second one has the second highest value, and so forth. Set m=1;
(4): Import the first m features to the indicator generation module, and apply PCA. The first principle component is used as a single indicator for the machine health condition.
(5): Evaluate the performance of the indicator generated in step (4) using FPAQ. Output the indicator if the performance value doesn't increase any more. Otherwise, set m=m+1, and go to step (4).

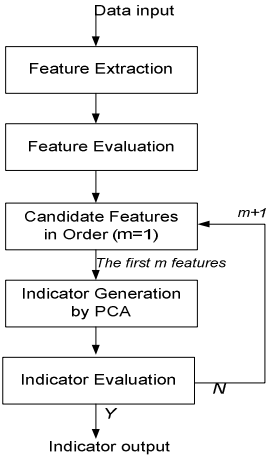


Fig. 1. Flow chart of the proposed feature fusion method

4 Experiments and Results

4.1 Experiment Data

The proposed method is applied to the condition monitoring of an impeller in a centrifugal pump. Impeller vane trailing edge damage, as one of the most prevalent types of impeller damages, is considered here. Two levels of trailing edge damage – slight and severe – were fabricated and tested. Six minutes of vibration data were collected by three tri-axial accelerometer sensors shown in Fig. 2 with a sampling rate of 5kHz.

4.2 Feature Extraction

For centrifugal pumps, pump frequency (1X) and its 2nd harmonic (2X), and vane passing frequency (5X since the impeller has 5 vanes) and its 2nd harmonic (10X) carry useful information on pump conditions [9], and thus are used in this paper. The features are listed in Table 1. Since accelerometer 3 is located on the bearing casing and is relative far from the impeller, to reduce computational burden, in this paper we consider signals measured by accelerometer 1 and accelerometer 2 only. For each vibration signal, 8 features are extracted from the original signal. The two accelerometers output 6 vibration signals which make the number of features equal to 48.

The geometry damage of the impeller will influence the flow field and may cause the impeller outlet recirculation to occur. We have also noticed that the conventional spectral analysis using Fourier Transform (FFT) treats the vibration signal as a real

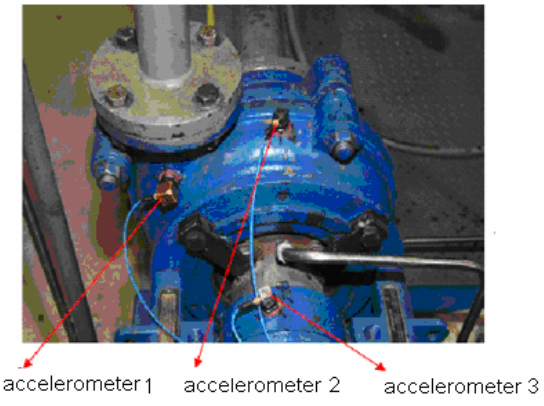


Fig. 2. Locations of the three accelerometers

Table 1. Extracted features

Feature No.	Feature name
1	Amplitude at 1X (1 times the pump rotating frequency)
2	Amplitude at 2X
3	Amplitude at 5X (this is the vane passing frequency)
4	Amplitude at 10X
5	Root mean square (RMS) between (0~1X)
6	RMS between (0~2X)
7	RMS between (0~5X)
8	RMS between (0~10X)

quantity and hence the corresponding frequency spectrum might lose important information like directivity, i.e. forward and backward whirl of vibration motion. As a result, vibration directivity might also be affected. Full spectrum analysis overcomes the limitation of FFT by retaining the relative phase information between two measured orthogonal vibration signals [10]. This attribute makes the full spectrum one of the important diagnostic tools for rotating machine fault detection [11]. By full spectrum, forward whirl frequency component and backward whirl frequency component can be obtained. To enrich the feature pool, the features listed in Table 1 are also extracted from forward whirl frequency components and backward whirl frequency components separately. Each accelerometer has three orthogonal vibration signals (x, y, z), thus full spectrum can be conducted on the 3 signal combinations (i.e. xy plane, yz plane and xz plane). As a result, 6 signal combinations from two accelerometers generate 96 features.

Therefore, totally 144 features are generated of which 48 are from FFT analysis and 96 are from full spectrum analysis.

4.3 Feature Selection and Indicator Generation

Feature selection aims to find candidate features that have monotonic trends with the damage levels. There are two monotonic trends, increasing and decreasing. Features

having better increasing (decreasing) trends with damage degree can be obtained by assigning the damage condition increasing (decreasing) numbers, e.g. 0 for undamaged, 1 (-1) for slight damaged), 2 (-2) for severe damage. The global fuzzy preference approximation quality (global FPAQ) is used to evaluate each feature's performance. The evaluation is conducted for increasing trend and decreasing trend separately, and the maximum evaluation value of increasing trend and decreasing trend is used as the estimation of the contribution of a feature to the fault condition.

Vibration data collected at 2400 revolutions per minute (RPM) is firstly used to illustrate the proposed method. Fig. 3 shows the estimation values for all features. It is obvious that the values are different for different features. The feature with the highest evaluation value (i.e. the first feature in candidate features) is shown in Fig. 4. It can be seen that even with the highest estimation value, its monotonic trend is not satisfactory. There are overlaps between the slight damage and the severe damage.

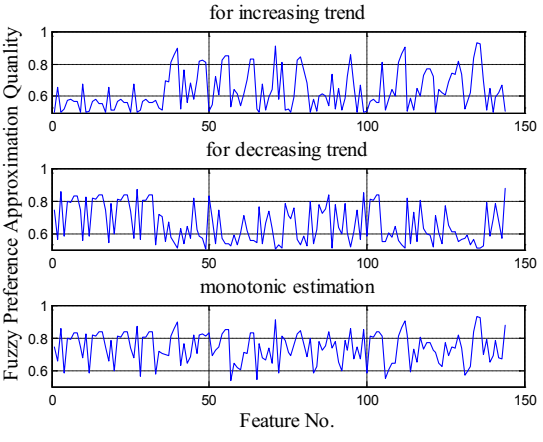


Fig. 3. Feature estimation by global FPAQ

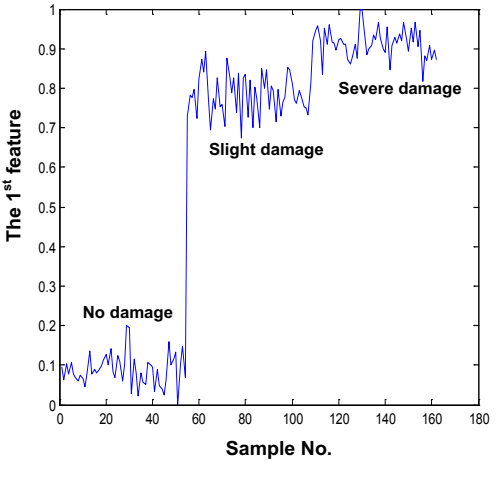


Fig. 4. The value of the 1st feature in the candidate feature pool versus damage levels

Using the proposed method, 25 features were finally selected to generate an indicator. Fig. 5 shows the indicator versus the damage level, in which a clearly monotonic trend is observed. Comparison of Fig. 4 and Fig. 5, it can be shown that PCA effectively combine the information contained in all selected features, and therefore the indicator outperforms individual features.

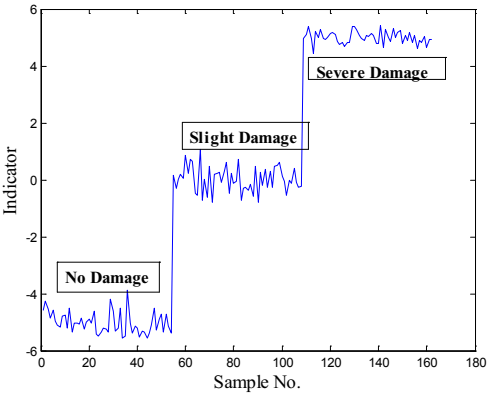


Fig. 5. Indicator Versus damage levels (with proposed method)

4.4 Results and Discussion

To check the contribution of global FPAQ, PCA was applied directly to all the 144 features. The result was shown in Fig. 6, from which it can be seen that the monotonic trend is now lost and the samples from the no damage set and the severe damage set are mixed. This can be explained as follows. As shown in Fig. 3, different features have different global FPAQ values. Those features whose global FPAQ are small give poor or even negative contribution to the indicator generation, which will result in an indicator without a monotonic trend.

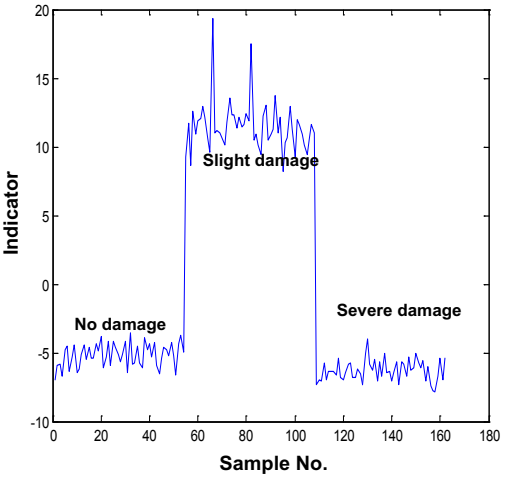


Fig. 6. Indicator generated without global FPAQ

According to the algorithm of PCA, it is also known that the indicator in Fig. 5 is actually a linear weighted sum of the 25 features. The weights can be obtained from the data collected at the 2400 RPM. To test the robustness and generalization ability of this indicator, the same 25 features are collected under 2200 RPM and 2000 RPM. The new values of this indicator are computed by summing up the weighted 25 features. The results are shown in Fig. 7, from which we can observe that the indicator monotonically varies with damage level and can classify different damage levels clearly, especially for the 2000 RPM. This shows the robustness of the proposed method.

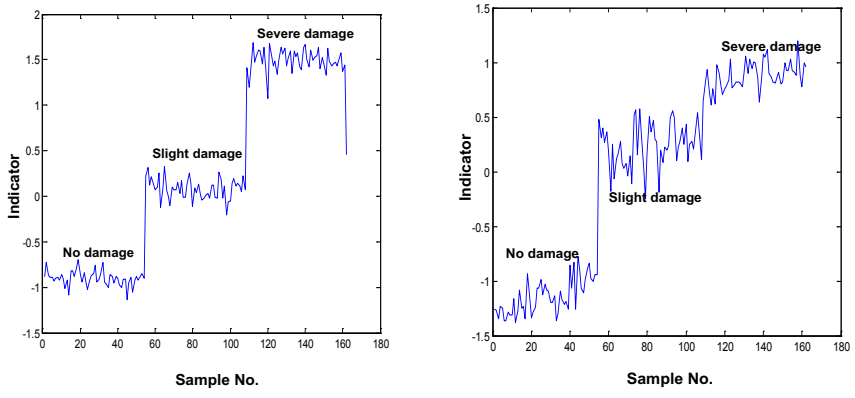


Fig. 7. Indicator Versus damage level (a-2000 RPM, b-2200 RPM)

5 Conclusion

Fuzzy preference based rough set model is useful in condition monitoring. The proposed method utilizes global FPAQ as a criterion to evaluate the performance of features, and uses PCA to combine the information contained in selected features. Experiment results show the effectiveness of this method. It can generate an indicator which monotonically represents the damage levels.

Acknowledgments. This research was supported by Syncrude Canada Ltd. and the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] Yesilyurt, I., Ozturk, H.: Tool condition monitoring in milling using vibration analysis. International Journal of Production Research 45(4), 1013–1028 (2007)

[2] Zhang, B., Georgoulas, G., Orchard, M., Saxena, A., Brown, D., Vachtsevanos, G., Liang, S.: Rolling Element Bearing Feature Extraction and Anomaly Detection Based on Vibration Monitoring. In: 16th Mediterranean Conference on Control and Automation, Ajaccio, France, June 25 -27, pp. 1792–1797 (2008)

[3] Natke, H., Cempel, C.: The symptom observation matrix for monitoring and diagnostics. Journal of Sound and Vibration 248(4), 597–620 (2001)
[4] Hu, Q., Yu, D., Xie, Z.: Information-preserving hybrid data reduction based on fuzzy-rough techniques. Pattern Recognition Letters 27(5), 414–423 (2006)
[5] Shen, Q., Jensen, R.: Selecting informative features with fuzzy-rough sets and its application for complex systems monitoring. Pattern Recognition 37(7), 1351–1363 (2004)
[6] Greco, S., Inuiguchi, M., Slowinsk, R.: Dominance-based rough set approach using possibility and necessity measure. In: Alpigini, J.J., Peters, J.F., Skowron, A., Zhong, N. (eds.) RSCTC 2002. LNCS (LNAI), vol. 2475, pp. 85–92. Springer, Heidelberg (2002)
[7] Hu, Q., Yu, D., Guo, M.: Fuzzy preference based rough sets. Information Sciences 180(10), 2003–2022 (2010)
[8] Turhan-Sayan, G.: Real time electromagnetic target classification using a novel feature extraction technique with PCA-based fusion. IEEE Transactions on Antennas and Propagation 53(2), 766–776 (2005)
[9] Volk, M.W.: Pump Characteristics and Applications, 2nd edn. Taylor & Francis Group, Boca Raton (2005)
[10] Goldman, P., Muszynska, A.: Application of full spectrum to rotating machinery diagnostics. In: Orbit First Quarter, pp. 17–21 (1999)
[11] Patel, T., Darpe, A.: Vibration response of misaligned rotors. Journal of Sound and Vibration 325, 609–628 (2009)

Graph-Based Optimization Method for Information Diffusion and Attack Durability in Networks

Zbigniew Tarapata and Rafał Kasprzyk

Military University of Technology, Cybernetics Faculty,
Gen. S. Kaliskiego Str. 2, 00-908 Warsaw, Poland
{zbigniew.tarapata,rafal.kasprzyk}@wat.edu.pl

Abstract. In this paper we present a graph-based optimization method for information diffusion and attack durability in networks using properties of Complex Networks. We show why and how Complex Networks with *Scale Free* and *Small World* features can help optimize the topology of networks or indicate weak or strong elements of the network. We define some efficiency measures of information diffusion and attack durability in networks. Using these measures we formulate multicriteria optimization problem to choose the best network. We show a practical example of using the method based on an analysis of a few social networks.

Keywords: complex networks, information diffusion, multicriteria graph optimization, networks attack durability.

1 Introduction

Identifying and measuring properties of any network is a first step towards understanding their topology, structure and dynamics. The next step is to develop a mathematical model, which typically takes a form of an algorithm for generating networks with the same statistical properties. Apparently, networks derived from real data (most often spontaneously growing) have a large number of nodes, "*six degree of separation*" characteristic, power law degree distributions, hubs occurring, tendency to form clusters and many other interesting features. These kinds of networks are known as Complex Networks [12]. Two very interesting models capture these features, which have been introduced recently [17]. A *Small World* network [11], [15], [18] is a type of network, in which most nodes are not neighbours of one another, but most of them can be reached from any other with a small number of steps. The *Scale Free* feature [2], [3] pertains to a network, in which most of nodes have relatively small amount of links, but there are some that have a huge amount of neighbours. The *Scale Free* and *Small World* networks, while being fault tolerant, are still very prone to acts of terrorism.

Most of the systems that surround us can be seen as a large scale infrastructure network intended to deliver resources, information or commands to every element of their components. The measures of networks centrality are of particular interest, because they help optimize the network topology from an efficiency and durability point of view. Based on defined centrality measures, we propose a new method to

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 698–709, 2010.

discover the critical elements of networks. It is likely that the identification and protection of the critical elements of a given network should be the first concern in order to reduce the consequence of faults or attacks. On the other hand, the critical elements of hostile networks are the main target to hit in order to disrupt hostile forces and to reduce their capability to optimal decision making.

2 Models and Methods for Complex Networks Analysis

2.1 Definition and Notation for Network Modelling

Formally, a graph is a vector $G=<V,E,P>$ where: V is a set of vertices, E *is* a set of edges, and P is an incidence relationship, i.e. $P \subset V \times E \times V$. The degree k_i of a vertex v_i is the number of edges originating from or ending in vertex v_i. The shortest path d_{ij} from v_i to v_j is the shortest sequence of alternating vertices and edges, starting in vertex v_i and ending in vertex v_j. The length of a path is defined as the number of links in it. Networks very often are represented in practice by a matrix called the adjacency matrix **A**, which in the simplest case is an n x n symmetric matrix, where n is the number of vertices in the network, $n = |V|$. The element of adjacency matrix A_{ij}=1, if there is an edge between vertices i and j, and 0.

In some cases the use of the graph does not provide a complete description of the real-world systems under investigation. For instance, if networks are represented as a simple graph, we only know whether systems are connected (data are exchanged between them), but we cannot model the kind/strength of that connection [16]. For now, however, we will only use the formal graph definition.

2.2 Standard Centrality Measures

Centrality measures address the question "Who (what) is most important or who (what) is the central person (node) in given network?" No single measure of centre is suited for all application.

We considered the five most important centrality measures [5], [9], [14]. Normalization into the range [0, 1] is used here to make the centrality of different vertices comparable, and also independent of the size of the network.

- *Degree centrality*

The degree centrality measure gives the highest score of influence to the vertex with the largest number of direct neighbours. This agrees with the intuitive way to estimate someone's influence from the size of his immediate environment: $k_i = \sum_{j=1}^{n} A_{ij}$. The degree centrality is traditionally defined analogically to the degree of a vertex, normalized with the maximum number of neighbours that this vertex could have. Thus, in a network of n vertices, the degree centrality of vertex v_i, is defined as:

$$center_i^{Degree} = \frac{k_i}{n-1} \tag{1}$$

- *Radius centrality*

If we need to find influential nodes in an area modeled by a network it is quite natural to use the radius centrality measures, which chooses the vertex using the pessimist's criterion. The vertex with the smallest length value of longest of the shortest paths is the most centrally located node [9]. So, if we need to find the most influential node for the most remote nodes it is quite natural and easy to use this measure. The radius centrality of vertex v_i, can be defined as:

$$center_i^{Radius} = \frac{1}{max_{j \in V} d_{ij}} \tag{2}$$

- *Closeness centrality*

This notion of centrality focuses on the idea of communication between different vertices. The vertex, which is 'closer' to all vertices, gets the highest score. In effect, this measure indicates, which one of two vertices needs fewer steps in order to communicate with some other vertex [14]. Because this measure is defined as 'closeness', the inverse of the mean distance from a vertex to all others is used:

$$center_i^{Closness} = \left[\frac{\sum_{j=1}^{n} d_{ij}}{n-1}\right]^{-1} = \frac{n-1}{\sum_{j=1}^{n} d_{ij}} \tag{3}$$

- *Betweenness (load) centrality*

This measure assumes that the greater number of paths, in which a vertex participates, the higher the importance of this vertex is for the network. Informally, betweenness centrality of a vertex can be defined as the percent of shortest paths connecting any two vertices that pass through that vertex [8]. If $p_{lk}(i)$ is the set of all shortest paths between vertices v_l and v_k passing through vertex v_i and p_{lk} is the set of all shortest paths between vertices v_l and v_k then:

$$center_i^{Betweenness} = \frac{2\sum_{l<k} \frac{p_{lk}(i)}{p_{lk}}}{(n-2)(n-1)} \tag{4}$$

This definition of centrality explores the ability of a vertex to be 'irreplaceable' in the communication of two random vertices.

- *Eigenvector centrality*

Where degree centrality gives a simple count of the number of connections that a vertex has, eigenvector centrality acknowledges that not all connections are equal [5]. In general, connections to vertices, which are themselves influential, will grant a vertex more influence than connections to less important vertexes. If we denote the centrality of vertex v_i by e_i, then we can allow for this effect by making e_i proportional to the average of the centralities of the v_i's network neighbors:

$$e_i = \frac{1}{\lambda}\sum_{j=1}^{n} A_{ij} e_j \Rightarrow \vec{e} = \frac{1}{\lambda} A\vec{e} \Rightarrow A\vec{e} = \lambda\,\vec{e} \tag{5}$$

So we have $A\vec{e} - \lambda I\ \vec{e} = 0$ and the λ value can be calculated using $\det(A - \lambda I) = 0$. Hence, we see that $\vec{e}$ is an eigenvector and λ – an eigenvalue of the adjacency matrix. Assuming that we wish the centralities to be non-negative, it can be shown that λ must be the largest eigenvalue[1] of the adjacency matrix and $\vec{e}$ the corresponding eigenvector.

3 Optimization Method for Information Diffusion and Attack Durability

3.1 Efficiency Measures of Information Diffusion

We were interested in the speed at which information originating from the central node diffuses through the rest of the network and then we investigated the speed of information diffusion in the opposite direction (from boundary to centre).

We assume that information is released in a chosen node (centre or boundary) and that it diffuses through the rest of the network in discrete steps. In the first step, the chosen node dispatches information to all of its neighbours, and in each next step each of the nodes that received information in the previous step dispatches it further on to all of its neighbours. It is also assumed that the time to traverse each of the links equals only one step, that there are no information losses, and that all links are of sufficient capacity to diffuse information further without any distortion.

Let $EID(G, k)$ be the function describing the percentages of nodes from G, which received information after the k-th step, $k \in N$. We can define two of the following measures of information diffusion efficiency:

$$k^*(G,x) = \min\{k \in N : EID(G,k) \geq x\} \tag{6}$$

$$G^*(x) \in \mathbf{G} \Rightarrow k^*\left(G^*(x),x\right) = \min_{G \in \mathbf{G}} k^*(G,x) \tag{7}$$

The first measure (6) describes the minimal step number, for which percentages of nodes from G, that received information is equal to x or greater than x. The smaller the value of this measure the better.

The second measure (7) describes such a graph from set $\mathbf{G}$ of graphs, for which $k^*(G^*(x),x)$ is minimal.

3.2 Network Durability Measures

To evaluate how well a G network is connected before and after the removal of a set of nodes we use the global connection efficiency (GCE) [7]. We assume that the connection efficiency between vertex v_i and v_j is inversely proportional to the shortest distance:

$$connection_{ij}^{efficiency}(G) = \frac{1}{d_{ij}} \tag{8}$$

[1] We consider only undirected graphs so adjacency matrix is always symmetric and computing eigenvalues in this way is numerically stable.

