Krzysztof A. Cyran
Stanisław Kozielski
James F. Peters
Urszula Stańczyk
Alicja Wakulicz-Deja (Eds.)

Man-Machine Interactions

Springer

Advances in Intelligent and
Soft Computing 59

Editor-in-Chief: J. Kacprzyk

Advances in Intelligent and Soft Computing

Editor-in-Chief

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Krzysztof A. Cyran, Stanisław Kozielski,
James F. Peters, Urszula Stańczyk,
Alicja Wakulicz-Deja (Eds.)

Man-Machine Interactions

Springer

Editors

Dr. Krzysztof A. Cyran
Institute of Informatics
Silesian University of Technology
Akademicka 16
44-100 Gliwice
Poland

Dr. Urszula Stańczyk
Institute of Informatics
Silesian University of Technology
Akademicka 16
44-100 Gliwice
Poland

Prof. Stanisław Kozielski
Institute of Informatics
Silesian University of Technology
Akademicka 16
44-100 Gliwice
Poland

Prof. Alicja Wakulicz-Deja
Institute of Informatics
University of Silesia
Będzińska 39
41-200 Sosnowiec
Poland

Prof. James F. Peters
Department of Electrical and
Computer Engineering
University of Manitoba
75A Chancellor's Circle
Winnipeg, MB R3T 5V6
Canada

To the memory of Professor Adam Mrózek

Preface

The International Conference on Man-Machine Interactions (ICMMI 2009) commemorates the life and work of Adam Mrózek (1948-1999). From 1970, he worked at the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, and from 1992 at the Institute of Informatics, Silesian University of Technology, Gliwice, Poland, in 1998 taking the post of a Vice Director of the Institute for Research and the head of the Computer Systems Theory and Design Division. During his lifetime the scientific interests of A. Mrózek were highly diversified.

This ICMMI 2009 volume reflects a number of research streams that were either directly or indirectly influenced by Adam Mrózek's work on the development of computer systems and software that makes it possible to employ them in a variety of human activities ranging from logic studies and artificial intelligence[1], rule-based control of technological processes[2], image analysis[3], expert systems and decision support[4], to assistance in creative works.

In particular, this ICMMI volume points to a number of new advances in man-machine communication, interaction between visualization and modeling, rough granular computing in human-centric information processing and the discovery of affinities between perceptual granules. The topical subdivisions of this volume include human-computer interactions, decision support, rough

---

[1] Jankowski, A., Skowron, A.: Logic for artificial intelligence: A Rasiowa-Pawlak school perspective. In: Ehrenfeucht, A., Marek, V.W., Srebrny, M. (eds.) Andrzej Mostowski and Foundational Studies, pp. 92–105. IOS Press, Amsterdam (2008).

[2] Mrózek, A.: Rough sets in computer implementation of rule-based control of industrial processes. In: Słowiński, R. (ed.) Intelligent decision support–Handbook of applications and advances of the rough sets theory, pp. 19–32. Kluwer, Boston (1992); Mrózek, A.: A new method for discovering rules from examples in expert systems. International Journal of Man-Machine Studies 36, 127–143 (1992)

[3] Mrózek, A., Płonka, L.: Rough sets in image analysis. Foundations of Computing and Decision Sciences 18(3-4), 258–273 (1993).

[4] Mrózek, A., Skabek, K.: Rough rules in Prolog. In: Półkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS, vol. 1424, pp. 458–466. Springer, Heidelberg (1998).

fuzzy investigations, advances in classification methodology, pattern analysis and signal processing, computer vision and image analysis, advances in algorithmics, databases and data warehousing, and embedded system applications. These Proceedings present seventy-one research papers reflecting the work by 117 researchers from ten countries, namely Canada, Chech Republic, P.R. China, Egypt, India, Norway, Poland, Saudi Arabia, Thailand, and USA.

This volume has been made possible thanks to the laudable efforts of a great many organizations that have made this conference possible, namely, Institute of Informatics, Silesian University of Technology, Gliwice, Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice, Institute of Computer Science, Polish Academy of Sciences, Warsaw, and Institute of Computer Science, University of Silesia, Katowice. The organization of the conference benefited also from contributions by Robert Budryk, Ewa Piętka, Henryk Rybiński, Michał Woźniak, and other generous persons.

Let us express our gratitude to James F. Peters who was the initiator of the whole idea and kindly accepted the invitation to serve as the Honorary Chair, and to deliver a keynote talk during the conference.