When there is no path in graph G between vertex v_i and v_j we have $d_{ij}=+\infty$ and consequently connection efficiency is equal zero.
The global connection efficiency is defined as the average connection efficiency over all pairs of nodes:

$$GCE(G) = \frac{2}{n(n-1)} \sum_{i<j} \frac{1}{d_{ij}} \tag{9}$$

Unlike the average path length, the global connection efficiency is a well-defined quantity as well as in the case of non-connected graphs.

Let $G^{-}(y, rs)$ describe graph G after the removal of $y \in N$ nodes using $rs \in RS$ removal strategy, $RS=\{rd, bc, cc, dc, ec, rc\}$. Elements of RS describe that such a node is removed from the graph, which has the greatest value of the following characteristics (rd – random node): bc - betweenness centrality, cc – closeness centrality, dc – degrees centrality, ec - eigenvector centrality, rc - radius centrality. Network durability measure is represented by the function:

$$GCE_{coef}\left(G, y, rs\right) = \frac{GCE(G)}{GCE(G^{-}(y, rs))} \tag{10}$$

The greater the value of the function (10) the smaller network durability (the greater susceptibility to attacks).

3.3 Multicriteria Approach to Information Diffusion and Attack Durability

Let $SG = \{G_1, G_2, ..., G_M\}$ set of graphs be given. Moreover, we have given $x_0 \in \langle 0,100]$, $y_0 \in N$ and $rs_0 \in RS$. The problem is to find such a graph G^{o} from SG that is the most durable and prone to information diffusion. We define this problem as multicriteria optimization problem (*MDID*) in space SG with relation R_D:

$$MDID = \left(SG, F, R_D\right) \tag{11}$$

where $F : SG \rightarrow N \times N \times R$,

$$F\left(G\right) = \left(k^{*}(G, x_0), k^{*}\left(G^{-}(y_0, rs_0), x_0\right), GCE_{coef}\left(G, y_0, rs_0\right)\right) \tag{12}$$

and

$$R_D = \left\{ \begin{array}{l} (Y,Z) \in SG \times SG: \ \left(k^{*}(Y, x_0) \le k^{*}(Z, x_0)\right) \wedge \\ \qquad \left(k^{*}\left(Y^{-}(y_0, rs_0), x_0\right) \le k^{*}\left(Z^{-}(y_0, rs_0), x_0\right)\right) \wedge \\ \qquad \left(GCE_{coef}\left(Y, y_0, rs_0\right) \le GCE_{coef}\left(Z, y_0, rs_0\right)\right) \end{array} \right\} \tag{13}$$

Let us note that $k^{*}(G, x_0)$ describes efficiency of information diffusion in a static situation (before a network attack) and $k^{*}\left(G^{-}(y_0, rs_0), x_0\right)$ describes efficiency of information diffusion in a dynamic situation (after a network attack).

There are many methods for solving the problem (11). One of the methods, which can be applied, is the trade-off method (one objective is selected by the user and the

other ones are considered as constraints with respect to individual minima). For example we could find such a G^o that:

$$GCE_{coef}\left(G^o, y_0, rs_0\right) = \min\left\{\begin{matrix} GCE_{coef}\left(G, y_0, rs_0\right) : k^*(G, x_0) \le k_0, \\ k^*\left(G^-(y_0, rs_0), x_0\right) \le k_0, G \in \mathbf{G} \end{matrix}\right\} \tag{14}$$

or

$$k^*(G^o, x_0) = \min\left\{\begin{matrix} k^*(G, x_0) : GCE_{coef}\left(G, y_0, rs_0\right) \le GCE_0, \\ k^*\left(G^-(y_0, rs_0), x_0\right) \le k_0, G \in \mathbf{G} \end{matrix}\right\} \tag{15}$$

or

$$k^*(G^{-o}(y_0, rs_0), x_0) = \min\left\{\begin{matrix} k^*\left(G^-(y_0, rs_0), x_0\right) : GCE_{coef}\left(G, y_0, rs_0\right) \le GCE_0, \\ k^*(G, x_0) \le k_0, G \in \mathbf{G} \end{matrix}\right\} \tag{16}$$

where k_0 and GCE_0 are given threshold values.

We can also use other methods for solving *MDID*: hierarchical optimization (the idea is to formulate a sequence of scalar optimization problems with respect to the individual objective functions subject to bounds on previously computed optimal values), method of distance functions in L_p-norm ($p \ge 1$), weighted sum of objectives.

4 An Experimental Analysis of the Complex Network

4.1 Efficiency of Information Diffusion

To compare networks of different structure type we have to chose such networks which have the same general properties: number of nodes equals 62 and the average degree of nodes equals 4.9. We use three networks: Krebs' network *TN* [10], [13] with *Scale Free* and *Small World* feature, random and hierarchical graphs (see Fig.1a).

The dynamic of information diffusion is represented in Fig. 1 (centre node with the highest value of betweenness centrality measure was chosen). Percentages of nodes that received information in specified time steps is contained in Table 1.

The information diffusion through Krebs' networks *TN* is referred to the *Small World* feature, meaning that it does not take many steps to get from one node to another. Together with secrecy of the given network (only 8% of all possible connections in the network really exist), this feature significantly improves operational quickness of terrorist network activities. In one moment the network may activate suddenly, so that an observer is really left with the impression of terrorists "gathering from nowhere and disappearing after action".

Taking into account efficiency measures of information diffusion defined in section 3.1 we obtain (see Table 1): $k^*(G_1, 100) = 4$, $k^*(G_2, 100) = 5$,

$k^*(G_3,100)=4$, hence $G^*(100)\in\{G_1,G_3\}$. In Fig.2 we present the dynamic of information diffusion ($k^*(G,x)$) in these three graphs. From this figure results that information diffuses much faster in the G_1 graph.

In Table 2 we present the fraction of *EID*(*G*, *k*) nodes that received information for each *k*-th time-step after removal of the node with the highest degree of centrality measure in *G* from centre to boundary (periphery). From this table results that the ability to diffuse information in the networks significantly deteriorates in comparison with the state before node removal (compare Table 1 with Table 2). This is especially visible in the hierarchical network.

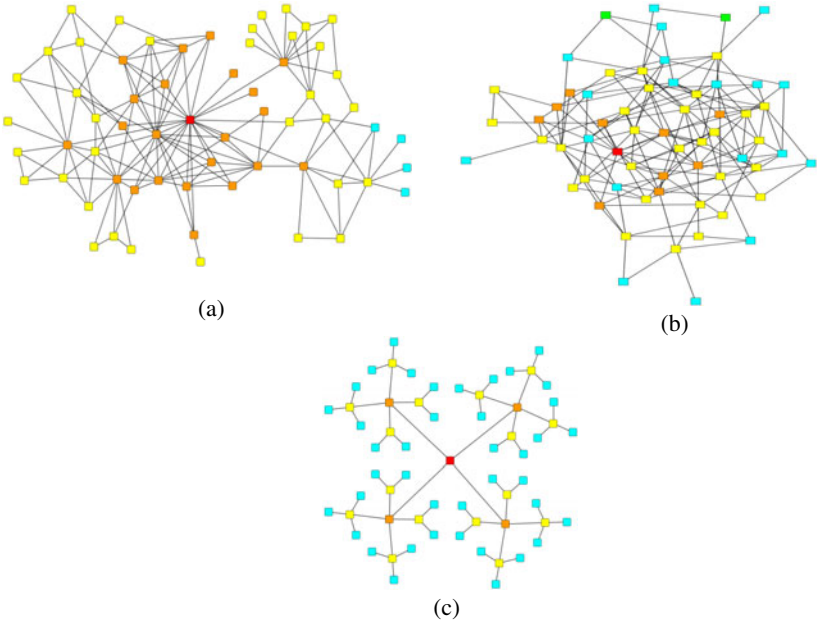


Fig. 1. Information diffusion from the centre to the boundary in: (a) *Scale Free* & *Small World* network *TN* (G_1); (b) random graph (G_2); (c) hierarchical graph (G_3).

Table 1. Fraction of *EID*(*G*, *k*) nodes that received information for each *k*-th time-step. Information diffusion from centre to boundary (periphery)

Type of graph (*G*)	*k*=1	*k*=2	*k*=3	*k*=4	*k*=5
Scale Free & Small World (G_1)	2%	37%	94%	100%	100%
Random (G_2)	2%	18%	66%	97%	100%
Hierarchical (G_3)	2%	8%	34%	100%	100%

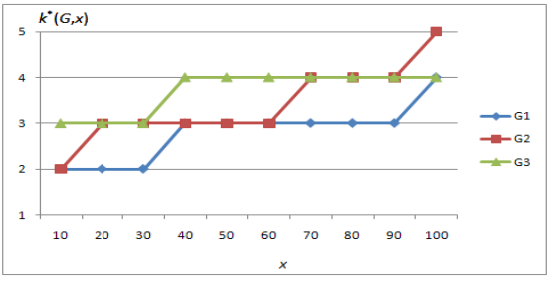


Fig. 2. Graphs of function $k^*(G,x)$ for data from Table 1: (a) *Scale Free* & *Small World* network *TN* (G1≡G_1); (b) random graph (G2≡G_2); (c) hierarchical graph (G3≡G_3)

Table 2. Fraction of *EID*(*G*, *k*) nodes that received information for each *k*-th time-step after node removal with the highest degree centrality measure. Information diffusion from centre to boundary (periphery)

Type of the graph (*G*)	*k*=1	*k*=2	*k*=3	*k*=4	*k*=5	*k*=6	*k*=7
Scale Free & Small World (G_1)	2%	28%	56%	74%	88%	97%	97%
Random (G_2)	2%	18%	59%	95%	100%	100%	100%
Hierarchical (G_3)	2%	8%	26%	26%	26%	26%	26%

4.2 Network Durability

Resistance of the network to any losses of their members is most importance for a network's survival [1], [4], [7]. There are at least three indicators of a network's destruction: (1) the information flow through the network is seriously reduced; (2) the network, as a decision body, can no longer reach consensus; (3) the network, as an organization, loses the ability to effectively perform its task.

Robustness analysis of sample network (Krebs' network *TN,* Fig.1a) with the *Scale Free* feature shows that this network is exceptionally resilient to the loss of a random node. This is not surprising, if we remember that most of the nodes in a *Scale Free* network may have a small degree. Another surprising finding is that *Scale Free* networks are not destroyed even when their central node is removed through redundancy and flexibility construction of this network, which enables quick reconstruction without losses of functionality. To destroy a *Scale Free* network, one must simultaneously remove at least 5% (3/62) of nodes in the central position (see Fig.3).

The average degree of nodes is 4.9. The degree distribution of nodes is particularly interesting. The degree of most nodes is small, while only few nodes have a high degree. It is easy to show, that degrees of nodes have exponentially distribution.

This property characterizes *Scale Free* networks, rising spontaneously, without a particular plan or intervention of architect. Nodes that are members of the network for a longer time are better connected with other nodes, and these nodes are more significant for network functionality and also more "visible" to new members.

The average radius centrality of nodes is 0.23. According to this measure, the node is central, if it is the closest to the farthest node (min-max criterion).

Random attack of Complex Networks is almost useless because the *Scale Free* networks remain connected even after up to 80% of all their nodes (based on

simulation results, see Fig.4) being attacked (destroyed or isolated). However, a clever attack (targeted attack) aimed at central nodes will disintegrate the network rapidly. So, if we know the network topology we can use centrality measures to identify most important nodes and then protect only those with the highest score to assure network functionality.

Fig. 4 shows the percentages of nodes that must be attacked using two attack strategies to reduce *GCE* by a factor of ten (we assumed that it would disintegrate the network as an entirety).

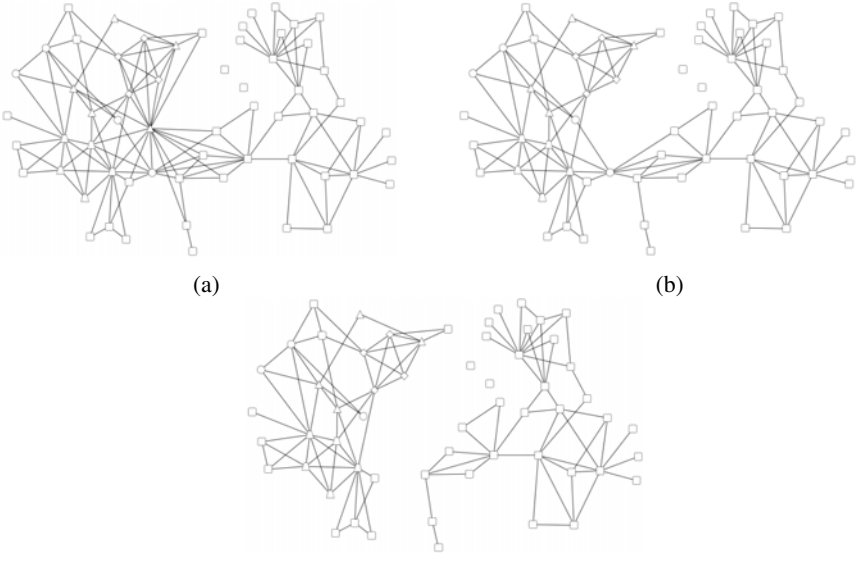


Fig. 3. Krebs' network after removal: (a) the first central node; (b) the second central node; (c) the third central node

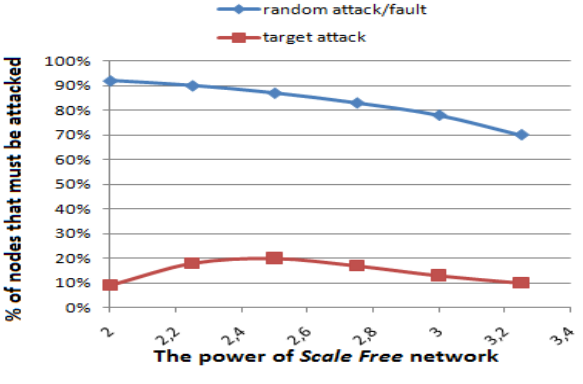


Fig. 4. The effect of two attack strategies for different *Scale Free* networks (by the power of their node degree distribution)

Table 3. The global connection efficiency before ($GCE(G)$) and after ($GCE\left(G^{-}(1,\square)\right)$) removal of a single node using three removal strategies: *rd*, *dc* and *bc*

Type of graph (G)	$GCE(G)$	$GCE\left(G^{-}(1,rd)\right)$	$GCE\left(G^{-}(1,dc)\right)$	$GCE\left(G^{-}(1,bc)\right)$
Scale Free & Small World (G_1)	0.1631	0.1614	0.1273	0.1566
Random (G_2)	0.2176	0.2143	0.1959	0.1966
Hierarchical (G_3)	0.0546	0.0555	0.0481	0.0481

Taking into account the function values in Table 3 and Table 4 we can observe that the worst case is an attack on the node with the highest degree centrality value in the G_1 network. The most attack-resistant network is a random one (G_2) but the information diffusion in this network is rather slower than in G_1.

Table 4. Network durability measure after removal of a single node using three removal strategies: *rd*, *dc* and *bc*

Type of graph (G)	$GCE_{coef}(G,1,rd)$	$GCE_{coef}(G,1,dc)$	$GCE_{coef}(G,1,bc)$
Scale Free & Small World (G_1)	1.0105	1.2812	1.0415
Random (G_2)	1.0154	1.1108	1.1068
Hierarchical (G_3)	0.9838	1.1351	1.1351

4.3 Multicriteria Network Choice

Taking into account *MDID* problem described in section 3.3 we can define and solve the problem of a two-criterion network selection. Let $x_0 = 90$, y_0=1 and $rs_0 = dc$. From Fig.2 we obtain: $k^*(G_1,90) = 3$, $k^*(G_2,90) = k^*(G_3,90) = 4$. From Table 2 we have: $k^*(G_1^-(1,dc),90) = 6$, $k^*(G_2^-(1,dc),90) = 4$, $k^*(G_3^-(1,dc),90) = +\infty$.

From Table 4 we obtain: $GCE_{coef}(G_1,1,dc) = 1.2812$, $GCE_{coef}(G_2,1,dc) = 1.1108$, $GCE_{coef}(G_3,1,dc) = 1.1351$. Let k_0=6, and GCE_0=1.3.

Solving problem (14) we obtain the best network $G^o \equiv G_2$ taking into account all criteria. Solving problem (15) we obtain the best network $G^o \equiv G_1$ taking into account all criteria. Solving problem (16) we obtain the best network $G^o \equiv G_2$ taking into account all criteria.

5 Summary

In this paper we show why and how Complex Networks with the *Scale Free* and *Small World* feature can help optimize the topology of communication networks. The first term – *Scale Free* feature – is a good protection against random attacks (it is hard

to hit a central node). The second term – *Small World* feature – can dramatically affect communication among network nodes. Thus both concepts and underlying theories are highly pertaining to the presenting idea subject and objectives.

Our models and methods of networks analysis have been used in the criminal justice domain to analyze large datasets of criminal groups in order to facilitate crime investigation. However, link analysis still faces many challenging problems, such as information overload, high search complexity, and heavy reliance on domain knowledge. Another problem is to take into account not only the structure of the network, but also a quantitative description (weights) of links [16].

Acknowledgements

This work was partially supported by projects: No. PBZ-MNiSW-DBO-01/I/2007 titled "Advanced methods and techniques for creating situation awareness in network centric warfare" and MNiSW OR00005006 titled "Integration of command and control systems".

References

1. Antkiewicz, R., Kasprzyk, R., Najgebauer, A., Tarapata, Z.: The concept of C4IS topology optimization using complex networks. In: Proceedings of the Military Communications and Information Systems Conference MCC 2009, Prague, Czech Republic, September 29-30 (2009), ISBN 978-80-7231-678-6
2. Barabasi, A.L., Reka, A.: Emergency of Scaling in Random Networks. Science 286, 509–512 (1999)
3. Barabasi, A.L., Reka, A.: Statistical mechanics of complex networks. Review of Modern Physics 74, 47–97 (2002)
4. Bhandari, R.: Survivable networks. Algorithms for divers routing. Kluver Academic Publishers, Dordrecht (1999)
5. Bonacich, P.: Factoring and Weighting Approaches to Status Scores and Clique Identification. Journal of Mathematical Sociology 2, 113–120 (1972)
6. Brandes, U.: A Faster Algorithm for Betweenness Centrality. Journal of Mathematical Sociology 25, 163–177 (2001)
7. Crucitti, P., Latora, V., Marchiori, M., Rapisarda, A.: Error and attack tolerance of complex networks. Physica A 340, 388–394 (2004)
8. Freeman, L.: A set of Measures of Centrality Based on Betweenness. Sociometry 40, 35–41 (1977)
9. Harary, F., Hage, P.: Eccentricity and centrality in networks. Social Networks 17, 57–63 (1995)
10. Krebs, V.: Mapping Networks of Terrorist Cells. Connections 24(3), 43–52 (2002)
11. Newman, M.E.: Models of the small world: A review. J. Stat. Phys. 101, 819–841 (2000)
12. Newman, M.E.: The structure and function of complex networks. SIMA Review 45(2), 167–256 (2003)

13. Penzar, D., Srbljinović, A.: About Modeling of Complex Networks With Applications To Terrorist Group Modelling. Interdisciplinary Description of Complex Systems 3(1), 27–43 (2005)
14. Sabidussi, G.: The Centrality Index of a Graph. Psychometrica 31, 581–603 (1966)
15. Strogatz, S.H.: Exploring complex networks. Nature 410, 268–276 (2001)
16. Tarapata, Z., Kasprzyk, R.: An application of multicriteria weighted graph similarity method to social networks analyzing. In: Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining (ASONAM 2009), Athens, Greece, July 20-22, pp. 366–368. IEEE Computer Society, Los Alamitos (2009)
17. Wang, X., Chen, G.: Complex Networks: Small-world, scale-free and beyond. IEEE Circuits and Systems Magazine 3(1), 6–20 (2003)
18. Watts, D.J., Strogatz, S.H.: Collective dynamics of "small-world" networks. Nature 393, 440–442 (1998)

Paraconsistent and Approximate Semantics for the OWL 2 Web Ontology Language

Linh Anh Nguyen

Institute of Informatics, University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
nguyen@mimuw.edu.pl

Abstract. We introduce a number of paraconsistent semantics, including three-valued and four-valued semantics, for the description logic $\mathcal{SROIQ}$, which is the logical foundation of OWL 2. We then study the relationship between the semantics and paraconsistent reasoning in $\mathcal{SROIQ}$ w.r.t. some of them through a translation into the traditional semantics. We also present a formalization of rough concepts in $\mathcal{SROIQ}$.

1 Introduction

The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies. It is considered one of the fundamental technologies underpinning the Semantic Web, and has attracted both academic and commercial interest. OWL has a formal semantics based on description logics (DLs) [1], which are formalisms concentrated around concepts (classes of individuals) and roles (binary relations between individuals), and aim to specify concepts and concept hierarchies and to reason about them. DLs belong to the most frequently used knowledge representation formalisms and provide a logical basis to a variety of well known paradigms, including frame-based systems, semantic networks and semantic web ontologies and reasoners. The extension OWL 2 of OWL, based on the DL $\mathcal{SROIQ}$ [4], became a W3C recommendation in October 2009.

Some of the main problems of knowledge representation and reasoning involve vagueness, uncertainty, and/or inconsistency. There are a number of approaches for dealing with vagueness and/or uncertainty, for example, by using fuzzy logic, rough set theory, or probabilistic logic. See [5] for references to some works on extensions of DLs using these approaches. A way to deal with inconsistency is to follow the area of paraconsistent reasoning. There is a rich literature on paraconsistent logics (see, e.g., [2,3] and references there).

Rough set theory was introduced by Pawlak in 1982 [13,14] as a new mathematical approach to vagueness. It has many interesting applications and has been studied and extended by a lot of researchers (see, e.g., [17,16,15]). In rough set theory, given a similarity relation on a universe, a subset of the universe is described by a pair of subsets of the universe called the lower and upper approximations. In [18,6] Schlobach et al. showed how to extend DLs with rough concepts. In [5] Jiang et al. gave some details about the rough version of the DL $\mathcal{ALC}$. In general, a traditional DL can be used to

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 710–720, 2010.

express and reason about rough concepts if similarity relations are used as roles and the properties of the similarity relations are expressible and used as axioms of the logic.