We also wish to express our thanks to Marek Kimmel, Andrzej Skowron and Ryszard Tadeusiewicz for accepting our invitation to be keynote speakers.

In addition, the editors and authors of this volume extend an expression of gratitude to Thomas Ditzinger, Dieter Merkle, Heather King and other staff at Springer for their support in making this volume possible. Furthermore, the editors extend their thanks to Janusz Kacprzyk and other members of the Programme Committee for helping in the compilation of this volume, and to Sebastian Deorowicz for extensive use of his typsetting skills.

The editors express their hopes that this volume and the ICMMI 2009 conference will commemorate Adam Mrózek not only by just reporting scientific and technological solutions which have already been achieved, but also by inspiring some new interdisciplinary efforts leading to further improvements and research on man-machine interactions, resulting in enhancing the quality of life for all of us.

Gliwice   Krzysztof A. Cyran
September 2009   Stanisław Kozielski
                                                                              James F. Peters
                                                                             Urszula Stańczyk
                                                                     Alicja Wakulicz-Deja

Contents

Part III: Computational Techniques in Biosciences

Part VI: Advances in Classification Methods

Part IX: Advances in Algorithmics

Part X: Databases and Data Warehousing

Part XI: Embedded Systems Applications

Part I
Keynote Talks

Speech Man-Machine Communication

Ryszard Tadeusiewicz and Grażyna Demenko

Abstract. For modern man-machine communication traditional methods based on a keyboard and a mouse are definitely insufficient. Especially thinking about poor and bad-qualified citizens of Information Society we must try to find the easiest and most comfortable method for man-machine communication. This optimal method of communication going from a man to a machine is (or should be) speech communication. In the paper some general remarks about speech man-machine communication are presented and some problems connected with this area of technological activity discussed. On the basis of such discussion the survey of speech recognition systems is shown and compact analysis of the most important and most up-to-date scientific and technological problems related to such systems is presented. Final conclusions are directed both toward presentation of existing speech recognitions systems as well as toward forecasting of solutions, which will be available in the near future.

Keywords: man-machine interactions, man-machine communication, automatic speech recognition, automatic language communication.

1 Introduction

Thinking about Man-Machine Interactions development, we must first improve Man-Machine Communication. Without effective communication every form of interaction will be crippled. On the other hand the communication between an

Ryszard Tadeusiewicz
Department of Automatics, AGH University of Science and Technology,
Mickiewicza Av. 30, 30-059 Cracow, Poland
e-mail: rtad@agh.edu.pl
http://www.tadeusiewicz.pl

Grażyna Demenko
Phonetic Department, SpeechLab, Adam Mickiewicz University,
Rubież 46, 61-612 Poznan, Poland
e-mail: lin@amu.edu.pl
http://www.speechlabs.pl

arbitrary technical system, shortly called here a machine, and a human being (man) must be discussed separately for the problem of communication from the machine to the man and communication initiated by the man and directed towards the machine (Fig. 1).
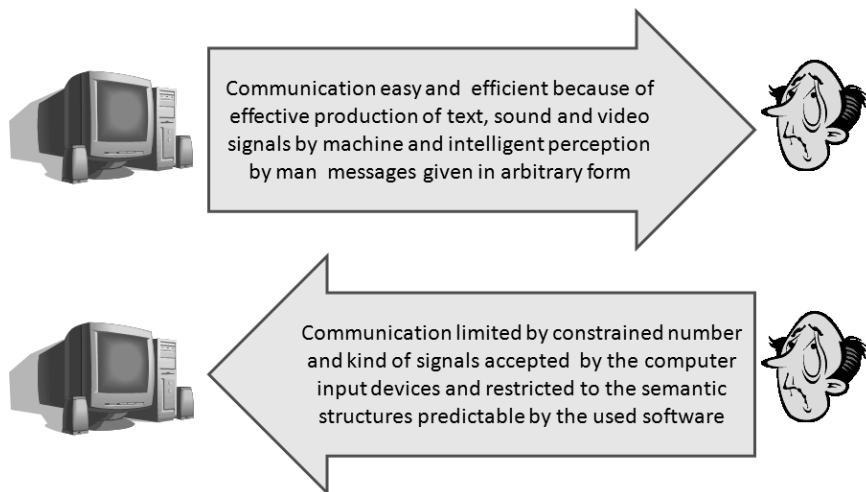
Fig. 1 Asymmetry in man-machine communication

The first kind of communication, routing form machines to the people, is much easier and quite well developed. Computers can produce information in the form of numerical data and text messages, as well as good quality images. This last form of communication from the machine to the man is particularly popular because people like to receive information in the graphical form. This type of communication is especially useful when the information has complicated structure and the user must perform analysis of complex relations between data elements instead of perception of the data itself. Tasks of this type are performed easier and faster when the information is given in the graphical form because of the developed structure and perfect functioning of human visual systems. Moreover, when visual communication is not preferred (for example during communication from some GPS system to the driver who must observe the way) we have an effective and convenient method of speech communication from the machine to the man producing either synthetic speech signals or using natural speech signal samples (Fig. 2).

The communication in the opposite way is much more complicated (Fig. 3). Thinking about communication from the man to the machine we obviously take into account a keyboard, a mouse, a touch screen, a control panel or sometimes some special devices, e.g., a joystick. For most people it is enough. For well educated and familiar with computers users, GUI (Graphic User Interface) is the favorite technology in man-machine communication and they do not need anything else. But for
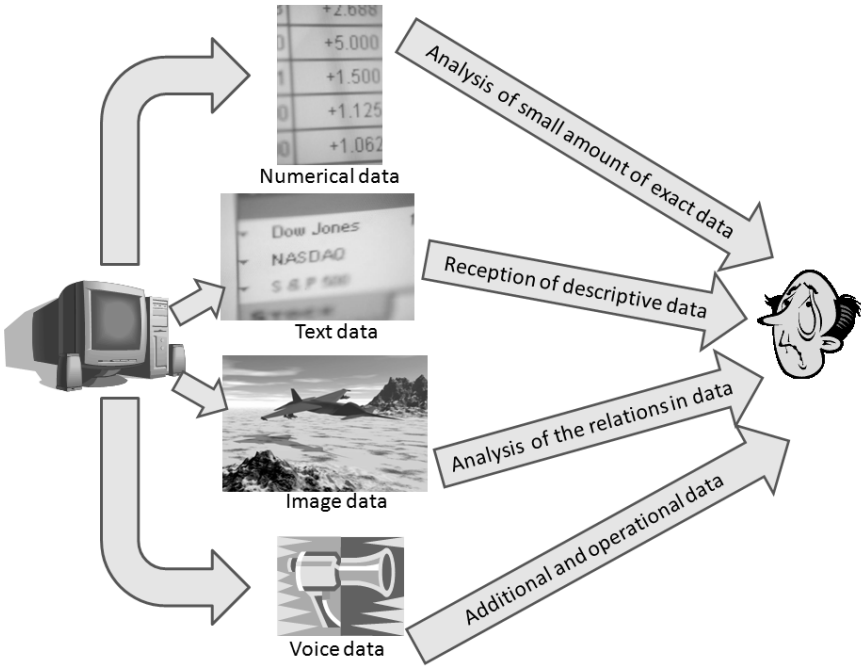
Fig. 2 Different methods used for communication from machine to man
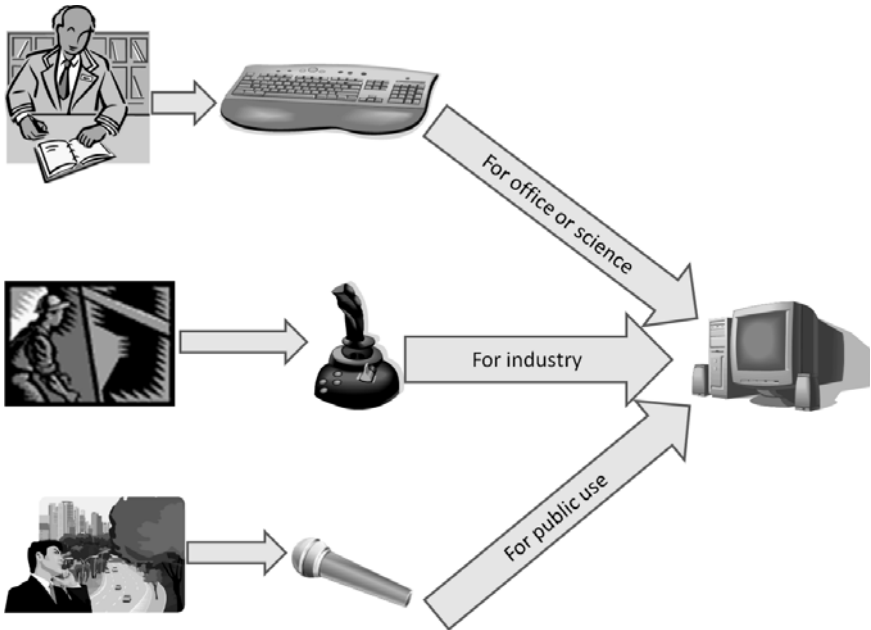
Fig. 3 Different methods used for communication from man to machine

other users this model of communication is too complicated, not natural and intuitive enough and in general not comfortable. Therefore for sustainable development of the Information Society, as well as for avoiding digital division or digital exclusion of phenomena we need better tools for man-machine communication. This tool may be a system for automatic speech recognition.