A number of researchers have extended DLs with paraconsistent semantics and paraconsistent reasoning methods [9,19,12,8,7,21,11]. The work [12] studies a constructive version of the basic DL $\mathcal{ALC}$. The remaining works except [11] are based on the well-known Belnap's four-valued logic. Truth values in this logic represent truth (t), falsity (f), the lack of knowledge (u) and inconsistency (i). However, there are serious problems with using Belnap's logic for Semantic Web (see [20,11]). In [11] together with Szałas we gave a three-valued paraconsistent semantics for the DL $\mathcal{SHIQ}$, which is related to the DL $\mathcal{SHOIN}$ used for OWL 1.1.

Both rough concepts and paraconsistent reasoning are related to approximation. Rough concepts deal with concept approximation, while paraconsistent reasoning is a kind of approximate reasoning. We can combine them to deal with both vagueness and inconsistency. In this paper, we study rough concepts and paraconsistent reasoning in the DL $\mathcal{SROIQ}$. As rough concepts can be expressed in $\mathcal{SROIQ}$ in the usual way, we just briefly formalize them. We concentrate on defining a number of different paraconsistent semantics for $\mathcal{SROIQ}$, studying the relationship between them, and paraconsistent reasoning in $\mathcal{SROIQ}$ w.r.t. some of such semantics through a translation into the traditional semantics. Our paraconsistent semantics for $\mathcal{SROIQ}$ are characterized by four parameters for:

- using two-, three-, or four-valued semantics for concept names
- using two-, three-, or four-valued semantics for role names
- interpreting concepts of the form $\forall R.C$ or $\exists R.C$ (two ways)
- using weak, moderate, or strong semantics for terminological axioms.

Note that, with respect to DLs, three-valued semantics has been studied earlier only for $\mathcal{SHIQ}$ [11]. Also note that, studying four-valued semantics for DLs, Ma and Hitzler [7] did not consider all features of $\mathcal{SROIQ}$. For example, they did not consider concepts of the form $\exists R.\mathtt{Self}$ and individual assertions of the form $\neg S(a,b)$.

Due to the lack of space, examples and proofs of our results are presented only in the long version [10] of the current paper.

2 The Description Logic $\mathcal{SROIQ}$

In this section we recall notations and semantics of the DL $\mathcal{SROIQ}$ [4]. Assume that our language uses a finite set $\mathbf{C}$ of *concept names*, a subset $\mathbf{N} \subseteq \mathbf{C}$ of *nominals*, a finite set $\mathbf{R}$ of role names including the universal role U, and a finite set $\mathbf{I}$ of individual names. Let $\mathbf{R}^- \stackrel{\text{def}}{=} \{r^- \mid r \in \mathbf{R} \setminus \{U\}\}$ be the set of *inverse roles*. A *role* is any member of $\mathbf{R} \cup \mathbf{R}^-$. We use letters like R and S for roles.

An *interpretation* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ consists of a non-empty set $\Delta^{\mathcal{I}}$, called the *domain* of $\mathcal{I}$, and a function $\cdot^{\mathcal{I}}$, called the *interpretation function* of $\mathcal{I}$, which maps every concept name A to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, where $A^{\mathcal{I}}$ is a singleton set if $A \in \mathbf{N}$, and maps every role name r to a binary relation $r^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$, with $U^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and maps every individual name a to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. Inverse roles are interpreted as usual, i.e., for $r \in \mathbf{R}$, we define $(r^-)^{\mathcal{I}} \stackrel{\text{def}}{=} (r^{\mathcal{I}})^{-1} = \{\langle x, y\rangle \mid \langle y, x\rangle \in r^{\mathcal{I}}\}$.

A *role inclusion axiom* is an expression of the form $R_1 \circ \ldots \circ R_k \sqsubseteq S$. A *role assertion* is an expression of the form $\mathtt{Ref}(R)$, $\mathtt{Irr}(R)$, $\mathtt{Sym}(R)$, $\mathtt{Tra}(R)$, or $\mathtt{Dis}(R,S)$, where $R, S \neq U$. Given an interpretation $\mathcal{I}$, define that $\mathcal{I} \models R_1 \circ \ldots \circ R_k \sqsubseteq S$ if $R_1^{\mathcal{I}} \circ \ldots \circ R_k^{\mathcal{I}} \subseteq S^{\mathcal{I}}$, where $\circ$ stands for composition, and that: $\mathcal{I} \models \mathtt{Ref}(R)$ if $R^{\mathcal{I}}$ is reflexive; $\mathcal{I} \models \mathtt{Irr}(R)$ if $R^{\mathcal{I}}$ is irreflexive; $\mathcal{I} \models \mathtt{Sym}(R)$ if $R^{\mathcal{I}}$ is symmetric; $\mathcal{I} \models \mathtt{Tra}(R)$ if $R^{\mathcal{I}}$ is transitive; $\mathcal{I} \models \mathtt{Dis}(R,S)$ if $R^{\mathcal{I}}$ and $S^{\mathcal{I}}$ are disjoint. By a *role axiom* we mean either a role inclusion axiom or a role assertion. We say that a role axiom φ is *valid* in $\mathcal{I}$ and $\mathcal{I}$ *validates* φ if $\mathcal{I} \models \varphi$.

An *RBox* is a set $\mathcal{R} = \mathcal{R}_h \cup \mathcal{R}_a$, where $\mathcal{R}_h$ is a finite set of role inclusion axioms and $\mathcal{R}_a$ is a finite set of role assertions. It is required that $\mathcal{R}_h$ is *regular* and $\mathcal{R}_a$ is *simple*. In particular, $\mathcal{R}_a$ is simple if all roles R, S appearing in role assertions of the form $\mathtt{Irr}(R)$ or $\mathtt{Dis}(R,S)$ are *simple roles* w.r.t. $\mathcal{R}_h$. These notions (of regularity and simplicity) will not be exploited in this paper and we refer the reader to [4] for their definitions. An interpretation $\mathcal{I}$ is a *model* of an RBox $\mathcal{R}$, denoted by $\mathcal{I} \models \mathcal{R}$, if it validates all role axioms of $\mathcal{R}$.

The set of *concepts* is the smallest set such that: all concept names (including nominals) and $\top$, $\bot$ are concepts; if C, D are concepts, R is a role, S is a simple role, and n is a non-negative integer, then $\neg C$, $C \sqcap D$, $C \sqcup D$, $\forall R.C$, $\exists R.C$, $\exists S.\mathtt{Self}$, $\geq nS.C$, and $\leq nS.C$ are also concepts. We use letters like A, B to denote concept names, and letters like C, D to denote concepts.

Given an interpretation $\mathcal{I}$, the interpretation function $\cdot^{\mathcal{I}}$ is extended to complex concepts as follows, where $\#\Gamma$ stands for the number of elements in the set Γ :

$$\top^{\mathcal{I}} \stackrel{\text{def}}{=} \Delta^{\mathcal{I}} \qquad \bot^{\mathcal{I}} \stackrel{\text{def}}{=} \emptyset \qquad (\neg C)^{\mathcal{I}} \stackrel{\text{def}}{=} \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$$
$$(C \sqcap D)^{\mathcal{I}} \stackrel{\text{def}}{=} C^{\mathcal{I}} \cap D^{\mathcal{I}} \qquad (C \sqcup D)^{\mathcal{I}} \stackrel{\text{def}}{=} C^{\mathcal{I}} \cup D^{\mathcal{I}}$$
$$(\forall R.C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \in \Delta^{\mathcal{I}} \mid \forall y[\langle x,y\rangle \in R^{\mathcal{I}} \text{ implies } y \in C^{\mathcal{I}}]\}$$
$$(\exists R.C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \in \Delta^{\mathcal{I}} \mid \exists y[\langle x,y\rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}]\}$$
$$(\exists S.\mathtt{Self})^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \in \Delta^{\mathcal{I}} \mid \langle x,x\rangle \in S^{\mathcal{I}}\}$$
$$(\geq n\, S.C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid \langle x,y\rangle \in S^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \geq n\}$$
$$(\leq n\, S.C)^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid \langle x,y\rangle \in S^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \leq n\}.$$

A *terminological axiom*, also called a *general concept inclusion* (GCI), is an expression of the form $C \sqsubseteq D$. A *TBox* is a finite set of terminological axioms. An interpretation $\mathcal{I}$ validates an axiom $C \sqsubseteq D$, denoted by $\mathcal{I} \models C \sqsubseteq D$, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. We say that $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$, denoted by $\mathcal{I} \models \mathcal{T}$, if it validates all axioms of $\mathcal{T}$.

We use letters like a and b to denote individual names. An *individual assertion* is an expression of the form $a \doteq b$, $C(a)$, $R(a,b)$, or $\neg S(a,b)$, where S is a simple role and $R, S \neq U$. Given an interpretation $\mathcal{I}$, define that: $\mathcal{I} \models a \doteq b$ if $a^{\mathcal{I}} \neq b^{\mathcal{I}}$; $\mathcal{I} \models C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$; $\mathcal{I} \models R(a,b)$ if $\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in R^{\mathcal{I}}$; and $\mathcal{I} \models \neg S(a,b)$ if $\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \notin S^{\mathcal{I}}$. We say that $\mathcal{I}$ *satisfies* an individual assertion φ if $\mathcal{I} \models \varphi$. An *ABox* is a finite set of individual assertions. An interpretation $\mathcal{I}$ is a *model* of an ABox $\mathcal{A}$, denoted by $\mathcal{I} \models \mathcal{A}$, if it satisfies all assertions of $\mathcal{A}$.

A *knowledge base* is a tuple $\langle \mathcal{R}, \mathcal{T}, \mathcal{A}\rangle$, where $\mathcal{R}$ is an RBox, $\mathcal{T}$ is a TBox, and $\mathcal{A}$ is an ABox. An interpretation $\mathcal{I}$ is a *model* of a knowledge base $\langle \mathcal{R}, \mathcal{T}, \mathcal{A}\rangle$ if it is a model of all $\mathcal{R}$, $\mathcal{T}$, and $\mathcal{A}$. A knowledge base is *satisfiable* if it has a model.

A (*conjunctive*) *query* is an expression of the form $\varphi_1 \wedge \ldots \wedge \varphi_k$, where each φ_i is an individual assertion. An interpretation $\mathcal{I}$ satisfies a query $\varphi = \varphi_1 \wedge \ldots \wedge \varphi_k$, denoted by $\mathcal{I} \models \varphi$, if $\mathcal{I} \models \varphi_i$ for all $1 \leq i \leq k$. A query φ is a *logical consequence* of $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$, denoted by $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle \models \varphi$, if every model of $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ satisfies φ.

3 Rough Concepts in Description Logic

Let $\mathcal{I}$ be an interpretation and R be a role standing for a similarity predicate. For $x \in \Delta^{\mathcal{I}}$, the *neighborhood of* x *w.r.t.* R is the set of elements similar to x specified by $n_R(x) \stackrel{\text{def}}{=} \{y \in \Delta^{\mathcal{I}} \mid \langle x, y \rangle \in R^{\mathcal{I}}\}$. The *lower* and *upper approximations of a concept* C *w.r.t.* R, denoted respectively by $\underline{C}_R$ and $\overline{C}_R$, are interpreted in $\mathcal{I}$ as follows:

$$(\underline{C}_R)^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \in \Delta^{\mathcal{I}} \mid n_R(x) \subseteq C^{\mathcal{I}}\} \quad (\overline{C}_R)^{\mathcal{I}} \stackrel{\text{def}}{=} \{x \in \Delta^{\mathcal{I}} \mid n_R(x) \cap C^{\mathcal{I}} \neq \emptyset\}$$

Intuitively, if the similarity predicate R reflects the perception ability of an agent then

- $x \in (\underline{C}_R)^{\mathcal{I}}$ means that all objects indiscernible from x are in $C^{\mathcal{I}}$
- $x \in (\overline{C}_R)^{\mathcal{I}}$ means that there are objects indiscernible from x which are in $C^{\mathcal{I}}$.

The pair $\langle \underline{C}_R, \overline{C}_R \rangle$ is usually called the *rough concept* of C w.r.t. the similarity predicate R. The following proposition is well known [18,5].

Proposition 3.1. *Let $\mathcal{I}$ be an interpretation, C be a concept, and R be a role. Then $(\underline{C}_R)^{\mathcal{I}} = (\forall R.C)^{\mathcal{I}}$ and $(\overline{C}_R)^{\mathcal{I}} = (\exists R.C)^{\mathcal{I}}$. That is, $\forall R.C$ and $\exists R.C$ are the lower and upper approximations of C w.r.t. R, respectively.* ◁

One can adopt different restrictions on a similarity predicate R. It is expected that the lower approximation is a subset of the upper approximation. That is, for every interpretation $\mathcal{I}$ and every concept C, we should have that $(\underline{C}_R)^{\mathcal{I}} \subseteq (\overline{C}_R)^{\mathcal{I}}$, or equivalently, $(\forall R.C)^{\mathcal{I}} \subseteq (\exists R.C)^{\mathcal{I}}$. The latter condition corresponds to seriality of $R^{\mathcal{I}}$ (i.e. $\forall x \in \Delta^{\mathcal{I}}\, \exists y \in \Delta^{\mathcal{I}}\, R^{\mathcal{I}}(x, y)$), which can be formalized by the global assumption $\exists R.\top$. Thus, we have the following proposition, which is clear from the view of the corresponding theory of modal logics.

Proposition 3.2. *Let $\mathcal{I}$ be an interpretation. Then $(\underline{C}_R)^{\mathcal{I}} \subseteq (\overline{C}_R)^{\mathcal{I}}$ holds for every concept C iff $\mathcal{I}$ validates the terminological axiom $\top \sqsubseteq \exists R.\top$.* ◁

In most applications, one can assume that similarity relations are reflexive and symmetric. Reflexivity of a similarity predicate R is expressed in $\mathcal{SROIQ}$ by the role assertion $\texttt{Ref}(R)$. Symmetry of a similarity predicate R can be expressed in $\mathcal{SROIQ}$ by the role assertion $\texttt{Sym}(R)$ or the role inclusion axiom $R^- \sqsubseteq R$. Transitivity is not always assumed for similarity relations. If one decides to adopt it for a similarity predicate R, then it can be expressed in $\mathcal{SROIQ}$ by the role assertion $\texttt{Tra}(R)$ or the role inclusion axiom $R \circ R \sqsubseteq R$. In particular, in $\mathcal{SROIQ}$, to express that a similarity predicate R stands for an equivalence relation we can use the three role assertions $\texttt{Ref}(R)$, $\texttt{Sym}(R)$, and $\texttt{Tra}(R)$. See [10] for an example illustrating rough concepts in DLs.

4 Paraconsistent Semantics for $\mathcal{SROIQ}$

Recall that, using the traditional semantics, every query is a logical consequence of an inconsistent knowledge base. A knowledge base may be inconsistent, for example, when it contains both individual assertions $A(a)$ and $\neg A(a)$ for some $A \in \mathbf{C}$ and $a \in \mathbf{I}$. Paraconsistent reasoning is inconsistency-tolerant and aims to derive (only) meaningful logical consequences even when the knowledge base is inconsistent. Following the recommendation of W3C for OWL, we use the traditional syntax of DLs and only change its semantics to cover paraconsistency. The general approach is to define a semantics $\mathfrak{s}$ such that, given a knowledge base KB, the set $Cons_{\mathfrak{s}}(KB)$ of logical consequences of KB w.r.t. semantics $\mathfrak{s}$ is a subset of the set $Cons(KB)$ of logical consequences of KB w.r.t. the traditional semantics, with the property that $Cons_{\mathfrak{s}}(KB)$ contains mainly only "meaningful" logical consequences of KB and $Cons_{\mathfrak{s}}(KB)$ approximates $Cons(KB)$ as much as possible.

In this paper, we introduce a number of paraconsistent semantics for the DL $\mathcal{SROIQ}$. Each of them, let's say $\mathfrak{s}$, is characterized by four parameters, denoted by $\mathfrak{s}_{\mathtt{C}}$, $\mathfrak{s}_{\mathtt{R}}$, $\mathfrak{s}_{\forall\exists}$, $\mathfrak{s}_{\mathtt{GCI}}$, with the following intuitive meanings:

- $\mathfrak{s}_{\mathtt{C}}$ specifies the number of possible truth values (2, 3, or 4) of assertions of the form $x \in A^{\mathcal{I}}$, where A is a concept name not being a nominal and $\mathcal{I}$ is an interpretation. In the case $\mathfrak{s}_{\mathtt{C}} = 2$, the truth values are $\mathfrak{t}$ (true) and $\mathfrak{f}$ (false). In the case $\mathfrak{s}_{\mathtt{C}} = 3$, the third truth value is $\mathfrak{i}$ (inconsistent). In the case $\mathfrak{s}_{\mathtt{C}} = 4$, the additional truth value is $\mathfrak{u}$ (unknown). When $\mathfrak{s}_{\mathtt{C}} = 3$, one can identify inconsistency with the lack of knowledge, and the third value $\mathfrak{i}$ can be read either as inconsistent or as unknown.
- $\mathfrak{s}_{\mathtt{R}}$ specifies the number of possible truth values (2, 3, or 4) of assertions of the form $\langle x, y\rangle \in r^{\mathcal{I}}$, where r is a role name different from the universal role U and $\mathcal{I}$ is an interpretation. The truth values are as in the case of $\mathfrak{s}_{\mathtt{C}}$.
- $\mathfrak{s}_{\forall\exists}$ specifies one of the two semantics studied by Straccia [19] for concepts of the form $\forall R.C$ or $\exists R.C$, which are denoted in this paper by $+$ and $+-$.
- $\mathfrak{s}_{\mathtt{GCI}}$ specifies one of the three semantics w (weak), m (moderate), s (strong) for general concept inclusions.

We identify $\mathfrak{s}$ with the tuple $\langle \mathfrak{s}_{\mathtt{C}}, \mathfrak{s}_{\mathtt{R}}, \mathfrak{s}_{\forall\exists}, \mathfrak{s}_{\mathtt{GCI}}\rangle$. The set $\mathfrak{S}$ of considered paraconsistent semantics is thus $\{2,3,4\} \times \{2,3,4\} \times \{+,+-\} \times \{w,m,s\}$.

For $\mathfrak{s} \in \mathfrak{S}$, an *$\mathfrak{s}$-interpretation* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}\rangle$ is similar to a traditional interpretation except that the interpretation function maps every concept name A to a pair $A^{\mathcal{I}} = \langle A^{\mathcal{I}}_+, A^{\mathcal{I}}_-\rangle$ of subsets of $\Delta^{\mathcal{I}}$ and maps every role name r to a pair $r^{\mathcal{I}} = \langle r^{\mathcal{I}}_+, r^{\mathcal{I}}_-\rangle$ of binary relations on $\Delta^{\mathcal{I}}$ such that:

- if $\mathfrak{s}_{\mathtt{C}} = 2$ then $A^{\mathcal{I}}_+ = \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}_-$
- if $\mathfrak{s}_{\mathtt{C}} = 3$ then $A^{\mathcal{I}}_+ \cup A^{\mathcal{I}}_- = \Delta^{\mathcal{I}}$
- if $\mathfrak{s}_{\mathtt{R}} = 2$ then $r^{\mathcal{I}}_+ = (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus r^{\mathcal{I}}_-$
- if $\mathfrak{s}_{\mathtt{R}} = 3$ then $r^{\mathcal{I}}_+ \cup r^{\mathcal{I}}_- = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
- if A is a nominal then $A^{\mathcal{I}}_+$ is a singleton set and $A^{\mathcal{I}}_- = \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}_+$
- $U^{\mathcal{I}}_+ = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ and $U^{\mathcal{I}}_- = \emptyset$.

The intuition behind $A^{\mathcal{I}} = \langle A^{\mathcal{I}}_+, A^{\mathcal{I}}_-\rangle$ is that $A^{\mathcal{I}}_+$ gathers positive evidence about A, while $A^{\mathcal{I}}_-$ gathers negative evidence about A. Thus, $A^{\mathcal{I}}$ can be treated as the function

from $\Delta^{\mathcal{I}}$ to $\{\mathtt{t},\mathtt{f},\mathtt{i},\mathtt{u}\}$ such that $A^{\mathcal{I}}(x)$ is: $\mathtt{t}$ if $x \in A^{\mathcal{I}}_+$ and $x \notin A^{\mathcal{I}}_-$; $\mathtt{f}$ if $x \in A^{\mathcal{I}}_-$ and $x \notin A^{\mathcal{I}}_+$; $\mathtt{i}$ if $x \in A^{\mathcal{I}}_+$ and $x \in A^{\mathcal{I}}_-$; $\mathtt{u}$ if $x \notin A^{\mathcal{I}}_+$ and $x \notin A^{\mathcal{I}}_-$. Informally, $A^{\mathcal{I}}(x)$ can be thought of as the truth value of $x \in A^{\mathcal{I}}$. Note that $A^{\mathcal{I}}(x) \in \{\mathtt{t},\mathtt{f}\}$ if $\mathfrak{s}_{\mathtt{C}} = 2$ or A is a nominal, and $A^{\mathcal{I}}(x) \in \{\mathtt{t},\mathtt{f},\mathtt{i}\}$ if $\mathfrak{s}_{\mathtt{C}} = 3$. The intuition behind $r^{\mathcal{I}} = \langle r^{\mathcal{I}}_+, r^{\mathcal{I}}_-\rangle$ is similar, and under which $r^{\mathcal{I}}(x,y) \in \{\mathtt{t},\mathtt{f}\}$ if $\mathfrak{s}_{\mathtt{R}} = 2$ or $r = U$, and $r^{\mathcal{I}}(x,y) \in \{\mathtt{t},\mathtt{f},\mathtt{i}\}$ if $\mathfrak{s}_{\mathtt{R}} = 3$.