2   Advantages of Speech Communication

For most people speech is the most natural and most comfortable method of communication. In case of control systems speech communication offers free-hand control, which is sometimes very important. This method of man-machine communication is particularly convenient for disabled persons or for ill patients guarantying them (when using some special robot systems for the whole day care) almost full independence and self-service. Moreover, the proposed speech control is fully effective also in darkness or when the user is in an uncomfortable position as well as can be (in special cases) under mechanical shocks, vibrations and G-force. Last but not least we must also emphasise that such form of communication is for almost every human being easy to learn and very natural in use.

Let us remind that using speech signal as a information carrier between a man and machines we can also use all devices designed for speech communication between people (e.g., phones, both stationary and mobile), which can be used as the cheapest and easiest form of remote control. Moreover, in case of emergency the vocal reaction of almost every human being can be much faster and more precise than any other forms of communication and steering manipulators.

3   Speech Synthesis and Generation

Speech communication in man-machine interaction can be performed in both directions: from the machine to the man and in the opposite direction. It is definitely easier to produce some vocal output directed from the machine to the man. Everybody know such solutions used contemporarily in many popular systems, among other in various call centers and in car GPS devices. Vocal output produced by the machine can be obtained in two ways, the first of which is the artificial speech synthesis. This method is based on the exploitation of some predefined set of natural voice samples. Available now comprehensive vocal databases together with fast methods of suitable speech element selection and efficient methods of speech segment coupling allow to generate artificial speech signals of really good quality by glueing together some predefined elements like phrases, words, syllables and also phonemes (Fig. 4).

Such 'reconstructed form elements' speech output can be formed very easily and guarantees good quality of speech and acceptance from the audience. However, this method of speech synthesis is limited to some predictable utterances, with elements
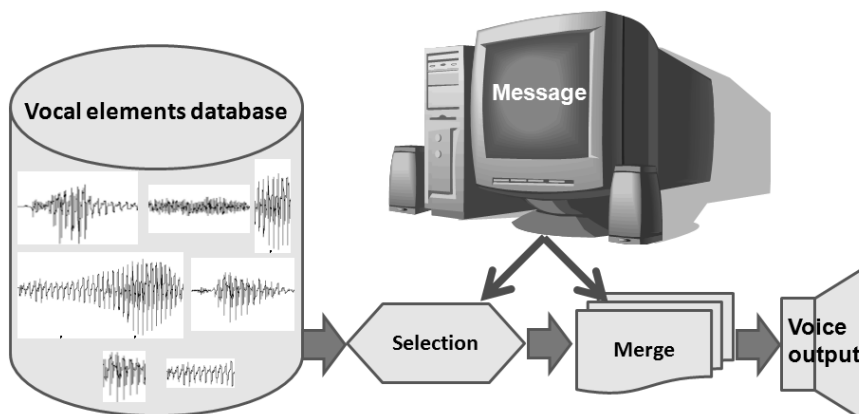
Fig. 4 Simplified scheme of the speech synthesis process

prepared before use. When we need a wide range of different speech communicates the preferred way of artificial speech production is its generation. During the speech generation process we try to simulate by a computer the natural speech production process, with simulation of pitch sound generator (artificial larynx and vocal fold activity), noise sound sources (when necessary) and next articulation and modulation processes (with simulation of tongue, palate, nasal cavity and lips activity). The last and the most difficult element of the structure under consideration is the artificial control of the whole speech production process in a similar way as it is performed in a natural human vocal tract (Fig. 5).

Speech generation by the machines and speech synthesis for many purposes is now fully developed technology and scientific research in this area can be considered as exhausted.

4 Polish Speech Recognition as an Open Scientific Problem

In contrast to this situation the opposite direction communication, e.g., automatic recognition of the speech by the machines – is much more complicated and not fully solved problm til now, both from the scientific and practical point of view. At the same time a vocal input to the machine, which must be based on automatic speech recognition is needed for many purposes: for vocal communication with many devices, for speech input during entering many kinds of data to many kinds of information systems, for speech control of robots and other automatic systems, for speech interactions between a user and many automatic service centers, e.g., phone-banking etc.