The interpretation function $\cdot^{\mathcal{I}}$ maps an inverse role R to a pair $R^{\mathcal{I}} = \langle R^{\mathcal{I}}_+, R^{\mathcal{I}}_-\rangle$ defined by $(r^-)^{\mathcal{I}} \stackrel{\text{def}}{=} \langle (r^{\mathcal{I}}_+)^{-1}, (r^{\mathcal{I}}_-)^{-1}\rangle$. It maps a complex concept C to a pair $C^{\mathcal{I}} = \langle C^{\mathcal{I}}_+, C^{\mathcal{I}}_-\rangle$ of subsets of $\Delta^{\mathcal{I}}$ defined as follows:

$$\top^{\mathcal{I}} \stackrel{\text{def}}{=} \langle \Delta^{\mathcal{I}}, \emptyset\rangle \qquad \bot^{\mathcal{I}} \stackrel{\text{def}}{=} \langle \emptyset, \Delta^{\mathcal{I}}\rangle \qquad (\neg C)^{\mathcal{I}} \stackrel{\text{def}}{=} \langle C^{\mathcal{I}}_-, C^{\mathcal{I}}_+\rangle$$

$$(C \sqcap D)^{\mathcal{I}} \stackrel{\text{def}}{=} \langle C^{\mathcal{I}}_+ \cap D^{\mathcal{I}}_+, C^{\mathcal{I}}_- \cup D^{\mathcal{I}}_-\rangle \quad (C \sqcup D)^{\mathcal{I}} \stackrel{\text{def}}{=} \langle C^{\mathcal{I}}_+ \cup D^{\mathcal{I}}_+, C^{\mathcal{I}}_- \cap D^{\mathcal{I}}_-\rangle$$

$$(\exists R.\mathtt{Self})^{\mathcal{I}} \stackrel{\text{def}}{=} \big\langle \{x \in \Delta^{\mathcal{I}} \mid \langle x,x\rangle \in R^{\mathcal{I}}_+\}, \{x \in \Delta^{\mathcal{I}} \mid \langle x,x\rangle \in R^{\mathcal{I}}_-\}\big\rangle$$

$$(\geq n\, R.C)^{\mathcal{I}} \stackrel{\text{def}}{=} \big\langle \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid \langle x,y\rangle \in R^{\mathcal{I}}_+ \text{ and } y \in C^{\mathcal{I}}_+\} \geq n\},\ \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid \langle x,y\rangle \in R^{\mathcal{I}}_+ \text{ and } y \notin C^{\mathcal{I}}_-\} < n\}\big\rangle$$

$$(\leq n\, R.C)^{\mathcal{I}} \stackrel{\text{def}}{=} \big\langle \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid \langle x,y\rangle \in R^{\mathcal{I}}_+ \text{ and } y \notin C^{\mathcal{I}}_-\} \leq n\},\ \{x \in \Delta^{\mathcal{I}} \mid \#\{y \mid \langle x,y\rangle \in R^{\mathcal{I}}_+ \text{ and } y \in C^{\mathcal{I}}_+\} > n\}\big\rangle;$$

if $\mathfrak{s}_{\forall\exists} = +$ then

$$(\forall R.C)^{\mathcal{I}} \stackrel{\text{def}}{=} \big\langle \{x \in \Delta^{\mathcal{I}} \mid \forall y(\langle x,y\rangle \in R^{\mathcal{I}}_+ \text{ implies } y \in C^{\mathcal{I}}_+)\},\ \{x \in \Delta^{\mathcal{I}} \mid \exists y(\langle x,y\rangle \in R^{\mathcal{I}}_+ \text{ and } y \in C^{\mathcal{I}}_-)\}\big\rangle$$

$$(\exists R.C)^{\mathcal{I}} \stackrel{\text{def}}{=} \big\langle \{x \in \Delta^{\mathcal{I}} \mid \exists y(\langle x,y\rangle \in R^{\mathcal{I}}_+ \text{ and } y \in C^{\mathcal{I}}_+)\},\ \{x \in \Delta^{\mathcal{I}} \mid \forall y(\langle x,y\rangle \in R^{\mathcal{I}}_+ \text{ implies } y \in C^{\mathcal{I}}_-)\}\big\rangle;$$

if $\mathfrak{s}_{\forall\exists} = +-$ then

$$(\forall R.C)^{\mathcal{I}} \stackrel{\text{def}}{=} \big\langle \{x \in \Delta^{\mathcal{I}} \mid \forall y(\langle x,y\rangle \in R^{\mathcal{I}}_- \text{ or } y \in C^{\mathcal{I}}_+)\},\ \{x \in \Delta^{\mathcal{I}} \mid \exists y(\langle x,y\rangle \in R^{\mathcal{I}}_+ \text{ and } y \in C^{\mathcal{I}}_-)\}\big\rangle$$

$$(\exists R.C)^{\mathcal{I}} \stackrel{\text{def}}{=} \big\langle \{x \in \Delta^{\mathcal{I}} \mid \exists y(\langle x,y\rangle \in R^{\mathcal{I}}_+ \text{ and } y \in C^{\mathcal{I}}_+)\},\ \{x \in \Delta^{\mathcal{I}} \mid \forall y(\langle x,y\rangle \in R^{\mathcal{I}}_- \text{ or } y \in C^{\mathcal{I}}_-)\}\big\rangle.$$

Note that $C^{\mathcal{I}}$ is computed in the standard way [8,7,21,11] for the case C is of the form $\top$, $\bot$, $\neg D$, $D \sqcap D'$, $D \sqcup D'$, $\geq n\, R.D$ or $\leq n\, R.D$. When $\mathfrak{s}_{\forall\exists} = +$, $(\forall R.C)^{\mathcal{I}}$ and $(\exists R.C)^{\mathcal{I}}$ are computed as in [8,7,21,11] and as using semantics A of [19]. When $\mathfrak{s}_{\forall\exists} = +-$, $(\forall R.C)^{\mathcal{I}}$ and $(\exists R.C)^{\mathcal{I}}$ are computed as using semantics B of [19]. De Morgans laws hold for our constructors w.r.t. any semantics from $\mathfrak{S}$ (see [10]).

The following proposition means that: if $\mathfrak{s}_{\mathtt{C}} \in \{2,3\}$ and $\mathfrak{s}_{\mathtt{R}} \in \{2,3\}$ then $\mathfrak{s}$ is a three-valued semantics; if $\mathfrak{s}_{\mathtt{C}} = 2$ and $\mathfrak{s}_{\mathtt{R}} = 2$ then $\mathfrak{s}$ is a two-valued semantics.

Proposition 4.1. *Let $\mathfrak{s} \in \mathfrak{S}$ be a semantics such that $\mathfrak{s}_{\mathtt{C}} \in \{2,3\}$ and $\mathfrak{s}_{\mathtt{R}} \in \{2,3\}$. Let $\mathcal{I}$ be an $\mathfrak{s}$-interpretation, C be a concept, and R be a role. Then $C^{\mathcal{I}}_+ \cup C^{\mathcal{I}}_- = \Delta^{\mathcal{I}}$ and $R^{\mathcal{I}}_+ \cup R^{\mathcal{I}}_- = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Furthermore, if $\mathfrak{s}_{\mathtt{C}} = 2$ and $\mathfrak{s}_{\mathtt{R}} = 2$ then $C^{\mathcal{I}}_+ = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}_-$ and $R^{\mathcal{I}}_+ = (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus R^{\mathcal{I}}_-$.* ◁

Let $\mathfrak{s} \in \mathfrak{S}$, $\mathcal{I}$ be an $\mathfrak{s}$-interpretation and $\langle \mathcal{R}, \mathcal{T}, \mathcal{A}\rangle$ be a knowledge base. We say that:

- $\mathcal{I}$ $\mathfrak{s}$-*validates* a role axiom $R_1 \circ \ldots \circ R_k \sqsubseteq S$ if $R^{\mathcal{I}}_{1+} \circ \ldots \circ R^{\mathcal{I}}_{k+} \subseteq S^{\mathcal{I}}_{+}$
- $\mathcal{I}$ $\mathfrak{s}$-*validates* a role assertion $\mathtt{Ref}(R)$ (resp. $\mathtt{Irr}(R)$, $\mathtt{Sym}(R)$, $\mathtt{Tra}(R)$) if $R^{\mathcal{I}}_{+}$ is reflexive (resp. irreflexive, symmetric, transitive)
- $\mathcal{I}$ $\mathfrak{s}$-*validates* a role assertion $\mathtt{Dis}(R, S)$ if $R^{\mathcal{I}}_{+}$ and $S^{\mathcal{I}}_{+}$ are disjoint
- $\mathcal{I}$ is an $\mathfrak{s}$-*model* of $\mathcal{R}$, denoted by $\mathcal{I} \models_{\mathfrak{s}} \mathcal{R}$, if it $\mathfrak{s}$-validates all axioms of $\mathcal{R}$

- $\mathcal{I}$ $\mathfrak{s}$-*validates* $C \sqsubseteq D$, denoted by $\mathcal{I} \models_{\mathfrak{s}} C \sqsubseteq D$, if:
  - case $\mathfrak{s}_{\mathtt{GCI}} = w : C^{\mathcal{I}}_{-} \cup D^{\mathcal{I}}_{+} = \Delta^{\mathcal{I}}$
  - case $\mathfrak{s}_{\mathtt{GCI}} = m : C^{\mathcal{I}}_{+} \subseteq D^{\mathcal{I}}_{+}$
  - case $\mathfrak{s}_{\mathtt{GCI}} = s : C^{\mathcal{I}}_{+} \subseteq D^{\mathcal{I}}_{+}$ and $D^{\mathcal{I}}_{-} \subseteq C^{\mathcal{I}}_{-}$
- $\mathcal{I}$ is an $\mathfrak{s}$-*model* of a TBox $\mathcal{T}$, denoted by $\mathcal{I} \models_{\mathfrak{s}} \mathcal{T}$, if it $\mathfrak{s}$-validates all axioms of $\mathcal{T}$

- $\mathcal{I}$ $\mathfrak{s}$-*satisfies* an individual assertion φ if $\mathcal{I} \models_{\mathfrak{s}} \varphi$, where
  $\mathcal{I} \models_{\mathfrak{s}} a \neq b$ if $a^{\mathcal{I}} \neq b^{\mathcal{I}}$
  $\mathcal{I} \models_{\mathfrak{s}} C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}_{+}$
  $\mathcal{I} \models_{\mathfrak{s}} R(a, b)$ if $\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}_{+}$
  $\mathcal{I} \models_{\mathfrak{s}} \neg S(a, b)$ if $\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in S^{\mathcal{I}}_{-}$
- $\mathcal{I}$ is an $\mathfrak{s}$-*model* of $\mathcal{A}$, denoted by $\mathcal{I} \models_{\mathfrak{s}} \mathcal{A}$, if it $\mathfrak{s}$-satisfies all assertions of $\mathcal{A}$

- $\mathcal{I}$ is an $\mathfrak{s}$-*model* of a knowledge base $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ if it is an $\mathfrak{s}$-model of all $\mathcal{R}$, $\mathcal{T}$, $\mathcal{A}$
- $\mathcal{I}$ $\mathfrak{s}$-*satisfies* a query $\varphi = \varphi_1 \wedge \ldots \wedge \varphi_k$, denoted by $\mathcal{I} \models_{\mathfrak{s}} \varphi$, if $\mathcal{I} \models_{\mathfrak{s}} \varphi_i$ for all $1 \leq i \leq k$
- φ is an $\mathfrak{s}$-*logical consequence* of a knowledge base $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$, denoted by $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle \models_{\mathfrak{s}} \varphi$, if every $\mathfrak{s}$-model of $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ $\mathfrak{s}$-satisfies φ.

In [7,8] Ma et al. use non-traditional inclusion axioms $C \mapsto D$, $C \sqsubset D$ and $C \rightarrow D$, which correspond to our inclusion $C \sqsubseteq D$ w.r.t. semantics $\mathfrak{s}$ with $\mathfrak{s}_{\mathtt{GCI}} = w$, m, s, respectively.

See [10] for an example demonstrating the usefulness of paraconsistent semantics.

5 The Relationship between the Semantics

The following proposition states that if $\mathfrak{s} \in \mathfrak{S}$ is a semantics such that $\mathfrak{s}_{\mathtt{C}} = 2$ and $\mathfrak{s}_{\mathtt{R}} = 2$ then $\mathfrak{s}$ coincides with the traditional semantics.

Proposition 5.1. *Let $\mathfrak{s} \in \mathfrak{S}$ be a semantics such that $\mathfrak{s}_{\mathtt{C}} = 2$ and $\mathfrak{s}_{\mathtt{R}} = 2$, let $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ be a knowledge base, and φ be a query. Then $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle \models_{\mathfrak{s}} \varphi$ iff $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle \models \varphi$.* ◁

Proposition 5.2. *Let $\mathfrak{s}, \mathfrak{s}' \in \mathfrak{S}$ be semantics such that $\mathfrak{s}_{\mathtt{R}} = \mathfrak{s}'_{\mathtt{R}} = 2$, $\mathfrak{s}_{\mathtt{C}} = \mathfrak{s}'_{\mathtt{C}}$, $\mathfrak{s}_{\mathtt{GCI}} = \mathfrak{s}'_{\mathtt{GCI}}$, but $\mathfrak{s}_{\forall\exists} \neq \mathfrak{s}'_{\forall\exists}$. Then $\mathfrak{s}$ and $\mathfrak{s}'$ are equivalent in the sense that, for every knowledge base $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ and every query φ, $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle \models_{\mathfrak{s}} \varphi$ iff $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle \models_{\mathfrak{s}'} \varphi$.* ◁

Let $\mathfrak{s}, \mathfrak{s}' \in \mathfrak{S}$. We say that $\mathfrak{s}$ is *weaker than or equal to* $\mathfrak{s}'$ (and $\mathfrak{s}'$ is *stronger than or equal to* $\mathfrak{s}$) if for any knowledge base KB, $Cons_{\mathfrak{s}}(KB) \subseteq Cons_{\mathfrak{s}'}(KB)$. (Recall that $Cons_{\mathfrak{s}}(KB)$ stands for the set of $\mathfrak{s}$-logical consequences of KB.)

Define that $\mathfrak{s}_{\mathtt{C}} \sqsubseteq \mathfrak{s}'_{\mathtt{C}}$ if $\mathfrak{s}'_{\mathtt{C}} \leq \mathfrak{s}_{\mathtt{C}}$, and that $\mathfrak{s}_{\mathtt{R}} \sqsubseteq \mathfrak{s}'_{\mathtt{R}}$ if $\mathfrak{s}'_{\mathtt{R}} \leq \mathfrak{s}_{\mathtt{R}}$, where $\leq$ stands for the usual ordering between natural numbers. Define $\mathfrak{s}_{\mathtt{GCI}} \sqsubseteq \mathfrak{s}'_{\mathtt{GCI}}$ according to $w \sqsubseteq m \sqsubseteq s$, where $\sqsubseteq$ is transitive. Define that $\mathfrak{s} \sqsubseteq \mathfrak{s}'$ if:

$$\mathfrak{s}_{\mathtt{C}} \sqsubseteq \mathfrak{s}'_{\mathtt{C}}, \mathfrak{s}_{\mathtt{R}} \sqsubseteq \mathfrak{s}'_{\mathtt{R}}, \mathfrak{s}_{\forall\exists} = \mathfrak{s}'_{\forall\exists}, \text{ and } \mathfrak{s}_{\mathtt{GCI}} \sqsubseteq \mathfrak{s}'_{\mathtt{GCI}}; \text{ or} \tag{1}$$

$$\mathfrak{s}_{\mathtt{C}} \sqsubseteq \mathfrak{s}'_{\mathtt{C}}, \mathfrak{s}_{\mathtt{R}} = \mathfrak{s}'_{\mathtt{R}} = 2, \text{ and } \mathfrak{s}_{\mathtt{GCI}} \sqsubseteq \mathfrak{s}'_{\mathtt{GCI}}; \text{ or} \tag{2}$$

$$\mathfrak{s}_{\mathtt{C}} = \mathfrak{s}'_{\mathtt{C}} = 2 \text{ and } \mathfrak{s}_{\mathtt{R}} = \mathfrak{s}'_{\mathtt{R}} = 2. \tag{3}$$

Theorem 5.3. *Let $\mathfrak{s}, \mathfrak{s}' \in \mathfrak{S}$ be semantics such that $\mathfrak{s} \sqsubseteq \mathfrak{s}'$. Then $\mathfrak{s}$ is weaker than or equal to $\mathfrak{s}'$ (i.e., for any knowledge base KB, $Cons_{\mathfrak{s}}(KB) \subseteq Cons_{\mathfrak{s}'}(KB)$).* ◁

The following corollary follows from the theorem and Proposition 5.1. It states that all the semantics from $\mathfrak{S}$ give only correct answers.

Corollary 5.4. *Let $\mathfrak{s} \in \mathfrak{S}$ and let $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ be a knowledge base and φ be a query. Then $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle \models_{\mathfrak{s}} \varphi$ implies $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle \models \varphi$.* ◁

6 A Translation into the Traditional Semantics

In this section we give a linear translation $\pi_{\mathfrak{s}}$, for $\mathfrak{s} \in \mathfrak{S}$ with $\mathfrak{s}_{\mathtt{C}} \in \{3, 4\}$, $\mathfrak{s}_{\mathtt{R}} \in \{2, 4\}$ and $\mathfrak{s}_{\forall\exists} = +$, such that, for every knowledge base KB and every query φ, $KB \models_{\mathfrak{s}} \varphi$ iff $\pi_{\mathfrak{s}}(KB) \models \pi_{\mathfrak{s}}(\varphi)$. In this section, if not otherwise stated, we assume that $\mathfrak{s}$ satisfies the mentioned conditions.

For $A \in \mathbf{C} \setminus \mathbf{N}$, let A_+ and A_- be new concept names. For $r \in \mathbf{R} \setminus \{U\}$, let r_+ and r_- be new role names. With respect to the considered semantics $\mathfrak{s}$, let $\mathbf{C}' = \{A_+, A_- \mid A \in \mathbf{C} \setminus \mathbf{N}\} \cup \mathbf{N}$, and $\mathbf{R}' = \mathbf{R}$ if $\mathfrak{s}_{\mathtt{R}} = 2$, and $\mathbf{R}' = \{r_+, r_- \mid r \in \mathbf{R} \setminus \{U\}\} \cup \{U\}$ if $\mathfrak{s}_{\mathtt{R}} = 4$.

We define also two auxiliary translations $\pi_{\mathfrak{s}+}$ and $\pi_{\mathfrak{s}-}$. In the following, if not otherwise stated, r, R, S, A, C, D, a, b, $\mathcal{R}$, $\mathcal{T}$, $\mathcal{A}$ are arbitrary elements of their appropriate types (according to the used convention) in the language using $\mathbf{C}$ and $\mathbf{R}$.

If $\mathfrak{s}_{\mathtt{R}} = 2$ then:

- $\pi_{\mathfrak{s}+}(R) \stackrel{\text{def}}{=} R$ and $\pi_{\mathfrak{s}}(\mathcal{R}) \stackrel{\text{def}}{=} \mathcal{R}$
- $\pi_{\mathfrak{s}}(R(a,b)) \stackrel{\text{def}}{=} R(a,b)$ and $\pi_{\mathfrak{s}}(\neg S(a,b)) \stackrel{\text{def}}{=} \neg S(a,b)$
- $\pi_{\mathfrak{s}+}(\exists R.\mathtt{Self}) \stackrel{\text{def}}{=} \exists R.\mathtt{Self}$ and $\pi_{\mathfrak{s}-}(\exists R.\mathtt{Self}) \stackrel{\text{def}}{=} \neg\exists R.\mathtt{Self}$.