Moreover, existing and good working solutions, used for particular languages (English, Japanese, German) cannot be easily adapted for other languages,
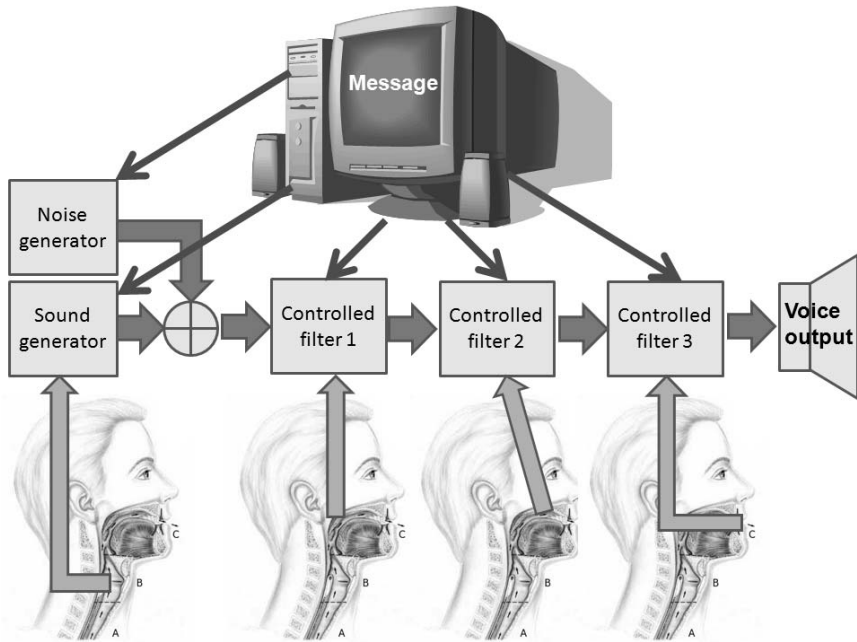
Fig. 5 Natural speech articulation simulation during the speech generation process

especially Slavish ones, because of many phonetic and linguistic differences. We can assure that a good system for speech communication between a man and machines, which can accept and use Polish language is discovered and developed near the Vistula River!

Therefore we must develop some research leading to the design and construction of automatic speech recognition system dedicated to Polish language recognition and based on Polish speech model.

5   Elements of Speech Recognition Process and Final Remarks

The scope of this paper is very limited one, while the problem under consideration is very complicated and includes many important details, which cannot be fully presented here. Therefore most problems originated by the Polish speech recognition problem will be presented and discussed in the oral presentation of a keynote lecture, while in this paper such problems are only mentioned.

The list of problems, which must be solved for successful design, construction and development of the speech automatic recognition system includes the following items:

- speech signal acquisition and feeding it into the system,
- acoustic signal preprocessing for enhancing its features, which can be used for recognition,
- segmentation of speech signals for localization of recognized elements (e.g., phonemes, triphones, syllables, words etc.),
- extraction of speech signal parameters, used as a base for elements recognition,
- recognition of the elements and reconstruction of the biggest speech parts (e.g., words, sentences, paragraphs etc.) on the base of recognized sequences of elements,
- analysis of the lexical, syntactical and semantic structure of the recognized speech parts on the base of language models for corrections of badly recognized elements and parts,
- understanding of the merit sense of the whole utterance and using the result of such understanding as the intelligent input to the final information system, which is the goal of our work.

For some of the problems listed above we have now good solutions also for Polish speech, but definitely not for all. At the same time the need for speech communication with machines increases rapidly and it is very urgent to solve these problems for opening ways to practical applications. Therefore Polish speech recognition problem is worth intensive scientific and engineering research and to initiating and accelerating such research this keynote speech is dedicated.