If $\mathfrak{s}_{\mathtt{R}} = 4$ then:

- $\pi_{\mathfrak{s}+}(U) \stackrel{\text{def}}{=} U$
- $\pi_{\mathfrak{s}+}(r) \stackrel{\text{def}}{=} r_+$ and $\pi_{\mathfrak{s}-}(r) \stackrel{\text{def}}{=} r_-$, where $r \neq U$
- $\pi_{\mathfrak{s}+}(r^-) \stackrel{\text{def}}{=} (r_+)^-$ and $\pi_{\mathfrak{s}-}(r^-) \stackrel{\text{def}}{=} (r_-)^-$, where $r \neq U$
- for every role axiom φ, $\pi_{\mathfrak{s}}(\varphi) \stackrel{\text{def}}{=} \varphi'$, where φ' is the role axiom obtained from φ by replacing each role R by $\pi_{\mathfrak{s}+}(R)$
- $\pi_{\mathfrak{s}}(\mathcal{R}) \stackrel{\text{def}}{=} \{\pi_{\mathfrak{s}}(\varphi) \mid \varphi \in \mathcal{R}\}$
- $\pi_{\mathfrak{s}}(R(a,b)) \stackrel{\text{def}}{=} \pi_{\mathfrak{s}+}(R)(a,b)$ and $\pi_{\mathfrak{s}}(\neg S(a,b)) \stackrel{\text{def}}{=} \pi_{\mathfrak{s}-}(S)(a,b)$, where $R, S \neq U$
- $\pi_{\mathfrak{s}+}(\exists R.\mathtt{Self}) \stackrel{\text{def}}{=} \exists \pi_{\mathfrak{s}+}(R).\mathtt{Self}$ and $\pi_{\mathfrak{s}-}(\exists R.\mathtt{Self}) \stackrel{\text{def}}{=} \exists \pi_{\mathfrak{s}-}(R).\mathtt{Self}$.

$$
\begin{array}{rcl@{\qquad}rcl}
\pi_{\mathfrak{s}+}(\top) & \stackrel{\text{def}}{=} & \top & \pi_{\mathfrak{s}-}(\top) & \stackrel{\text{def}}{=} & \bot \\
\pi_{\mathfrak{s}+}(\bot) & \stackrel{\text{def}}{=} & \bot & \pi_{\mathfrak{s}-}(\bot) & \stackrel{\text{def}}{=} & \top \\
\pi_{\mathfrak{s}+}(\neg C) & \stackrel{\text{def}}{=} & \pi_{\mathfrak{s}-}(C) & \pi_{\mathfrak{s}-}(\neg C) & \stackrel{\text{def}}{=} & \pi_{\mathfrak{s}+}(C) \\
\pi_{\mathfrak{s}+}(C \sqcap D) & \stackrel{\text{def}}{=} & \pi_{\mathfrak{s}+}(C) \sqcap \pi_{\mathfrak{s}+}(D) & \pi_{\mathfrak{s}-}(C \sqcap D) & \stackrel{\text{def}}{=} & \pi_{\mathfrak{s}-}(C) \sqcup \pi_{\mathfrak{s}-}(D) \\
\pi_{\mathfrak{s}+}(C \sqcup D) & \stackrel{\text{def}}{=} & \pi_{\mathfrak{s}+}(C) \sqcup \pi_{\mathfrak{s}+}(D) & \pi_{\mathfrak{s}-}(C \sqcup D) & \stackrel{\text{def}}{=} & \pi_{\mathfrak{s}-}(C) \sqcap \pi_{\mathfrak{s}-}(D) \\
\pi_{\mathfrak{s}+}(\forall R.C) & \stackrel{\text{def}}{=} & \forall \pi_{\mathfrak{s}+}(R).\pi_{\mathfrak{s}+}(C) & \pi_{\mathfrak{s}-}(\forall R.C) & \stackrel{\text{def}}{=} & \exists \pi_{\mathfrak{s}+}(R).\pi_{\mathfrak{s}-}(C) \\
\pi_{\mathfrak{s}+}(\exists R.C) & \stackrel{\text{def}}{=} & \exists \pi_{\mathfrak{s}+}(R).\pi_{\mathfrak{s}+}(C) & \pi_{\mathfrak{s}-}(\exists R.C) & \stackrel{\text{def}}{=} & \forall \pi_{\mathfrak{s}+}(R).\pi_{\mathfrak{s}-}(C) \\
\pi_{\mathfrak{s}+}(\geq n\, R.C) & \stackrel{\text{def}}{=} & \geq n\, \pi_{\mathfrak{s}+}(R).\pi_{\mathfrak{s}+}(C) & \pi_{\mathfrak{s}-}(\geq (n+1)\, R.C) & \stackrel{\text{def}}{=} & \leq n\, \pi_{\mathfrak{s}+}(R).\neg\pi_{\mathfrak{s}-}(C) \\
 & & & \pi_{\mathfrak{s}-}(\geq 0\, R.C) & \stackrel{\text{def}}{=} & \bot \\
\pi_{\mathfrak{s}+}(\leq n\, R.C) & \stackrel{\text{def}}{=} & \leq n\, \pi_{\mathfrak{s}+}(R).\neg\pi_{\mathfrak{s}-}(C) & \pi_{\mathfrak{s}-}(\leq n\, R.C) & \stackrel{\text{def}}{=} & \geq (n+1)\, \pi_{\mathfrak{s}+}(R).\pi_{\mathfrak{s}+}(C)
\end{array}
$$

Fig. 1. A partial specification of $\pi_{\mathfrak{s}+}$ and $\pi_{\mathfrak{s}-}$

If A is a nominal then $\pi_{\mathfrak{s}+}(A) \stackrel{\text{def}}{=} A$ and $\pi_{\mathfrak{s}-}(A) \stackrel{\text{def}}{=} \neg A$.

If A is a concept name but not a nominal then $\pi_{\mathfrak{s}+}(A) \stackrel{\text{def}}{=} A_+$ and $\pi_{\mathfrak{s}-}(A) \stackrel{\text{def}}{=} A_-$.

The translations $\pi_{\mathfrak{s}+}(C)$ and $\pi_{\mathfrak{s}-}(C)$ for the case C is not of the form A or $\exists R.\texttt{Self}$ are defined as in Figure 1.

Define $\pi_{\mathfrak{s}}(C \sqsubseteq D)$ and $\pi_{\mathfrak{s}}(\mathcal{T})$ as follows:

- case $\mathfrak{s}_{\texttt{GCI}} = w : \pi_{\mathfrak{s}}(C \sqsubseteq D) \stackrel{\text{def}}{=} \{\top \sqsubseteq \pi_{\mathfrak{s}-}(C) \sqcup \pi_{\mathfrak{s}+}(D)\}$
- case $\mathfrak{s}_{\texttt{GCI}} = m : \pi_{\mathfrak{s}}(C \sqsubseteq D) \stackrel{\text{def}}{=} \{\pi_{\mathfrak{s}+}(C) \sqsubseteq \pi_{\mathfrak{s}+}(D)\}$
- case $\mathfrak{s}_{\texttt{GCI}} = s : \pi_{\mathfrak{s}}(C \sqsubseteq D) \stackrel{\text{def}}{=} \{\pi_{\mathfrak{s}+}(C) \sqsubseteq \pi_{\mathfrak{s}+}(D),\ \pi_{\mathfrak{s}-}(D) \sqsubseteq \pi_{\mathfrak{s}-}(C)\}$
- case $\mathfrak{s}_{\texttt{C}} = 3 : \pi_{\mathfrak{s}}(\mathcal{T}) \stackrel{\text{def}}{=} \bigcup_{\varphi \in \mathcal{T}} \pi_{\mathfrak{s}}(\varphi) \cup \{\top \sqsubseteq A_+ \sqcup A_- \mid A \in \mathbf{C} \setminus \mathbf{N}\}$
- case $\mathfrak{s}_{\texttt{C}} = 4 : \pi_{\mathfrak{s}}(\mathcal{T}) \stackrel{\text{def}}{=} \bigcup_{\varphi \in \mathcal{T}} \pi_{\mathfrak{s}}(\varphi)$.

Define that:

- $\pi_{\mathfrak{s}}(a \neq b) \stackrel{\text{def}}{=} a \neq b$ and $\pi_{\mathfrak{s}}(C(a)) \stackrel{\text{def}}{=} \pi_{\mathfrak{s}+}(C)(a)$
- $\pi_{\mathfrak{s}}(\mathcal{A}) \stackrel{\text{def}}{=} \{\pi_{\mathfrak{s}}(\varphi) \mid \varphi \in \mathcal{A}\}$
- $\pi_{\mathfrak{s}}(\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle) \stackrel{\text{def}}{=} \langle \pi_{\mathfrak{s}}(\mathcal{R}), \pi_{\mathfrak{s}}(\mathcal{T}), \pi_{\mathfrak{s}}(\mathcal{A}) \rangle$
- for a query $\varphi = \varphi_1 \wedge \ldots \wedge \varphi_k$, define $\pi_{\mathfrak{s}}(\varphi) \stackrel{\text{def}}{=} \pi_{\mathfrak{s}}(\varphi_1) \wedge \ldots \wedge \pi_{\mathfrak{s}}(\varphi_k)$.

Note that, if $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ is a knowledge base and φ is a query in $\mathcal{SROIQ}$ using $\mathbf{C}$ and $\mathbf{R}$, then $\pi_{\mathfrak{s}}(\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle)$ is a knowledge base and $\pi_{\mathfrak{s}}(\varphi)$ is a query in $\mathcal{SROIQ}$ using $\mathbf{C}'$ and $\mathbf{R}'$, with the property that:

- the length of $\pi_{\mathfrak{s}}(\varphi)$ is linear in the length of φ
- the size of $\pi_{\mathfrak{s}}(\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle)$ is linear in the size of $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ in the case $\mathfrak{s}_{\texttt{C}} = 4$, and linear in the sizes of $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ and $\mathbf{C} \setminus \mathbf{N}$ in the case $\mathfrak{s}_{\texttt{C}} = 3$.[1]

Theorem 6.1. *Let $\mathfrak{s} \in \mathfrak{S}$ be a semantics such that $\mathfrak{s}_{\texttt{C}} \in \{3, 4\}$, $\mathfrak{s}_{\texttt{R}} \in \{2, 4\}$ and $\mathfrak{s}_{\forall\exists} = +$. Let $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ be a knowledge base and φ be a query in the language using $\mathbf{C}$ and $\mathbf{R}$. Then $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle \models_{\mathfrak{s}} \varphi$ iff $\pi_{\mathfrak{s}}(\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle) \models \pi_{\mathfrak{s}}(\varphi)$.* ◁

[1] Where the notions of length and size are defined as usual.

To check whether $\pi_{\mathfrak{s}}(\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle) \models \pi_{\mathfrak{s}}(\varphi)$ one can use, e.g., the tableau method given in [4]. We have the following corollary of Theorem 6.1 by taking $C = \bot$.

Corollary 6.2. *Let $\mathfrak{s} \in \mathfrak{S}$ be a semantics such that $\mathfrak{s}_{\mathbf{C}} \in \{3, 4\}$, $\mathfrak{s}_{\mathbf{R}} \in \{2, 4\}$ and $\mathfrak{s}_{\forall\exists} = +$, and let $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ be a knowledge base in the language using $\mathbf{C}$ and $\mathbf{R}$. Then $\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle$ is $\mathfrak{s}$-satisfiable iff $\pi_{\mathfrak{s}}(\langle \mathcal{R}, \mathcal{T}, \mathcal{A} \rangle)$ is satisfiable (w.r.t. the traditional semantics).* ◁

7 Conclusions

$\mathcal{SROIQ}$ is a powerful DL used as the logical foundation of OWL 2. In this work, we introduced and studied a number of different paraconsistent semantics for $\mathcal{SROIQ}$ in a uniform way. We gave a translation of the problem of conjunctive query answering w.r.t. some of the considered paraconsistent semantics into a version that uses the traditional semantics. This allows to directly use existing tools and reasoners of $\mathcal{SROIQ}$ for paraconsistent reasoning. We also presented a formalization of rough concepts in $\mathcal{SROIQ}$.

Note that answering queries that contain negative individual assertions of the form $\neg S(a, b)$ using a paraconsistent semantics is first studied in this work. Also note that only a four-valued paraconsistent semantics has previously been introduced for $\mathcal{SROIQ}$ [7] (without some important features of $\mathcal{SROIQ}$). If $\mathfrak{s}, \mathfrak{s}' \in \mathfrak{S}$ are semantics such that $\mathfrak{s} \sqsubseteq \mathfrak{s}'$ then, by Theorem 5.3, for the conjunctive query answering problem, $KB \models_{\mathfrak{s}'} \varphi$ approximates $KB \models \varphi$ better than $KB \models_{\mathfrak{s}} \varphi$ does. Our postulate is that, if $\mathfrak{s} \sqsubseteq \mathfrak{s}'$ and KB is $\mathfrak{s}'$-satisfiable, then it is better to use $\mathfrak{s}'$ than $\mathfrak{s}$. In particular, one should use semantics $\mathfrak{s}$ with $\mathfrak{s}_{\mathbf{C}} = \mathfrak{s}_{\mathbf{R}} = 4$ (i.e. four-valued semantics) only when the considered knowledge base is $\mathfrak{s}'$-unsatisfiable in semantics $\mathfrak{s}'$ with $\mathfrak{s}'_{\mathbf{C}} = 3$.

Acknowledgments. This work is supported by grants N N206 399334 and N N206 370739 from the Polish Ministry of Science and Higher Education.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): Description Logic Handbook. Cambridge University Press, Cambridge (2002)
2. Béziau, J.-Y., Carnielli, W., Gabbay, D.M. (eds.): Handbook of Paraconsistency. Logic and cognitive systems, vol. 9. College Publications (2007)
3. Easterbrook, S.M., Chechik, M.: A framework for multi-valued reasoning over inconsistent viewpoints. In: Proc. of ICSE 2001, pp. 411–420. IEEE Computer Society Press, Los Alamitos (2001)
4. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: Proc. of KR 2006, pp. 57–67. AAAI Press, Menlo Park (2006)
5. Jiang, Y., Wang, J., Tang, S., Xiao, B.: Reasoning with rough description logics: An approximate concepts approach. Inf. Sci. 179(5), 600–612 (2009)
6. Klein, M.C.A., Mika, P., Schlobach, S.: Rough description logics for modeling uncertainty in instance unification. In: Proc. of URSW 2007. CEUR Workshop Proc., vol. 327 (2007)
7. Ma, Y., Hitzler, P.: Paraconsistent reasoning for OWL 2. In: Polleres, A. (ed.) RR 2009. LNCS, vol. 5837, pp. 197–211. Springer, Heidelberg (2009)

8. Ma, Y., Hitzler, P., Lin, Z.: Paraconsistent reasoning for expressive and tractable description logics. In: Proc. of Description Logics (2008)
9. Meghini, C., Straccia, U.: A relevance terminological logic for information retrieval. In: Proc. of SIGIR 1996, pp. 197–205. ACM, New York (1996)
10. Nguyen, L.A.: The long version of the current paper, http://www.mimuw.edu.pl/~nguyen/pSROIQ-long.pdf
11. Nguyen, L.A., Szałas, A.: Three-valued paraconsistent reasoning for Semantic Web agents. Accepted for KES-AMSTA 2010 (2010)
12. Odintsov, S.P., Wansing, H.: Inconsistency-tolerant description logic. part II: A tableau algorithm for CACL^{c}. Journal of Applied Logic 6(3), 343–360 (2008)
13. Pawlak, Z.: Rough sets. Int. Journal of Computer and Information Science 11, 341–356 (1982)
14. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
15. Pawlak, Z., Skowron, A.: Rough sets and boolean reasoning. Inf. Sci. 177(1), 41–73 (2007)
16. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Inf. Sci. 177(1), 28–40 (2007)
17. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Inf. Sci. 177(1), 3–27 (2007)
18. Schlobach, S., Klein, M.C.A., Peelen, L.: Description logics with approximate definitions - precise modeling of vague concepts. In: Proc. of IJCAI 2007, pp. 557–562. AAAI Press, Menlo Park (2007)
19. Straccia, U.: A sequent calculus for reasoning in four-valued description logics. In: Galmiche, D. (ed.) TABLEAUX 1997. LNCS, vol. 1227, pp. 343–357. Springer, Heidelberg (1997)
20. Vitória, A., Maluszyński, J., Szałas, A.: Modeling and reasoning in paraconsistent rough sets. Fundamenta Informaticae 97(4), 405–438 (2009)
21. Zhang, X., Qi, G., Ma, Y., Lin, Z.: Quasi-classical semantics for expressive description logics. In: Proc. of Description Logics (2009)

Representation of Granularity for Non-Euclidian Relational Data by Jaccard Coefficients and Binary Classifications

Shoji Hirano and Shusaku Tsumoto

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

Abstract. In this paper we present a method for representing the granularity for asymmetric, non-Euclidean relational data. It firstly builds a set of binary classifications based on the directional similarity from each object. After that, the strength of discrimination knowledge is quantified as the indiscernibility of objects based on the Jaccard similarity coefficients between the classifications. Fine but weak discrimination knowledge supported by the small number of binary classifications is more likely to be coarsened than those supported by the large number of classifications, and coarsening of discrimination knowledge causes the merging of objects. Accoding to this feature, we represent the hierarchical structure of data granules by a dendrogram generated by applying the complete-linkage hierarchical grouping method to the derived indiscernibility. This enables users to change the coarseness of discrimination knowledge and thus to control the size of granules.

1 Introduction

Non-Euclidean relational data play an important role in application areas such as social sciences, where asymmetric relationships between subjects can be observed and need to be analyzed. Examples include subjectively judged relations between students and input/output of the persons between countries [1]. Non-Euclidean relational data involves the following properties: (1) objects are not represented in a usual feature vector space but their relationships (usually similarity or dissimilarity) are measured and stored in a relational data matrix. (2) The dissimilarity can be non-metric; that means the dissimilarity may not satisfy the triangular inequality nor symmetry.

Building granules in Non-Euclidean relational data is still a challenging problem. Since attribute vectors do not exist therein, splitting or merging of blocks in the attribute space may not be directly applied. Additionally, since dissimilarities are non-metric, the choice of grouping methods is in general much limited compared to the cases of metric and/or non-relational data. For example, methods such as k-means may not be directly applied to this type of data as they assume the existence of data vectors. Conventional hierarchical clusterings are

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 721–729, 2010.

capable of dealing with relative or subjective measures. However, they involve other problems such as erosion or expansion of data space by intermediate objects between large clusters, and in some cases the results may change according to the order of processing objects [4]. The NERF c-means proposed by Hathaway et al. [5] is an extension of fuzzy c-means and capable of handling the non-Euclidean relational data. However, as it is a sort of partitional clustering method, it is still difficult to examine the structure of the data, namely, the hierarchy of data groups. Additionally, most of these methods are not designed to deal with asymmetric dissimilarity.

In this paper we present a method for representing the granularity for asymmetric, non-Euclidean relational data. It firstly builds a set of binary classifications based on the directional similarity from each object. After that, the strength of discrimination knowledge is quantified as the indiscernibility of objects based on the Jaccard similarity coefficients between the classifications. Fine but weak discrimination knowledge supported by the small number of binary classifications is more likely to be coarsened than those supported by the large number of classifications, and coarsening of discrimination knowledge causes the merging of objects. Accoding to this feature, we represent the hierarchical structure of data granules by a dendrogram generated by applying the complete-linkage hierarchical grouping method to the derived indiscernibility. This enables users to change the coarseness of discrimination knowledge and thus to control the size of granules.

The remainder of this paper is organized as follows. Section 2 briefly provides definitions used in this work. Section 3 shows the method in detail with some examples. Section 4 shows experimental results on a synthetic data, and Section 5 shows conclusions and future work.

2 Preliminaries

This section provides basic definitions about indiscernibility, mostly came from the literature of Rough Sets [2]. Let $U \neq \phi$ be a universe of discourse and X be a subset of U. An equivalence relation R classifies U into a set of subsets $U/R = \{X_1, X_2, ...X_N\}$ that satisfies the following conditions: (1) $X_i \subseteq U$, $X_i \neq \phi$ for any i, (2)$X_i \cap X_j = \phi$ for any $i, j, i \neq j$, (3) $\cup_{i=1,2,...N} X_i = U$. Any subset X_i is called a category and represents an equivalence class of R. A category in R containing an object $x \in U$ is denoted by $[x]_R$. Objects x_i and x_j in U are *indiscernible on* R if $(x_i, x_j) \in P$ where $P \in U/R$. For a family of equivalence relations $\mathbf{P} \subseteq \mathbf{R}$, an indiscernibility relation over $\mathbf{P}$ is defined as the intersection of individual relations $Q \in \mathbf{P}$.

The Jaccard coefficient $J(A, B)$ between two sets A and B is defined by

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

For two objects x_i and x_j each of which has p binary attributes, the Jaccard similarity coefficient $J(x_i, x_j)$ is defined by

$$J(x_i, x_j) = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$

where f_{11} represents the number of attributes whose values are 1 for both of x_i and x_j. Analogously, f_{01} and f_{10} are defined as the number of attributes whose values are (0,1) and (1,0) for x_i and x_j respectively [3].

3 Method

The proposed method consists of three steps:

1. Assign a binary classification to each object.
2. Compute the Jaccard similarity coefficient for each pair of objects according to the binary classifications.
3. Construct a dendrogram that represents hierarchy of granules by applying hierarchical linkage algorithm based on the derived indiscernibility.

3.1 Binary Classifications

Let $U = \{x_1, x_2, \ldots, x_n\}$ be a set of objects we are interested in, and

$$S = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{pmatrix} \tag{1}$$

be a similarity matrix for objects in U where s_{ij} denotes similarity between x_i and x_j. Since we deal with non-Euclidean relational data, S can be asymmetric; hence s_{ij} can be $\neq s_{ji}$.

To begin with, for each object x_i, we consider a binary classification of U based on s_i. This binary classification is formalized using an equivalence relation R_i as follows.

$$U/R_i = \{P_i,\ U - P_i\}, \tag{2}$$

where

$$P_i = \{x_j |\ s_{ij} \geq Th_i\}, \quad \forall x_j \in U. \tag{3}$$

Th_i denotes a threshold value of similarity for x_i. Set P_i contains objects that are indiscernible to x_i, and $U - P_i$ contains objects that are discernible to x_i. Note that P_i is determined with respect to the similarity observed from x_i, hence, $x_j \in P_i$ does not necessarily imply $x_i \in P_j$ when $s_{ij} \neq s_{ji}$.

[Example 1]: Binary Classification
Let us assume $U = \{x_1, x_2, x_3, x_4, x_5\}$ and consider an asymmetric, non-Euclidean dissimilarity matrix shown in Table 1. Suppose we define binary classifications U/R_i as

$$\begin{aligned} U/R_i &= \{P_i,\ U - P_i\}, \\ P_i &= \{x_j |\ s_{ij} \geq 0.5\}, \quad \forall x_j \in U. \end{aligned} \tag{4}$$

Then we obtain the following five binary classifications.

$$\begin{aligned} U/R_1 &= \{\{x_1, x_2, x_3\}, \{x_4, x_5\}\}, \\ U/R_2 &= \{\{x_1, x_2, x_3\}, \{x_4, x_5\}\}, \\ U/R_3 &= \{\{x_2, x_3, x_4\}, \{x_1, x_5\}\}, \\ U/R_4 &= \{\{x_1, x_2, x_3, x_4\}, \{x_5\}\}, \\ U/R_5 &= \{\{x_4, x_5\}, \{x_1, x_2, x_3\}\}. \end{aligned} \tag{5}$$

Note that these five classifications are derived independently. Objects such as x_1 and x_3 are classified as indiscernible in U/R_1 and U/R_2, but classified as discernible in U/R_3. This reflects asymmetric property of the similarity; since $s_{13} = 0.9$, x_3 is included in P_1, however, since $s_{31} = 0.3$, x_1 is not include in P_3. □

Table 1. An example of asymmetric, non-Euclidean dissimilarity matrix

	x_1	x_2	x_3	x_4	x_5
x_1	1.0	0.9	0.9	0.3	0.1
x_2	0.8	1.0	0.9	0.4	0.2
x_3	0.3	0.9	1.0	0.8	0.2
x_4	0.8	0.7	0.8	1.0	0.4
x_5	0.3	0.4	0.1	0.9	1.0

3.2 Indiscernibility Based on Jaccard Similarity Coefficient

As described in the previous section, asymmetry of the similarity can cause the difference of class belongingness of objects over all binary classifications. These classifications also represent the global similarity between objects because similar objects are likely to be classified into the same class in most of the classifications. In other words, when a pair of objects is classified into the same class by most of the equivalence relations, there are less argument for treating these objects as indiscernible.