References

1. Cernys, P., Kubilius, V., Macerauskas, V., Ratkevicius, K.: Intelligent control of the lift model. In: Proceedings of the IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Lviv (2003)
2. Demenko, G., Grocholewski, S., Klessa, K., Ogórkiewicz, J., Wagner, A., Lange, M., Śledziński, D., Cylwik, N.: JURISDIC-Polish speech database for taking dictation of legal texts. In: Proceedings of the International Conference on Language Resources and Evaluation (2008)
3. ELRA: European Language Resources Association, http://www.elra.info/
4. Engdahl, T.: How to use disk drive stepper-motors, Communications Limited (2004), http://www.epanorama.net/circuits/diskstepper.html
5. Johnson, J.: Working with stepper motors. Electronic Inventory Online (1998), http://eio.com/jasstep.htm
6. Laboratory of Speech and Language Technology, http://www.speechlabs.pl
7. Loof, J., Gollan, C., Hahn, S., Heigold, G., Hoffmeister, B., Plahl, D., Rybach, R., Schluter, S., Ney, H.: The RWTH TC-STAR evaluation system for European English and Spanish. In: Proceedings of Interspech, pp. 2145–1249 (2007)
8. Pires, G., Nunes, U.: A wheelchair steered through voice commands and assisted by a reactive fuzzy-logic controller. Journal of Intelligent and Robotic Systems 34(3), 301–314 (2002)

9. Shaughnessy, D.: Interacting with computers by voice: automatic speech recognition and synthesis. Proceedings of the IEEE 91(9), 1272–1305 (2003)
10. Simpson, R., LoPresti, E., Hayashi, S., Nourbakhsh, I., Miller, D.: The smart wheelchair component system. Journal of Rehabilitation Research & Development 41(3B), 429–442 (2004)
11. Young, S.: Large vocabulary continuous speech recognition. IEEE Signal Processing Magazine 13(5), 45–57 (1996)

Stochastic Effects in Signaling Pathways in Cells: Interaction between Visualization and Modeling

Marek Kimmel

Abstract. We review a research program under way at Rice University in Houston, TX and University of Texas Medical Branch in Galveston, TX, in the United States and at the Systems Engineering Group at the Silesian University of Technology in Gliwice in Poland. Individual biological cells display stochastic variability in their responses to activating stimuli. This variability can be measured using recent molecular biology techniques, which prove that in many respects no cell in the population behaves like an average cell. In cells taking part in the innate immune response this variability seems very important. In prokaryotes, which are small, importance of stochastic effects at all levels of the transcription/translation process was recognized early on. Eukaryotic cells are much larger and also have more complex mechanisms of transcription initiation. Since stochastic effects arise mainly through interactions of a limited number of discrete entities (such as transcription factors, promoter binding sites, receptors and so forth), it is to be expected that in eukaryotic cells these effects will be mainly due to transcription initiation and to signaling mediated by small numbers of active molecules (such as recognition of foreign antigens by T lymphocytes. We present the biological system which is the subject of our research, as well as an outline of mathematical and computational methods which we use to analyze it. Visualization and modeling are two major elements of this approach.

Keywords: stochastic process, robustness, gene transcription and control, modeling, eukaryotic cells.

Marek Kimmel
Systems Engineering Group, Department of Statistics, Rice University,
Houston, US
e-mail: `kimmel@rice.edu`
and
Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: `marek.kimmel@polsl.pl`

1 Introduction

This paper reviews a research program under way at Rice University in Houston, TX and University of Texas Medical Branch in Galveston, TX, in the United States and at the Systems Engineering Group at the Silesian University of Technology in Gliwice in Poland. The description concerns past and current as well as future research.

Individual biological cells display stochastic variability in their responses to activating stimuli. This variability can be measured using recent molecular biology techniques, which prove that in many respects no cell in the population behaves like an average cell. In cells taking part in the innate immune response this variability seems very important. *It confers a kind of Stochastic Robustness : Subpopulations of cells may react differently to the same stimulus, but all of them react in a well-defined way.* There is a growing interest in stochastic effects in the dynamics of gene transcription and signal transduction in eukaryotic cells [4]. In prokaryotes, which are small, importance of stochastic effects at all levels of the transcription/translation process was recognized early on [1]. Eukaryotic cells are much larger and also have more complex mechanisms of transcription initiation [36]. Since stochastic effects arise mainly through interactions of a limited number of discrete entities (such as transcription factors, promoter binding sites, receptors and so forth), it is to be expected that in eukaryotic cells these effects will be mainly due to transcription initiation and to signaling mediated by small numbers of active molecules (such as recognition of foreign antigens by T lymphocytes, [10]). As a result, variability will arise in cells in the population, which has a dynamic nature, as demonstrated by the experiments of Nelson et al. [25], Sigal et al. [31] and Geva-Zatorsky et al. [13], and which results in a possibly stable and mixing stochastic process [31]. A number of researchers posed questions concerning the importance of stochastic effects for functioning of cell populations and organisms. As an example, Blake et al. [2] considered transcriptional noise in yeast concluding, based on a mathematical model, that there exists a noise level, which optimizes transcriptional efficiency. Raser and O'Shea [29] conclude that noise (variability) among cells has multiple sources, including the stochastic or inherently random nature of the biochemical reactions of gene expression; they also comment on recent investigations into the sources, consequences, and control of noise in gene expression.