Based on these observations, we propose to quantify the indiscernibility of objects according to the Jaccard similarity coefficient of the binary classifications obtained from asymmetric similarity matrix. The key point is that, we assess the indiscernibility of objects according to the global similarity of local classifications.

We firstly introduce a binary classification matrix, C, defined as follows.

$$C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix} \tag{6}$$

where

$$c_{ij} = \begin{cases} 1 & \text{if } x_i \in P_j \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Matrix C can be translated from a binary classification table shown in Table 2. Each column C_{R_i} in C corresponds to the transpose of binary classification U/R_i such that objects in P_i is represented as value 1.

Table 2. Binary classification table

	C_{R_1}	C_{R_2}	$\cdots$	C_{R_n}
x_1	c_{11}	c_{12}	$\cdots$	c_{1n}
x_2	c_{21}	c_{22}	$\cdots$	c_{2n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
x_n	c_{n1}	c_{n2}	$\cdots$	c_{nn}

Let the i-th raw of C be $C_i = \{c_{i1}, c_{i2}, \ldots, c_{in}\}$. Then, we define the indiscernibility of objects x_i and x_j as follows.

$$\text{indis}(x_i, x_j) = J(C_i, C_j) \tag{8}$$

where $J(C_i, C_j)$ denotes the Jaccard similarity coefficient between two sets C_i and C_j that contain binary values. According to the definition in Section 2, the term f_{11} corresponds to the number of cases that satisfy $c_{ik} = c_{jk} = 1$ for $1 \leq k \leq n$. It means that f_{11} quantifies the number of binary classifications in which x_i and x_j are positively classified as indiscernible with respect to the similarity from x_k.

[Example 2]: Indiscernibility based on Jaccard Similarity Coefficient Let us consider the case in Example 1. According to $U/R_1 = \{\{x_1, x_2, x_3\}, \{x_4, x_5\}\}$, the first column of binary classification table can be written as

$$C_{R_1}^T = \{1\ 1\ 1\ 0\ 0\}.$$

By applying this from U/R_1 to U/R_5, we obtain

$$C = \begin{pmatrix} C_{R_1}^T & C_{R_2}^T & C_{R_3}^T & C_{R_4}^T & C_{R_5}^T \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{9}$$

Table 3. Indiscernibility of objects in Example 1

	x_1	x_2	x_3	x_4	x_5
x_1	3/3	3/4	3/4	1/5	0/4
x_2		4/4	4/4	2/5	0/5
x_3			4/4	2/5	0/5
x_4				3/3	1/3
x_5					1/1

Table 4. Indiscernibility of objects in Example 1(recalculated)

	x_1	x_2	x_3	x_4	x_5
x_1	1.0	0.75	0.75	0.2	0.0
x_2		1.0	1.0	0.4	0.0
x_3			1.0	0.4	0.0
x_4				1.0	0.33
x_5					1.0

The first and second raws of C are $C_1 = \{1, 1, 0, 1, 0\}$ and $C_2 = \{1, 1, 1, 1, 0\}$ respectively. Therefore, we obtain the indiscernibility of objects x_1 and x_2 as

$$\begin{aligned} \mathrm{indis}(x_1, x_2) &= J(C_1, C_2) \\ &= \frac{3}{1+0+3} = \frac{3}{4}. \end{aligned} \tag{10}$$

Similarly, we obtain the indiscernibility for all pairs as shown in Table 3. □

3.3 Hierarchical Representation of Data Granularity

The indiscernibility $\mathrm{indis}(x_i, x_j)$ can be associated with the strength of knowledge for discriminating objects. The larger value of $\mathrm{indis}(x_i, x_j)$ implies that there are less binary classifications that can discriminate these objects. In contrast, the smaller value of it implies that there are more binary classifications that can discriminate them. If we merge objects with some $\mathrm{indis}(x_i, x_j) < 1$ as indiscernible, it means that we disable the ability of knowledge for discriminating them; in other words, it corresponds to the coarsening of classification knowledge. Knowledge that is supported by a small number of binary classifications is fine but weak and more likely to be coarsened compared to that supported by a large number of classifications. And it is a stepwise abstraction process that goes hierarchically from bottom to top according to the indiscernibility. Therefore, it is possible to construct a dendrogram that represents the hierarchy of indiscernibility by using conventional hierarchical grouping method. By setting an appropriate threshold on the dendrogram, one can obtain abstracted granules of objects that meet the given level of indiscernibility. Namely, one can interactively change the granularity of data. The lowest threshold produces the finest groups of objects (granules) and the highest threshold produces the coarsest groups. In order to ensure that all pairs in the group are indiscernible with respect to the given threshold value of indiscernibility, we use complete-linkage criterion for hierarchical grouping.

[Example 3]: Hierarchical Representation of Data Granularity
Let us recall the case in Example 2. The matrix of indiscernibility is provided in Table 3. For easy understandings, we provide in Table 4 recalculated values.

Table 5 and Figure 1 provide the detail of merging process and the dendrogram respectively. Since $\mathrm{indis}(x_2, x_3) = 1.0$, these objects are indiscernible at the

Table 5. Hierarchical merge process

Step	pairs	indis	clusters
1	x_2, x_3	1.0	$\{x_1\}\{x_2, x_3\}\{x_4\}\{x_5\}$
2	x_1, x_2	0.75	$\{x_1, x_2, x_3\}\{x_4\}\{x_5\}$
3	x_4, x_5	0.33	$\{x_1, x_2, x_3\}\{x_4, x_5\}$
4	x_1, x_5	0.0	$\{x_1, x_2, x_3, x_4, x_5\}$

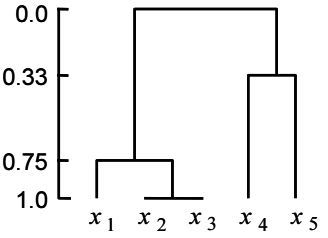


Fig. 1. Dendrogram for Example 3

lowest level; thus $\{x_1\}, \{x_2, x_3\}, \{x_4\}, \{x_5\}$ constitute the finest sets of objects (granules) next to the independent objects. At indis = 0.75, x_1 becomes indiscernible with x_2. Since x_2 and x_3 are also indiscernible, $\{x_1, x_2, x_3\}, \{x_4\}, \{x_5\}$ constitute an abstracted sets of objects. Similarly, at indis = 0.33, x_4 becomes indiscernible with x_5 and $\{x_1, x_2, x_3\}, \{x_4, x_5\}$ constitute the more abstracted sets of objects. Finally, at indis = 0.0, all objects are considered to be indiscernible and the most abstracted set is obtained. The level of abstraction can be interactively set by changing the threshold value on the dendrogram.

The coarsening is performed based on the complete-linkage criterion. For example, on indis = 0.3, all pairs in the groups $\{x_1, x_2, x_3\}, \{x_4, x_5\}$ surely satisfy $\text{indis}(x_i, x_j) > 0.3$. □

4 Experimental Results

We applied our method to a synthetic dataset in order to test its basic functionality. The dataset contained 19 objects in two-dimensional space as shown in Figure 2. The dataset was generated by Neyman-Scott method [6] with cluster number = 3. The label 'cls 1' to 'cls 3' shows the original class that each object belongs to.

The proposed method starts with determining a binary classification, U/R_i, for each object x_i, $i = 1, 2, \ldots, 19$. In order to seclude the inference of methods/parameters for determining U/R_i, we used the following exact binary classifications, which were generated based on the class labels of data.

$$\begin{aligned} U/R_i &= \{P_i,\ U - P_i\}, \\ P_i &= \{x_j |\ c[x_i] = c[x_j]\}, \quad \forall x_j \in U. \end{aligned} \tag{11}$$

Then, in order to simulate the non-Euclidean properties, we applied random disturbance to the exact binary classifications. Taking the randomly disturbed exact classifications as input, we calculated the indiscernibility and constructed a dendrogram. Table 6 provides all the disturbed binary classifications ($U - P_i$ omitted for simplicity. x of x_i also omitted in P_i for simplicity).

Using the binary classifications in Table 6, we calculated indiscernibility based on Jaccard similarity coefficient for each pair of objects. Then we generated the dendrogram using complete-linkage criterion as shown in Figure 3.

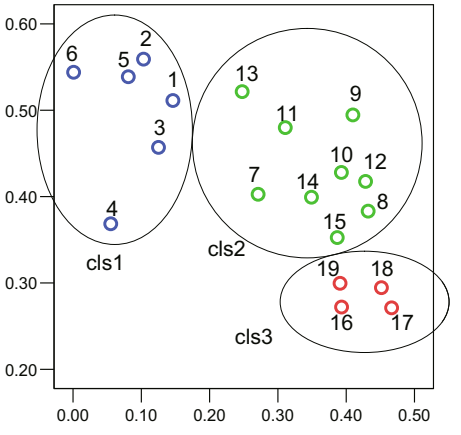


Fig. 2. 2D plot of the test data

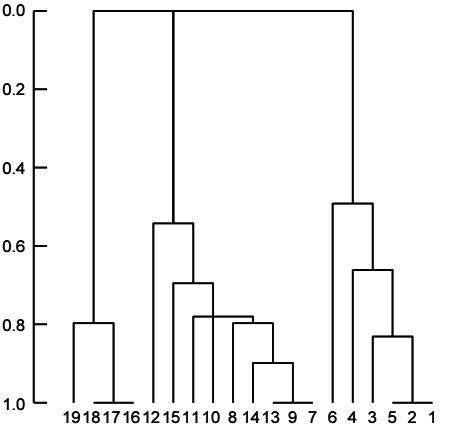


Fig. 3. Dendrogram for the Test data

At the lowest level of indiscernibility, 13 sets of objects were generated as the finest granules next to the independent objects because the randomly disturbed binary classifications were slightly different each other and in ensemble they provide very fine classification knowledge. However, it meant that the ability for discriminating objects was provided by a small number of binary classifications that assigned different classes to the objects. In other words, the strength of knowledge for discriminating objects was relatively weak. Therefore, in general they could be easily coarsened at the early steps of merging objects with high indiscernibility. In the dendrogram in Figure 3, we could observe this property around indis = 0.8. Weak discrimination knowledge was disabled, and most of the objects that had belonged to the same original class became indiscernible. Around indis = 0.5, objects were completely classified into the original three classes, based on the strong classification knowledge inherited from the exact binary classifications.

The above results demonstrated that (1)the proposed method could visualize the hierarchy of indiscernibility using dendrogram, (2) by changing the threshold

Table 6. Binary classifications for the test data

x_i	P_i of U/R_i	x_i	P_i of U/R_i
x_1	1 2 4 5 6 15	x_{11}	7 8 9 10 11 12 13 14 15 12
x_2	1 2 3 4 5 4	x_{12}	7 8 9 10 11 13 14 15
x_3	1 2 3 4 5 6 6	x_{13}	7 8 9 10 11 12 13 14 15 6
x_4	1 2 3 4 5 6 12	x_{14}	7 8 9 10 12 13 14 15 15
x_5	1 2 3 4 5 6 19	x_{15}	7 8 9 10 11 12 13 14 15 6
x_6	1 2 3 5 6 14	x_{16}	16 17 18 19
x_7	7 8 9 10 11 13 14 15	x_{17}	16 17 18 19
x_8	7 9 10 11 12 13 14 15	x_{18}	16 17 18 19
x_9	7 8 9 11 12 13 14 15	x_{19}	16 17 18 19
x_{10}	7 8 9 10 11 12 13 14		

level on the dendrogram, users could interactively change the granularity of objects defined based on the indiscernibility level, and (3) the method could handle non-Euclidean relational data in which asymmetry and local disturbance of the triangular inequality could occur.

5 Conclusions

In this paper, we presented a method for representing the granularity of data that have non-Euclidean relational properties. Asymmetric, relational similarity is translated into a binary classification for each object, and then the strength of the discrimination knowledge of binary classifications in ensemble is quantified by the indiscernibility between objects. Complete-linkage grouping is then applied based on the indiscernibility to build a dendrogram that represents hierarchy of granules. Using a simple synthetic dataset, we have demonstrated that the method could produce granules that meet the user-specified level of granularity, and could handle asymmetric dissimilarities. It remains as a future work to apply this method to other real-world data.

References

1. Romesburg, H.C.: Cluster Analysis for Researchers. Krieger Publishing Inc. (1989)
2. Pawlak, Z.: Rough Sets, Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
3. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley, Reading (2005)
4. Everitt, B.S., Landau, S., Leese, M.: Cluster Analysis Fourth Edition. Arnold Publishers (2001)
5. Hathaway, R.J., Bezdek, J.C.: NERF c-means: Non-Euclidean relational fuzzy clustering. Pattern Recognition 27(3), 429–437 (1994)
6. Neyman, J., Scott, E.L.: Statistical Approach to Problems of Cosmology. Journal of the Royal Statistical Society B20, 1–43 (1958)

Information Systems in Modeling Interactive Computations on Granules

Andrzej Skowron[1] and Piotr Wasilewski[2]

[1] Institute of Mathematics
Warsaw University
Banacha 2, 02-097 Warsaw, Poland
[2] Computational Inteligence Labratory
Department of Electrical and Computer Engineering
University of Manitoba
75A Chancellor's Circle, MB R3T 5V6 Winnipeg, Canada
skowron@mimuw.edu.pl, piotr@ee.umanitoba.ca

Abstract. In this paper we discuss the importance of information systems in modeling interactive computations performed on (complex) granules and propose a formal approach to interactive computations based on information systems. The basic concepts of information systems and rough sets are interpreted in the framework of interactive computations. We also show that information systems can be used for modeling more advanced forms of interactions such as hierarchical ones. The role of hierarchical interactions is emphasized in modeling interactive computations. Some illustrative examples of interactions used in the hierarchical multimodal classification method as well as in the ACT-R 6.0 system are reported.

Keywords: interactive computing, interactive systems, multi-agent systems, rough sets, granular computing, wisdom technology.

1 Introduction

The idea of interactive computing stems from many fields in computer science such as concurrent processes, non-terminating reactive processes (e.g. operating systems), distributed systems, distributed nets and objective programing (see [23], [24], [7]). The interaction paradigm is based on the idea of objects, in Artificial Intelligence often referred to as agents. Interaction is a form of computing which is performed by an agent interacting with the environment including possibly other agents. Therefore interactive systems can be composed of many objects. Every object takes inputs and produces outputs, in addition they can have internal states and also remember information about previous states. The crucial idea here is that interactive systems consist of one or many objects interacting with an environment that they cannot completely control. The main difference between agents and algorithms is in agents's flexibility [23,24] - through interaction with its environment agent can change the algorithms and as a consequence the way

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 730–739, 2010.

that computation is performed by taking into account new inputs coming during a computational process. Conversely, traditional algorithms are inflexible. Once computation with a given algorithm begins, it must be completed according to that algorithm regardless of new inputs that may arrive, possibly essential for an expected output. The concept of interaction should be clearly distinguished from those of parallelism (concurrency) and distribution [23]. The crucial point here is interaction: all algorithmic components of interactive systems, via its interface, interact with an unpredictive and uncontrolled external environment making the whole system interactive.

The idea of interactive computing is still in a developing stage and its foundations are not yet clarified. There are at least two main schools of thought, one pioneered by Peter Wegner [23,24,5] and another by Yuri Gurevich [6,7]. There is still no consensus between theoreticians on the statement that interactive systems are more powerful than classical algorithms and cannot be simulated by Turing machines. However, the idea of interactive computing still seems to be appealing from a practical point of view: interaction with or harnessing the external environment is inevitable to capture (and steer) behaviour of systems acting in the real world [15]. For unpredictive and uncontrolled environments it is impossible to specify the exact set of input states. In data mining or machine learning the most common case is when we start searching for patterns or constructing concepts on the basis of sample of objects since the whole universe of objects (data) is not known or it would be impractical to begin with the basis of a whole object universe.

Interactive systems have huge learning potential and are highly adaptive. Interacting algorithms can not only learn knowledge from experience (which is also done by classical non-interacting learning algorithms), they can change themselves during the learning process in response to experience. This property creates an open space for a new technology called Wisdom technology (Wistech) [8] and moreover for the case of intelligent agents this technology becomes inevitable. Intelligent agents make decisions during dynamic interactions within their environment. To meet this challenge they need to use complex vague concepts. In Wistech, wisdom is a property of algorithms, it is an adaptive ability of making correct judgments to a satisfactory degree in the face of real-life constraints (*e.g.*, time constraints) [8]. These decisions are made on the basis of knowledge possessed by an agent. Thus in Wistech, wisdom is expressed metaphorically by the so called *wisdom equation*:

$$wisdom = knowledge + adaptive\ judgment + interactions.$$

Adaptive ability means the ability to improve the judgment process quality taking into account agent experience. Adaptation to the environment on the basis of perceived results of interactions and agent knowledge is needed since *e.g.*, agents make decisions using concepts which are approximated by classification algorithms (classifiers) and these approximation are changed over time as a result of acting classifiers on variable data or represented knowledge. The wisdom equation suggests also another interaction of higher order: agents making decisions

based from ongoing experience, which is particular, apply possessed knowledge, which is general. Therefore making decisions in itself is a kind of logical interaction between general knowledge and particular experience. Vague concepts in this case help cover the gap between generality and particularity while Wisdom technology is required to improve decision making. In the last section we present ACT-R system taken from artificial intelligence and cognitive science as an example of a highly interactive complex granule.

2 Elements of Rough Set Analysis of Interactions

In this section we discuss interaction of granules relative to information systems. Rough sets, introduced by Zdzisław Pawlak [10,11,12], were intended to analyze information systems also called information tables. *An information system* is a triple $\mathcal{A} = \langle U, At, \{Val_a\}_{a \in At} \rangle$ where U is a set of objects, At is a set of attributes, and each Val_a is a value domain of an attribute $a \in At$, where $a : U \longrightarrow \mathcal{P}(Val_a)$ ($\mathcal{P}(Val_a)$ is a power set of Val_a). If $a(x) \neq \emptyset$ for all $x \in U$ and $a \in At$, then $\mathcal{A}$ is *total*. If $card(a(x)) = 1$ for every $x \in U$ and $a \in At$, then $\mathcal{A}$ is *deterministic*, otherwise $\mathcal{A}$ is *indeterministic*. It is worthwhile mentioning that information systems can be treated as a representation of result for agent interaction with the environment using condition attributes as part of the process of perception of an object's environment.

One of the key ideas in rough set theory is that knowledge is based on the ability to discern objects [11,12,14]. In a given information system $\mathcal{A} = \langle U, At, \{Val_a\}_{a \in At} \rangle$ this ability is presented by the *indiscernibility relation* $ind(B)$, where $B \subseteq At$ [11,14]. In particular, the indiscernibility relation $ind(B)$ can be interpreted as restricted to B perception history of an agent possessing a given set of attributes $B \subseteq At$. Indiscernibility relations play a crucial role in rough set theory providing a basis for reduction of information (elimination of attributes) and an approximation of concepts (subsets of the universe of objects).

For dealing with classification problems, decision information systems were distinguished [11,14]. Decision tables represent the result of agent interaction from perception using information systems with the human expert defined the decision attributes. Information systems were also used for representation of concurrent systems [13,17]. In this approach, attributes (columns) represent local concurrent processes, while objects (rows) represent global states of the system. A value of an attribute for a given object represents the state of a local process. Such representation makes analysis of whole concurrent systems possible. Interactions in this case are represented by rules over descriptors of information systems describing conditions of coexistence of local states within global states. Here decision attributes represent outputs of whole system, in particular, for actions taken by a system in its environment.

Theoretical implications of *cognitive architectures* [9,1] coming from cognitive science (see *e.g.* [22,20]) for the case of interactive computation, lead us to conclude that a given agent (represented by an object in an information system) can be also a coalition (collection) of interacting agents (granules). An illustrative

example is presented in Section 4. For this case, components of a granule as sensors or effectors can be viewed as processes interacting within an environment. A granule can also contain coordinating or controlling components which govern interactions between other its components or make decisions about actions in the environment based on these interactions, respectively. These components can be treated as coordinating or controlling processes respectively and have to be differentiated from processes responsible for storage of knowledge or information (*memory processes*). Therefore, complex granules can also represent concurrent/parallel systems. In this case, attributes represent also physical as well as logical sensors and effectors of the granule, *i.e.,* values of attributes represent the results of interaction between sensors and the physical environment.

The discussion above leads us to conclude that also behaviors of complex granules can be represented by means of decision information systems since they can be treated as (incomplete) specifications of concurrent/parallel systems. Since granule's effectors likely depend on its sensors as well as on the internal states, processes responsible for action steering or for sending messages should be represented by decision attributes divided into two disjoint sets of action steering or messages sending attributes respectively. Additionally, sensory processes (responsible for receiving messages and for perception of other stimuli) can be represented by condition attributes divided also into two respective sets. It should be noted that one can add new attributes created on the basis of existing ones. Also attribute value domains can include mathematical structures representing structures of objects (*e.g.* relational or algebraic structures see [18,8]). As a result, one can expand the set of attributes within an information system representing a complex granule. The new information system (with new attributes) is the result from interaction of a given information system with a granule representing a searching strategy for new attributes.

3 Hierarchical Granule Interactions

Granules can be interpreted logically [23,24], when their inputs and outputs are not physical sensors and effectors. An example of *logical* granules (objects) can be taken from a hierarchical approach to multimodal classification [19]. In this approach data models induced within classifier construction are often collections of multiple parts such that each piece explains only part of the data [19]. These parts can overlap or may not cover all of the data. To deal with the problems of overlapping and insufficient coverage, hierarchical or layered construction of a classifier is applied [19]. Instead of searching for a single, optimal model, a hierarchy of models is constructed under gradually relaxing conditions [19]. Overlapping granules can be understood as a kind of interaction between granules. Insufficient coverage can be seen as a result of interaction of a coalition of granules from a given level with its environment. In this approach a model from a higher level of hierarchy is constructed on the basis of models from the lower levels. This can be understood as hierarchical interaction between

granules or coalitions of granules from different levels. With this perspective, also coalitions consisting of granules from different levels are possible.