Mathematical modeling of stochastic effects in gene transcription and control has a tradition reaching at least as long as Peccoud and Ycart [27], Arkin et al. [1], and Kierzek et al. [19], all using mainly Gillespie's [15] discrete stochastic simulation algorithm. In eukaryotes, the large numbers of mRNA and protein molecules (in excess of 10^2 and 10^4, respectively), require that the system be partly continuous. The breakthrough innovation seems to belong to Kepler and Elston [18]. In the context of NFκB module, Lipniacki et al. [22, 21] introduced their 'continuous' model, essentially a version of a stochastic differential equation (SDE) system. These models explain results of single-cell experiments of Nelson et al. [25] and previous experimental work [21].

As detailed further on, the stochastic nature of transcription initiation is likely due to fluctuation at the level of assembly of the transcription complexes attracting

RNA Polymerase II [28]. There is an open question as to what is the mean period of stochastic oscillations of this process, which was assayed in variety of cell systems. Estimates based on photobleaching experiments have the order of 5-10 seconds [30]. Indirect estimates based on mathematical modeling in [28] imply the order of hours. Constants used in [21] imply 10–20 minutes. Apparent divergence of these estimates may result from estimation inaccuracy or from incomplete understanding of the mechanistic principles underlying transcrition activation.

The biological system we consider is constituted by 3 pathways involving NFκB family of transcription factors playing a decisive role in innate immunity in mammals, with evolutionary roots reaching into the past as deep as *Drosophila* [37].

The innate immune response plays the role of a first line of defense from potentially harmful organisms. In this response, infecting organisms induce cellular signaling pathways to protective cytokines such as interferon. These three well-defined NF-κB signaling pathways, known as the canonical, the RIG-I-MAVS-, and the non-canonical pathways, are activated by distinct stimuli, producing dynamic cytoplasmic-nuclear oscillatory behavior and serve as informative models for computational analysis of sources of stochasticity in a biologically important eukaryotic signaling pathway. We plan to extend the current understanding of these pathways by applying recent advances in dynamic single cell imaging using fluorescent fusion proteins, analysis of transcription at a single mRNA molecule resolution, and chromatin exchange using photobleaching and fluorescence lifetime measurements.

Mathematical tools for model building, analysis, computation and estimation of parameteres in cell systems involving stochastic effects do not seem to be sufficiently developed. Papers such as [18] and [22] are based on special cases or heuristic derivations. The transition between discrete and continuous part in the mixed models is not well-justified. Models of processes at nuclear level lead to difficult partial differential equations of diffusion (FRAP models) and transport type (Gillespie-type models). The only paper known to us concerning systemic approach to estimation is Fujarewicz et al. [12]).

The aim of this research is to develop tools for better understanding and more accurate modeling of stochastic phenomena related to dynamics of gene transcription and signal transduction in eukaryotic cells. This involves analyzing mathematical models of specific signaling pathways, developing computational tools of more general applicability and constructing mathematical framework for statistical inference. The models are based on jointly devised experiments carried out by our biological co-workers and consultants.

2 Biological System Considered and Aims

Mucosal surfaces in the airway, gastrointestinal tract and skin form play an important role in maintaining physiological homeostasis by forming barriers that prevent foreign organisms from entering and causing disease. Here, the innate immune response plays a first line of defense from these potentially harmful substances or organisms. In the innate immune response, products of viruses, fungi, or bacteria

are recognized by mucosal epithelial cells and induce cellular signaling pathways that produce inflammatory and protective cytokines such as interferons [3]. Two important consequences of this inflammatory response are to: 1. induce expression of protective proteins in adjacent epithelial cells; and, 2. recruit effector immune cells to further combat the infection. In this way, the innate immune response serves as a critical signaling pathway in maintaining health.

Studies from our group and others have shown that nuclear factor-κB (NF-κB) is a highly inducible transcription factor that plays a central role in the innate immune response [6], [23]. NF-κB is a family of 11 highly homologous proteins whose activity is controlled by distinct activation pathways and in stimulus-specific manner [5], [14]. Although the signaling pathways controlling innate immune response by NF-κB to infectious stimuli are being intensively studied, the theoretical understanding of stochasticity in these signaling pathways is poorly developed. We propose a series of mathematical and biological experiments on three interrelated signaling pathways regulating the nuclear factor-κB (NF-κB) and interferon response factor (IRF) pathways that can be used to develop more general mathematical tools for understanding the role of stochasticity in biological systems. These pathways include the canonical-, the RIG-I-MAVS-, and the noncanonical NF-κB activation pathways.