A hierarchical modeling of complex patterns (granules) in hierarchical learning (see *e.g.*, [3,19] can be described using the rough set approach based on information systems. In such description a construction of every model is described/made on the basis of a particular information system; with the result of construction of an information system from a given level of hierarchical modeling built from information systems from lower levels of its hierarchy. Let us consider two illustrative examples [18]. For any attribute we consider a relational structure $\mathcal{R}_a = (V_a, \{r_i\}_{i\in I})$. As examples of such structures one can consider $(V_a, =)$, $(V_a, \leq)$ and $(V_a, \leq, +, \cdot, 0, 1)$ taking $V_a = \mathbb{R}$, where $\mathbb{R}$ is a set of reals, or (V_a, τ), τ is a tolerance relation on V_a (*i.e.* τ is reflexive and symmetric). By L_a we denote a set of formulae interpreted over $\mathcal{R}_a$ as subsets of V_a while by $||\alpha||_{\mathcal{R}_a}$ a meaning (interpretation) of a formula $\alpha \in L_a$. So for every $\alpha \in L_a$, $||\alpha||_{\mathcal{R}_a} \subseteq V_a$. In the case of a particular information system $\mathcal{A} = \langle U, At, \{Val_a\}_{a\in At}\rangle$, $||\alpha||_{\mathcal{R}_a}$ for $a \in At$, can be used to define semantics of α over $\mathcal{A}$ by taking $||\alpha||_{\mathcal{A}} = \{x \in U : a(x) \in ||\alpha||_{\mathcal{R}_a}\}$, where $||\alpha||_{\mathcal{R}_a} \subseteq V_a$.

Relational structures corresponding to attributes can be fused. We present here an illustrative example from [18]. We assume that $\mathcal{R}_{a_i} = (V_{a_i}, r_{\mathcal{R}_{a_i}})$ are relational structures with the binary relation $r_{\mathcal{R}_{a_i}}$ for $i = 1, \cdots, k$. Their fusion is a relational structure over $V_{a_1} \times \cdots \times V_{a_k}$ consisting of a relation $r \subseteq (V_{a_1} \times \cdots \times V_{a_k})^2$ such that for any $(v_1, \cdots, v_k), (v'_1, \cdots, v'_k) \in V_{a_1} \times \cdots \times V_{a_k}$ we have $(v_1, \cdots, v_k) r (v'_1, \cdots, v'_k)$ if and only if $v_i r_{\mathcal{R}_{a_i}} v'_i$ for $i = 1, \cdots, k$. Intuitively, a vector $(v_1, \cdots, v_k)$ represents a set of objects possessing values $v_1, \cdots, v_k$ for attributes $a_1, \cdots, a_k$, respectively. Thus some vectors from $V_{a_1} \times \cdots \times V_{a_k}$ (not necessarily all) represent granules consisting of objects (some vectors from $V_{a_1} \times \cdots \times V_{a_k}$ correspond to the empty set). Therefore a relation r corresponds to a relation between granules. If $r_{\mathcal{R}_{a_i}}$ is a tolerance for $i = 1, \cdots, k$, then r is also tolerance relation.

In hierarchical modeling, object signatures at a given level of hierarchy can be used for constructing structural objects on the next level of hierarchy. (for an object object $x \in U$ the A-signature of x has the following form $Inf_A(x) = \{(a, a(x)) : a \in A\}$ where $A \subseteq At$). These structural objects are relational structures in which signatures are linked by relations expressing constraints for coexistence of signatures in relational structures.

Discovery of relevant attributes on each level of the hierarchy is supported by domain knowledge provided *e.g.* by concept ontology together with illustration of concepts by means of samples of objects taken from this concepts and their complements [3]. Such application of domain knowledge often taken from human experts serves as another example of interaction of a system (classifier) with its environment. Additionally, for the support of relevant attributes, discovery on a given level as well as on other levels of the hierarchy can be found using different ontologies. These ontologies can be described by different sets of formulas and possibly by different logics. Note that in a hierarchical modeling of relevant complex patterns also top-down interactions of higher levels of hierarchy with

lower levels should be considered, *e.g.*, if the patterns constructed on higher levels are not relevant for the target task the top-down interaction should inform lower levels about necessity of searching for new patterns.

4 Cognitive Architectures

The notion of cognitive architectures was proposed by Allen Newell [9]. Cognitive architecture (CA) is a computational form of unified theory of cognition (unified in the sense that it should unify psychological, neurobiological as well as computational aspects of various human cognitive performance and learning). Newell treats the mind as being *the control system that guides the behaving organism in its complex interactions with the dynamic real world* (see [9] p.43). He postulates that the central assumption of CAs should be that a human is a symbol/information processing system. Therefore the basic concepts and principles of intelligent systems as representation, knowledge, symbols and search apply to both humans and machines and they are central for CAs [9]. Since CA is about human cognition, it describes human behaviours, explains them in information processing terms, predicts and gives prescriptions for control of human behaviour. Cognitive architecture has three main constraints: it must be neurogically plausible, has to be a structure supporting mind-like behaviour (psychological experiments as well as computer simulations), and also take into account *real-time constraint on human cognition* [9]. Thus one can note that implemented CAs are real interactive systems. As the first example of CA, Newell points out the system ACT* proposed by John Anderson [1], and also presents his own system SOAR [9]. Here, as an example of CA, we present the system ACT-R (current version ACT-R 6.0), the successor of ACT*, introduced in [2] and discussed for example in [21].

The central idea of ACT-R (as well as ACT*) is a distinction between declarative and procedural memory. Thus ACT-R has two separate memory structures, a declarative one processing facts and a procedural one processing rules (IF-THEN structures also called productions). Memory in ACT*, ACT-R is goal directed, and as a result ACT-R 6.0 contains intentional module retrieving goals. ACT-R 6.0 adds two modules responsible for interaction with the external world; a visual and a manual module. As a consequence, the declarative memory structure (a production system implementing procedural memory) became one of the modules while the procedural memory structure became the central unit within the system. The ACT-R 6.0 system contains one central and four peripheral components (connected only to the central unit), two internal *i.e.* that are not connected directly to, or do not interact with the external environment (intentional module and declarative memory module) and two external modules, connected to and interacting with the external world (a visual module for perception and a manual module for acting in the environment). Procedural memory is connected to every peripheral module (and joining them together) but it communicates with modules through separate buffers. The presence of buffers makes communication of procedural memory with peripheral modules a more complicated interaction. Each buffer can contain only a piece of information at a given

moment. This is a constraint on information processing within ACT-R 6.0 and it represents some human information processing constraint (*e.g.* visual buffer represents selectiveness of human visual attention). Every module operates in a serial manner (*e.g.* a declarative memory module can retrieve only one item at a given moment) however modules within the system operate asynchronously and in parallel. Items retrieved in declarative memory, called *chunks*, represent declarative knowledge (propositions about facts). Items retrieved in procedural memory (productions) represent procedural knowledge (skills). ACT-R 6.0 architecture is activation based. Declarative memory has the form of a semantic web where chunks have different levels of *activation* which reflect their usage (chunks frequently used or chunks used recently have greater activation).

For every production rule there is an attached a real value provided by a *utility* function. Utility is calculated by the system on the basis of cost estimation (time needed for achieving the goal) and the estimate of achieving that goal if the production is chosen. ACT-R 6.0 (as its predecessors) is equipped with learning mechanisms. It has direct declarative learning where new chunks or associations created by productions have high activation to start with and if they are chosen frequently they maintain that high activation. New productions can be created. ACT-R 6.0 employs also learning mechanisms which update activations of chunks and utilities of productions. The parameters used to calculate utility based on experience are also constantly updated providing a continuous tuning effect (values of utilities are greater with use of their productions and smaller with disuse). Chunks and productions can be selected with some noise, but an item with the highest activation or utility has the greater probability of being selected even though other items can be chosen. This may produce errors but also enables ACT-R to explore evolving knowledge and strategies. Thus the learning mechanisms and noisy selections allow ACT-R 6.0 to interact dynamically with an environment, learn from experience and especially to harness the environment in achieving its goals. The ACT-R system can be used in multiagent simulations for steering the behaviour of a particular agent within a coalition working together where an ACT-R based agent interacts with other ACT-R based agents [20,21]. In addition, the ACT-R system is composed from mutually interacting components. These two things make hierarchical interactions within the ACT-R system possible. Therefore ACT-R can be treated as an example of a highly interactive complex granule which can be involved in hierarchical interactions.

5 Interactive Computations

In this section, the global states are defined as pairs $(s_a(t), s_e(t))$, where $s_a(t)$ and $s_e(t)$ are states of a given agent a and the environment e at time t, respectively. We now explain how the transition relation $\longrightarrow$ between global states are defined in the case of interactive computations. In Figure 1, the idea of transition from the global state $(s_a(t), s_e(t))$ to the global state $(s_a(t+\Delta), s_e(t+\Delta))$ is illustrated, where Δ is a time necessary for performing the transition, i.e., when $(s_a(t), s_e(t))$

$\longrightarrow (s_a(t+\Delta), s_e(t+\Delta))$ holds. $A(t)$, $E(t)$ denote the set of attributes available by agent a at the moment of time t and the set of attributes (sensors) used by environment e at time t, respectively. $Inf_{A(t)}(s_a(t), s_e(t))$ is the signature [14] of $(s_a(t), s_e(t))$ relative to the set of attributes $A(t)$ and $Inf_{E(t)}(s_a(t), s_e(t))$ is the signature of $(s_a(t), s_e(t))$ relative to the set of attributes $E(t)$. These signatures are used as arguments of strategies Sel_Int_a, Sel_Int_e selecting interactions I_a and I_e of agent a with the environment and the environment e with the agent a, respectively. $I_a \bigotimes I_e$ denotes the result of the interaction product $\bigotimes$ on I_a and I_e. Note that the agent a can have very incomplete information about I_e as well as the result $I_a \bigotimes I_e(s_a(t+\delta), s_e(t+\delta))$ only, where δ denotes the delay necessary for computing the signatures and selection of interactions (for simplicity of reasoning we assume that these delays for a and e are the same). Hence, information perceived by a about $s_a(t+\Delta)$ and $s_e(t+\Delta)$ can be very incomplete too. Usually, the agent a can predict only estimations of $s_a(t+\Delta)$ and $s_e(t+\Delta)$ during planning selection of the interaction I_a. These predictions can next be compared with the perception of the global state $(s_a(t+\Delta), s_e(t+\Delta))$ by means of attributes $A(t+\Delta)$. Note that $I_a \bigotimes I_e$ can change the content of the agent state as well as the environment state. Assuming that the current set of attributes $A(t)$ is a part of the agent state $s_a(t)$ this set can be changed, for example by adding new attributes discovered using I_a, for example with the help of hierarchical modeling discussed previously. Analogously, assuming that the description of the strategy Sel_Int_a is stored in the current state of the agent $s_a(t)$ this strategy can be modified as the result of interaction. In this way, sets of attributes as well as strategies for selecting interactions can be adopted in time.

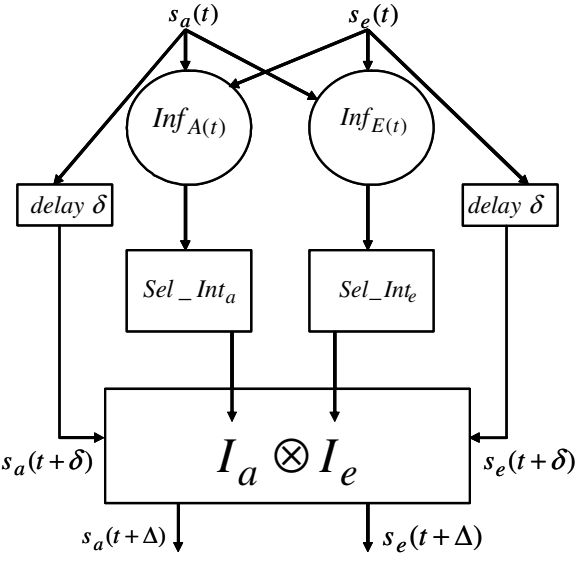


Fig. 1. Transition from global state $(s_a(t), s_e(t))$ to global state $(s_a(t+\Delta), s_e(t+\Delta))$

Computations observed by the agent a using the strategy Sel_Int_a in interaction with the environment e can now be defined with a help of the transition relation $\longrightarrow$ defined on global states and signatures of global states relative to the set of attributes of agent a. More formally, any sequence

$$sig_1, \ldots, sig_n, \ldots \tag{1}$$

is *a computation observed by a in interaction with e* if and only if for some t, Δ and for any i, sig_i is the signature of a global state $(s_a(t+i\Delta), s_e(t+i\Delta))$ relative to the attribute set $A(t+i\Delta))$ available by a at a moment of time $t+i\Delta$ and $(s_a(t+i\Delta), s_e(t+i\Delta)) \longrightarrow (s_a(t+(i+1)\Delta), s_e(t+(i+1)\Delta))$[1].

Let us assume that there is given a quality criterion over a quality measure defined on computations observed by the agent a and let sig_1 be a given signature (relative to the agent attributes). One of the basic problems for the agent a is to discover a strategy for selecting interactions (i.e., selection strategy) in such a way that any computation (e.g., with a given length l) observed by a and starting from any global state with the signature sig_1 and realized using the discovered selection strategy will satisfy the quality criterion to a satisfactory degree (e.g., the target goal of computation has been reached or that the quality of performance of the agent a in computation is satisfactory with respect to the quality criterion). The hardness of the selection strategy discovery problem by the agent a is due to the uncertainty about the finally realized interaction, i.e., the interaction being the result of the interaction product on interactions selected by agent a and the environment e. In planning the strategy, the agent a can use (a partial) information on history of computation stored in the state. One may treat the problem as searching for the winning strategy in a game between the agent a and the environment e with a highly unpredictable behavior.

6 Conclusions

This paper presents examples of interactions between different types of granules derived from information systems together with a proposal of a formal approach to interactive computations. A new reach class of interactions appears when we analyze interactions within time (*e.g.*, we take as values of attributes time series or their parts). This research is aimed in construction of a granule interaction based language for modeling of computations on information granules of different types. Such computations can result from *e.g.*, searching strategies for new properties or searching strategies for structures of interactions between processes discovered in data. This paper presents an introductory step towards this objective.

Acknowledgements. The research has been supported by the grants N N516 077837, N N516 368334 from Ministry of Science and Higher Education of the Republic of Poland, the Natural Sciences and Engineering Research Council of Canada (NSERC) grant 185986, Canadian Arthritis Network grant SRI-BIO-05, and Manitoba Centre of Excellence Fund (MCEF) grant T277.

[1] As usual one can consider finite and infinite computations.

References

1. Anderson, J.R.: The architecture of cognition. Harvard University Press, Cambridge (1983)
2. Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., Qin, Y.: An integrated theory of mind. Psychological Review 111, 1036–1060 (2004)
3. Bazan, J.: Hierarchical classifiers for complex spatio-temporal concepts. In: Peters, J.F., Skowron, A., Rybiński, H. (eds.) Transactions on Rough Sets IX. LNCS, vol. 5390, pp. 474–750. Springer, Heidelberg (2008)
4. Goldin, D., Smolka, S., Wegner, P. (eds.): Interactive Computation: The New Paradigm. Springer, Heidelberg (2006)
5. Goldin, D., Wegner, P.: Principles of interactive computation. In: [4], pp. 25–37 (2006)
6. Gurevich, Y.: Evolving Algebra 1993: Lipari Guide. In: Borger, E. (ed.) Specification and Validation Methods, pp. 9–36. Oxford University Press, Oxford (1995)
7. Gurevich, Y.: Interactive Algorithms 2005. In: [4], pp. 165–181 (2006)
8. Jankowski, J., Skowron, A.: Wisdom Technology: A Rough-Granular Approach. In: Marciniak, M., Mykowiecka, A. (eds.) Bolc Festschrift. LNCS, vol. 5070, pp. 3–41. Springer, Heidelberg (2009)
9. Newell, A.: Unified theories of cognition. Harvard University Press, Cambridge (1990)
10. Pawlak, Z.: Information Systems – theoretical foundation. Information systems 6, 205–218 (1981)
11. Pawlak, Z.: Rough sets. International Journal of Computing and Information Sciences 18, 341–356 (1982)
12. Pawlak, Z.: Rough sets. In: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, Dordrecht (1991)
13. Pawlak, Z.: Concurrent versus sequential – the rough sets perspective. Bulletin of the EATCS 48, 178–190 (1992)
14. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Science 177, 3–27 (2007)
15. Simons, H.: The Science of the Artificial, 2nd edn. MIT Press, Cambridge (1982)
16. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
17. Skowron, A., Suraj, Z.: Rough sets and concurrency. Bulletin of the Polish Academy of Sciences 41, 237–254 (1993)
18. Skowron, A., Szczuka, M.: Toward interactive computations: A rough-granular approach. In: Koronacki, J., Wierzchon, S., Ras, Z., Kacprzyk, J. (eds.) Commemorative Volume to Honor Ryszard Michalski, pp. 1–20. Springer, Heidelberg (2009)
19. Skowron, A., Wang, H., Wojna, A., Bazan, J.: A Hierarchical Approach to Multimodal Calssification. In: Ślęzak, D., Yao, J., Peters, J.F., Ziarko, W.P., Hu, X. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3642, pp. 119–127. Springer, Heidelberg (2005)
20. Sun, R. (ed.): Cognition and Multi-Agent Interaction. From Cognitive Modeling to Social Simulation. Cambridge University Press, Cambridge (2006)
21. Taatgen, N., Lebiere, C., Anderson, J.: Modeling Paradigms in ACT-R 29. In: [20], pp. 29–52 (2006)
22. Thagard, P.: Mind: Introduction to Cognitive Science, 2nd edn. MIT Press, Cambridge (2005)
23. Wegner, P.: Why interactions is more powerful than algorithms. Communications of ACM 40, 81–91 (1997)
24. Wegner, P.: Interactive foundation of computing. Theoretical Computer Science 192, 315–351 (1998)

Distributed Representations to Detect Higher Order Term Correlations in Textual Content

Pinar Öztürk, R. Rajendra Prasath*, and Hans Moen

Department of Computer and Information Science (IDI)
Norwegian University of Science and Technology (NTNU),
Sem Sælands Vei 7-9, NO - 7491, Trondheim, Norway
{pinar,rajendra}@idi.ntnu.no, hnsmoen@gmail.com,
http://www.idi.ntnu.no/people/pinar
http://www.idi.ntnu.no/~rajendra

Abstract. Case Based Reasoning(CBR), an artificial intelligence technique, solves new problem by reusing solutions of previously solved similar cases. In conventional CBR, cases are represented in terms of structured attribute-value pairs. Acquisition of cases, either from domain experts or through manually crafting attribute-value pairs from incident reports, constitutes the main reason why CBR systems have not been more common in industries. Manual case generation is a laborious, costlier and time consuming task. Textual CBR (TCBR) is an emerging line that aims to apply CBR techniques on cases represented as textual descriptions. Similarity of cases is based on the similarity between their constituting features. Conventional CBR benefits from employing domain specific knowledge for similarity assessment. Correspondingly, TCBR needs to involve higher-order relationships between features, hence domain specific knowledge. In addition, the term order has also been contended to influence the similarity assessment. This paper presents an account where features and cases are represented using a distributed representation paradigm that captures higher-order relations among features as well as term order information.

1 Introduction

Case-based reasoning (CBR) aims to store experiences, then to remember and reuse them when solving a similar new problem. A CBR system embeds a case base and a reasoning engine. Cases are knowledge units comprising at least two parts: a problem description and a corresponding solution. The reasoning engine conducts a search in the problem space that, in turn, provides links to the relevant solutions in the problem solving space. In conventional CBR, typically a frame-based representation language is used to represent cases making the ontological commitment that the world can be represented in terms of objects and attributes-value pairs. Acquisition of cases through manually crafting and structuring the attribute-value pairs constitutes the main reason why CBR systems have not been more common in industries. Manual case generation is a daunting and costlier task.

* This work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme.

M. Szczuka et al. (Eds.): RSCTC 2010, LNAI 6086, pp. 740–750, 2010.

This paper addresses CBR when both the new problem and past experiences are in textual format, meaning that the reasoning engine should assess the similarity of two problem descriptions represented in a natural language, hence *textual CBR*. Textual CBR (TCBR) investigates the representation and reasoning methods that will extend the applicability of CBR to situations where cases are inherently stored in free text format. More specifically, we investigate how to automatically derive the meaning of features and cases from an unannotated collection of case reports in free text format, without resorting to any other 'structured' knowledge source such as dictionaries or ontologies. TCBR is the subfield of CBR that employs CBR on textual cases.

Assessment of similarity between two cases is based on the similarity between the features that constitute the cases. It is, therefore, of vital importance to identify which features and what kind of information about them are necessary for a proper similarity judgment. Presence of lexically identical features in the two cases under scrutiny would obviously increase similarity of the cases. However, a term may have more then one sense (i.e., polysemous, e.g., one as a verb an another as noun) and similarity between two lexically identical features is conditioned on their property of having the same sense in both cases. On contrary, two lexically different features (i.e., synonymous) may have the same sense. Yet another situation is that two lexically different features may be related and their relationship can be explained by a chain of concepts. That is, they may have a higher-order relation. An effective similarity assessment mechanism is required to discover such relations. In conventional CBR these relations are captured in form of a domain specific knowledge base which is typically hand-crafted. CBR approaches that employ domain knowledge are called 'knowledge-intensive' [1] and are reported to increase the quality of similarity assessment [1,5].

Taken into consideration the complexity of natural languages, TCBR is faced with the puzzle of making an efficient search in a huge search space and ensuring at the same time a similarity assessment at a fair depth of the meaning of cases. The TCBR retrieval mechanism presented in this paper employs a special representation paradigm that is capable of capturing both feature co-occurence information, latent semantic relations between features and structural information such as term order. This is managed by employing random indexing [10,15] which is a distributed representation that reduces dimensionality implicitly and independently from the case collection/base, and allows evolution of the meaning of features incrementally.

Next section describes the importance of domain specific knowledge for case retrieval. Then in section 3, we illustrates random indexing that captures higher order feature relations. Section 4 describes the holographic reduced representations to capture feature order relations. Experimental results are discussed in section 5. Finally conclusion completes the paper.

2 Domain Specific Knowledge

Similarity between two cases is judged on the basis of similarity between the features that occur in these cases. Similarity between two features, in turn, is measured on the basis of the frequency of their co-occurrence in same cases. Features co-occurring in same cases are said to have first-order relations between them. However, first-order

relationships may not be sufficient for a thorough similarity assessment. A reason is that natural language is highly redundant and gives room for individual preferences with respect to word choices; two words may have the same meaning (i.e., synonymous) which may not be discovered merely by direct co-occurrence information because they hardly occur in the same incident report.

Similarity assessment is conjectured to involve higher-order relationships, particularly in models of analogical reasoning [7] and in problem solving in general. Motivated by these, knowledge-intensive CBR methods highly rely on domain specific knowledge. The example in Figure 1 illustrates how cases and general domain knowledge are connected. Note that the example is from conventional CBR where cases are structured as a set of attribute-value pairs, and the content of cases as well as the domain knowledge (shown in the upper half of the figure) are constructed by the knowledge engineer. Each type of link in the domain knowledge also has a weight, again ad hod defined. Similarity was calculated on the basis of presence or absence of features in a case as well as higher-order relations among features.

We investigate how higher-order relations can be discovered from textual data and used in the retrieval stage of TCBR. Features in the new case and a past case may look dissimilar in the first sight but render to be closely related, upon a deeper look into the related domain knowledge. For example, in Figure 1, **drug abuse** and **dental surgey** do not occur in the same patient case but both patients had *fever* which is the source of an indirect connection between these features. This is a higher-order relation determined by the path [*drug abuse, fever, dental surgey*] connecting the two features. The whole

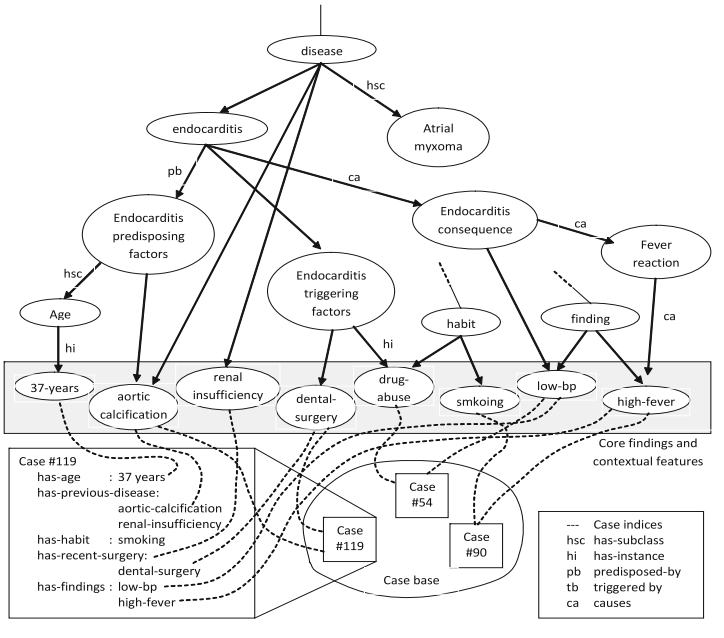


Fig. 1. Knowledge-intensive CBR (from [11])

story may be that situations of two patients in two cases may be similar in the context of, for example, endocarditis (an infectious heart disease) because both abuse of drug (where several addicts may use the same needle) and dental therapy may be sources of infection. Putting these correlations together, even when they cannot be explained from a true causal perspective, provide fortunate clues for perception of latent similarities. The cornerstone of the method lies in its representation model which is described in the next section.

3 Discovering Higher Order Relations

3.1 Distributed Representation of Meaning: Random Indexing

The vector space model of IR uses a local representation of features where each element of the vector alone represents the number of occurrences of the feature in a particular document. Random Indexing(RI), on the contrary, encodes a feature as a ternary vector with a predetermined length. Hence, each feature is initially encoded as a fixed length vector consisting of zeros and a small number of randomly distributed 1 and -1s, i.e., feature's *index vector*. RI is motivated by Johnson-Lindenstrauss Lemma [8] which asserts that a set of points in a high dimensional vector space can be mapped down into a reduced dimensional space such that the distance between any two points changes only insignificantly. Some examples of collections of high dimensional data that require compression can be found in the fields of audio, video, textual content and genome sequences. For more details on RI, the reader is referred to [15].

RI is an alternative to latent semantic indexing (LSI) that reduces dimensionality [4]. TCBR research has appreciated LSI's ability to discover higher-order distributional relations [3]. Despite both LSI and RI are capable of reducing dimensions and discovering higher-order relations, there are important differences between them. The most fundamental difference is that RI introduces a new representation paradigm for text representation. This characteristic, in turn, enables an implicit dimensionality reduction. LSI is known to be computationally expensive because it generates a huge feature-document matrix first and subsequently reduces the dimension applying single value decomposition (SVD) on this matrix. In LSI, latent dimensions are computed whereas in RI they are randomly selected in advance and are nearly orthogonal. Moreover, LSI is not incremental; it requires the availability of entire case collection when the process starts. In RI, reduction in dimensionality is done implicitly, by deciding the length of the representation vector. In addition, RI is incremental in the sense that meaning of a feature evolves simply by updating the feature's context vector when a case comprising that feature arrives at the case base.

3.2 Training of Feature Context Vectors in RI

The features are represented as vectors that accumulate information by training throughout the case corpus. The amount of information that a *feature context vector* represents is proportional to its interactions with other features in the domain/corpus. At time $t = 0$ the context vector is equal to the index vector of the feature. When the whole corpus is scanned, the context vector of the feature will represent what is learned

about the behavior of the feature. *Superposition*, i.e., vector addition, is used when updating the context vector. Adding two vectors x and y, the superposition of these is vector z where $z = x + y$. The cosine similarities between x and z, and between y and z will be high.

For each occurrence of a given feature in all cases, we focus on a fixed window of size $(2 * k) + 1$ centered at the given feature (e.g., [14] suggests 5 as the window size). Then feature context vector for *feature i* is computed using the following equation:

$$C_{feature_i} = C_{feature_i} + \sum_{j=-k;j\neq 0}^{+k} I_{feature_{(i+j)}} \times \frac{1}{d^{|j|}} \tag{1}$$

where $\frac{1}{d^{|j|}}$ is the weight proportion with respect to the size j of the window ($d = 2$ in this work).

Let us assume that we have the new case: "The fisherman caught a big salmon today", window size k is equal to two, and we are training the feature **big**. Our windowed sentence for the feature *big* looks like this:

The, [*fisherman, caught,* **big**, *salmon, today*].

The feature-context-vector C_{big} for **big** becomes now:

C_{big}=C_{big}+(0.25 $\times$ $I_{fisherman}$)+(0.5 $\times$ I_{caught})+(0.5 $\times$ I_{salmon})+(0.25 $\times$ I_{today})
Meaning of a case is captured in the collective representation of the constituent features. A case is also represented as a vector of the same length as the feature index and feature context have, and the case vector is computed as a weighted superposition of context vectors of features that occur in the case. The **case-context-vector**, representing the meaning of *case* is computed based on the feature-context-vectors of the features that constitute it. It is simply:

$$C_{case} = \sum_{i=1} f_i \times C_{feature_i} \tag{2}$$

where f_i is the number of occurrences of $feature_i$ in *case*.

4 Learning Term Order Information - Holographic Reduced Representations

It has been contended that merely co-occurence information is not enough to unreveal similarities. Quoting Firth [6], "you shall know a word by both the company it keeps and how it keeps it". In LSI, structural information (e.g., term order) is not is captured. The same applies to RI. Similar to learning of semantic behavior, word order information also can be learned in an unsupervised way, using *Holographic Reduced Representations*(HRR) which have been extensively used in image and signal processing [12]. It was also used in cognitive psychology to model mental lexicon [9] and analogy-based reasoning.

The representation of features in HRR is the same as in RI while the updating of feature vectors is different. In addition to superposition, HRR employs another vector operation, called *circular convolution* (depicted by $\circledast$) which is a multiplicative operation that enables the association (i.e., 'binding') of two or more different types

of information about a feature. In circular convolution the length of the vector after circular convolution does not change. Circular convolution $z = x \circledast y$ of two vectors x and y are computed according to the following equation [12]:

$$z_j = \sum_{k=0;}^{n-1} x_k \times y_{j-k} \quad for\ j = 0\ to\ n-1 \tag{3}$$

In this way, the convoluted vector will never increase in size over time, making it ideal to use in a vector model representation. Circular convolution has two special properties that are important for the work presented in this paper. First, the convolved vector is near orthogonal to its component vectors meaning that it is a new vector not similar to its component vectors. Adding and multiplying vectors capture two different types of information. When two vectors are convolved, the result is a vector containing a compressed version of both parent vectors. It is important to note that the product vector from circular convolution has no similarities to its parent vectors when it comes to cosine similarity. For example, if we convolve the index vectors of big, I_{big}, and of salmon, I_{salmon} , we obtain

$V_{big,salmon} = I_{big} \circledast I_{salmon}$ and
$\text{sim}(V_{big,salmon}, I_{big}) \approxeq 0$, and $\text{sim}(V_{big,salmon}, I_{salmon}) \approxeq 0$

While superposition of index vectors of these 'big' and 'salmon' encodes the information, these two words co-occur in the same context, convolution encodes that features 'big' and 'salmon' occur consecutively. The second property of circular convolution relates to its multiplicative character; it is commutative, which means:

$$I_{big} \circledast I_{salmon} = I_{salmon} \circledast I_{big}$$

In other words, we are not capturing the correct order information, only that these two features are located next to each other. That is, 'blind venusian' and 'venusian blind' would be encoded by the same convolved vector. Plate [12] suggests a method to solve this problem, which Jones and Mewhort [9] applied: It is possible to acquire both non-commutative and non-associative vectors by using different index vectors when a feature is located to the left or right of the targeted feature in the sentence. We adopt the same approach where, in the first place, each feature has two index vectors, one is used when a feature is located on the left side of the feature that is being trained ($I_{Lfeature}$), and one for the right side ($I_{Rfeature}$). A placeholder vector Φ is used to represent the trained feature. Assume that we have the following two sentences: "blind venusian" and "venusian blind". and we are training *blind* in each. Our convolutions will look like this:

"blind venusian" $\rightarrow \Phi \circledast I_{Rvenusian}$, and "venusian blind" $\rightarrow I_{Lvenusian} \circledast \Phi$

The resulting product vectors will have zero similarity (cosine similarity) to each other.

Window of a certain size was used during the training of feature context vectors. The same window is used for convolution (except we are not weighting the neighboring features). In order to capture all term order information located in a window, multiple convolutions are performed, and then the product vectors are added into the feature's order vector using superposition. When window size (and sentence) is larger than two in length, convolution is done in an $n-$gram fashion. The following example (using 'fisherman sentence') demonstrates how $n-$gram works for training feature "big":

[caught, a, **big**, salmon, today]

In order to encode all term order information located in this window into **big**'s order vector, we have to find all combinations in the sentence that includes the feature **big**:

[a, **big**], [**big**, salmon] , [caught, a, **big**] , [a, **big**, salmon] , [**big**, salmon, today] , [caught, a, **big**, salmon] , [a, **big**, salmon, today] , [caught, a, **big**, salmon, today]

All together 8 different n-gram vectors are created from the windowed sentence, and all these vectors are added to **big**'s order vector:

$(I_{La} \circledast \Phi)$, $(\Phi \circledast I_{Rsalmon})$, $(I_{Lcaught} \circledast I_{La} \circledast \Phi)$, $(I_{La}\ \Phi \circledast I_{Rsalmon})$, $(\Phi \circledast I_{Rsalmon} \circledast I_{Rtoday})$, $(I_{Lcaught} \circledast I_{La} \circledast \Phi \circledast I_{Rsalmon})$, $(I_{La} \circledast \Phi\ I_{Rsalmon} \circledast I_{Rtoday}))$, $(I_{Lcaught} \circledast I_{La} \circledast \Phi \circledast I_{Rsalmon} \circledast I_{Rtoday}))$

This is done for all occurrences of **big** in the entire corpus, and everything is then added to **big**'s order vector (i.e., O_{big}) using superposition. This process is repeated through the entire case collection.

The case vector is computed according to the following equation:

$$O_{case_j} = \sum_i f_i \times O_{feature_i} \tag{4}$$

where $O_{feature_i}$ depicts order vector of feature i that occur in the case j.

Similarity between two cases are then found by computing cosine between them.

5 Results and Discussion

In this section, we will compare the results of conventional bag of words with the results of random indexing and HRR methods. Our evaluation method is similar to the one given in [13]. To show the effectiveness of retrieved textual cases for the given problem description, we correlate the retrieved cases with rank aggregated score over two data sets. We use cosine similarity to measure pairwise similarity between the given new case and cases in the case base.

5.1 Dataset

In these experiments, we considered two different data sets: *Reuters 21578*: top 10 classes of Reuters 21578; and *TASA*: all 9 classes each with 100 documents. Reuters 21578 newswire data[1] is often used in the evaluation of text categorization. This collection contains 21,578 newswire articles (hence the name) in English. Each document, in this collection, was tagged by a human indexer with class label(s) that fell into five categories: TOPICS (119), PLACES (147), PEOPLE (114), ORGS (32), EXCHANGES (32)[the value inside brackets shows the number of subclasses]. We have omitted documents that contain empty body text. From this collection, we have taken top 10 categories, namely *acq*, *uk*, *japan*, *canada*, *west-germany*, *grain*, *crude*, *earn*, *usa* and *money-fx*, of Reuters 21578 dataset, each having randomly chosen 100 cases and formed Reuters1000 dataset for our experiments.

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578/

TASA dataset[2] consists of unmarked high school level English text. There are 37, 600 text documents arranged in 9 categories: *Business*, *Health*, *HomeEconomics*, *IndustrialArts*, *LanguageArts*, *Science*, *SocialStudies*, *Miscellaneous* and *Unspecified*. Due to memory limitations during the training phase of feature context vectors, we have considered 100 documents from each category and formed TASA900 dataset for our experiments. During preprocessing, we considered only the text portion of a document by ignoring all markups, tags and special symbols. Stop words are removed using the SMART stop words list[3]. The features are not stemmed in our experiments.

5.2 Evaluation Methodology

In our experiments, we randomly split the Reuters-21578 dataset into two parts, one split with 60% for training and another with 40% for testing. Each document is considered as a case. For each feature, we obtained a feature context vector as described in section 3.2. We use the following feature vector weighting schemes similar to [2,16]:

Case feature weights(W_{c_i}):

$$\frac{w_{c_i}}{\sqrt{\sum_{i=1}^{m} w_{c_i}^2}} \tag{5}$$

where w_{c_i} is superposition of feature context vectors, each multiplied with its frequency, in the case c_i. Query(new case) feature weights(W_{q_i}):

$$\frac{w_{q_i}}{\sqrt{\sum_{i=1}^{m} w_{q_i}^2}} \tag{6}$$

where w_{q_i} is the superposition of feature context vectors, each multiplied with its frequency, in the query case q_i. Similarity between the query (new case) and the case in the case base is computed by :

$$sim(q_i, c_i) = \sum_{\text{matching features}} W_{q_i} \times W_{c_i} \tag{7}$$

During the training of feature context vectors, the values of the vector increases with the number of occurrences in the case collection. In such situations, we could apply vector length normalization. At the same time, any normalization factor has an effect of decreasing weight of the document features thereby reducing the chances of retrieval of the document. Therefore, higher the normalization factor for a document, the lower are the chances of retrieval of that document[16]. Thus, depending upon size of the data set, suitable normalization factor may be chosen. In this work, we perform vector normalization using $\overrightarrow{V}_{norm(t_i)} = \overrightarrow{V_{t_i}}/sum(abs(\overrightarrow{V_{t_i}}(c)))$. Applying normalization at the end is not a fair idea whereas progressive normalization suits as the better way.

In our experiments, given a new case, we have extracted top cases using bag of words, random indexing and HRR approaches. Then from the retrieved case list,

[2] Owned by Pearson Knowledge Technologies at the University of Colorado.

[3] ftp://ftp.cs.cornell.edu/pub/smart/english.stop

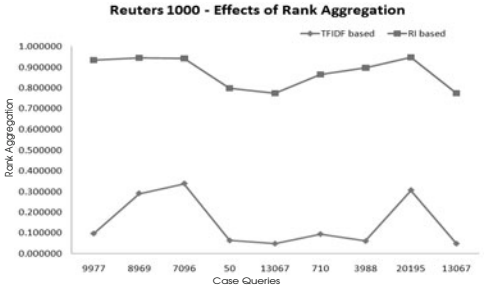


Fig. 2. Effects of rank aggregation of top k best matching cases of Reuters 1000

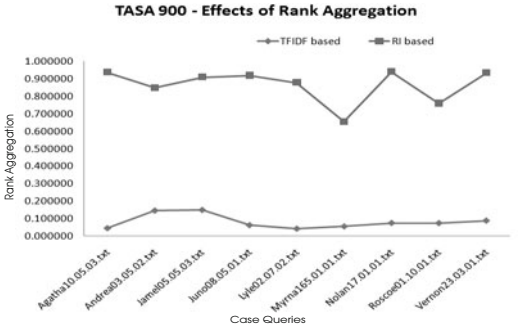


Fig. 3. Effects of rank aggregation of top k best matching cases of TASA 900 with TFIDF and RI methods

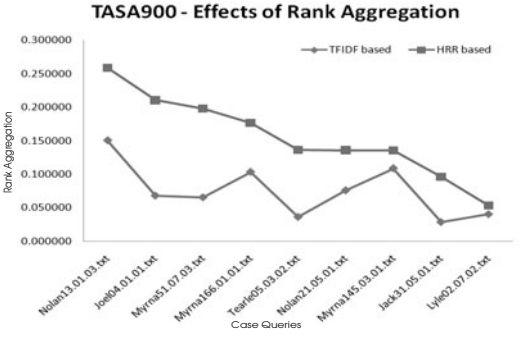


Fig. 4. Effects of rank aggregation of top k best matching cases of TASA 900 with TFIDF and HRR methods

rank of all k top cases are aggregated with respect to their actual similarity scores and the results are compared. Figure 2 illustrates TFIDF and RI results for Reuters 1000 dataset while Figure 3 illustrates the similar analysis for TASA 900 dataset. In

both, it can be seen that RI outperforms TFIDF approach. In Figure 4 we present the comparison between TFIDF and HRR. We should make it clear that in the experiments, 'HRR' is imposed on the TFIDF results, that is, in Figure 4 'HRR' includes the TFIDF and term order information. It can also be noted that HRR also performs better then TFIDF meaning that term order information improves the TFIDF results. However, RI alone seems to better capture the case meanings than TFIDF + order information does.

6 Conclusion

The paper presented a retrieval account for TCBR. Employing the distributed representation of Random Indexing, it enabled the dimension reduction in an effective way, doing it implicitly, not as a separate stage as in LSI. The method takes into consideration both semantic and structural properties of features. We need to investigate how the information about the two types of feature behaviors (i.e., discovered by RI and HRR, respectively) can be combined in a sensible way so that similarity between cases can be assessed properly. In addition, to explore the effects in depth it is essential to apply RI and HRR on larger data sets.

References

1. Aamodt, A.: Knowledge-intensive case-based reasoning in creek. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 1–15. Springer, Heidelberg (2004)
2. Buckley, C., Salton, G., Allan, J., Singhal, A.: Automatic query expansion using smart: Trec 3. In: TREC (1994)
3. Chakraborti, S., Mukras, R., Lothian, R., Wiratunga, N., Watt, S., Harper, D.: Supervised latent semantic indexing using adaptive sprinkling. In: Proc. of the 20th Int. Joint Conf. on AI, pp. 1582–1587. Morgan Kaufmann, San Francisco (2007)
4. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. JASIS 41(6), 391–407 (1990)
5. Díaz-Agudo, B., González-Calero, P.A.: Cbronto: A task/method ontology for cbr. In: Proc. of 15th Int. Florida AI Research Society, pp. 101–105. AAAI Press, Menlo Park (2002)
6. Firth, J.R.: A synopsis of linguistic theory, 1930-1955. In: Studies in Linguistic Analysis, pp. 1–32 (1957)
7. Gentner, D., Forbus, K.D.: MAC/FAC: A model of similarity-based retrieval. Cognitive Science 19, 141–205 (1991)
8. Johnson, W., Lindenstrauss, L.: Extensions of lipschitz maps into a hilbert space. Contemporary Mathematics 26, 189–206 (1984)
9. Jones, M.N., Mewhort, D.J.K.: Representing word meaning and order information in a composite holographic lexicon. Psychological Review 114, 1–37 (2007)
10. Kanerva, P., Kristofersson, J., Holst, A.: Random indexing of text samples for latent semantic analysis. In: Proc. of the 22nd Annual Conf. of the Cognitive Science Society, pp. 103–106. Erlbaum, Mahwah (2000)

11. Öztürk, P., Aamodt, A.: A context model for knowledge-intensive case-based reasoning. Int. J. Hum.-Comput. Stud. 48(3), 331–355 (1998)
12. Plate, T.: Holographic reduced representations. IEEE Transactions on Neural Networks 6(3), 623–641 (1995)
13. Raghunandan, M.A., Wiratunga, N., Chakraborti, S., Massie, S., Khemani, D.: Evaluation measures for tcbr systems. In: Althoff, K.-D., Bergmann, R., Minor, M., Hanft, A. (eds.) ECCBR 2008. LNCS (LNAI), vol. 5239, pp. 444–458. Springer, Heidelberg (2008)
14. Sahlgren, M.: Vector-based semantic analysis: Representing word meanings based on random labels. In: ESSLI Workshop on Semantic Knowledge Acquistion and Categorization. Kluwer Academic Publishers, Dordrecht (2001)
15. Sahlgren, M.: An introduction to random indexing. In: Methods and Applications of Semantic Indexing Workshop at 7th Int. Conf. on Terminology and Knowledge Eng., TKE 2005 (2005)
16. Singhal, A., Salton, G., Mitra, M., Buckley, C.: Document length normalization. Inform. Process. Manage. 32(5), 619–633 (1996)

Author Index