1. Measure *time-dependent parameters* of key NF-κB signaling proteins *in cell populations*. Western immunoblots of cytoplasmic and nuclear proteins will be used to measure the major regulatory components of NF-κB at various times after stimulation for the canonical-, the RIG-I-MAVS-, and the non-canonical pathways. In parallel, kinetics of target gene expression will be measured by quantitative RT-PCR (Q-RT-PCR). These data will be used as parameters in deterministic models of each pathway.

2. Measure *dynamics of nuclear oscillations* of signaling proteins and *stochastics of transcriptional response*. In this aim, cytoplasmic and nuclear distributions of fluorescent protein tagged RelA and IRF3 in response to activators of the canonical and non-canonical pathways will be measured. Dose response curves for TNF activation will be measured in *single cells* using a high throughput confocal imager. Measurement of the transcriptional response will be performed in cells at the single molecule level using stably integrated genes detected by fluorescence in situ hybridization (FISH).

3. *Perturb the two major negative feedback loops* and determine their effects on the NF-κB regulatory module. NF-κB activation induces the expression of two autoregulatory loops, known as the IκBα and A20 feedback loops responsible for oscillatory DNA binding and inhibition of the major regulatory IκB kinase. In this aim, *we will construct synthetic feedback loops* incorporating either low or highly inducible IκBα or A20 expression, and determine their effects on fluorescent RelA nuclear translocation and IKK activity.

4. Measure dynamics of NF-κB *binding to rapidly and slowly responding genes* using *FRAP* and fluorescence loss in photobleaching (*FLIP*). Arrays of rapidly (IκBα) and slowly (Naf1) NF-κB inducible genes linked to monomeric red fluorescent protein reporter genes will be introduced into cells containing a regulated green fluorescent protein tagged RelA. The dynamic exchange rates of

RelA binding to each will be measured in stimulated cells by measuring the time required for exchange of the bleached for fluorescent RelA molecules.

3 Mathematical, Statistical, and Computational Aims

1. Identify sources of stochastic effects in gene transcription and regulation on single-cell, nuclear and molecular level and develop *mathematical models* of these effects, taking into account processes and quantities observable and measurable using recent technological advances.
2. Investigate the mathematical *properties of these models* by: (a) Finding deterministic and corresponding stochastic solutions in the form of probability generating functions (discrete models) and distribution densities (continuous models). (b) Developing limit theory, which justifies a wide range of approximate solutions. (c) Investigating qualitative properties of the models.
3. Develop efficient and accurate *computational algorithms* for calculation of model predictions under various experimental scenarios and including variations of parameters within the models. Implement computer programs for these algorithms.
4. Apply formal *statistical methodologies* for estimating parameters in the models from experimental data and making inferences about these parameters, and assess the goodness of fit of the models. We propose to apply existing Bayesian and non-Bayesian methods and to develop new methods for inference with complex computer models.

4 Stochastic Effects in Dynamics of Interaction of Transcription Factors with Binding Sites in Cell Nuclei

Kinetics of binding of transcription factors and co-factors to promoter regions of genes is not very accurately known. It has been recently determined by a series of works utilizing diverse visualization techniques including fluorescence recovery after photobleaching (FRAP) that there may exist a pronounced stochastic component of this kinetics, depending on the number of binding sites and transcription factor molecules interacting. In this section, we will review the relevant evidence for different sources of stochasticity. However, we will start from description of a technique, which has been mostly applied in the cases when the number of molecules and sites is large enough for the system to be considered deterministic.

4.1 Deterministic Approximation, Kinetics of Photobleaching, Diffusion-Type PDEs

Fluorescence recovery after photobleaching (FRAP) is a popular technique that has been used to measure mobilities of fluorescently tagged proteins inside living cells

[20]. In this technique, a small spot in the cytoplasm, nucleus or plasma membrane of a living cell that expresses or is microinjected with a fluorescently tagged protein, is exposed to an intense laser beam at the excitation wavelength of the fluorophore. The intense irradiation causes photobleaching of the fluorescent protein in the spot making it optically invisible, although its binding functions are not altered. Because non-bleached fluorescent molecules present in surrounding areas diffuse into the irradiated region, fluorescence recovery occurs in the spot and this can be used to estimate the diffusion coefficient of the protein. If the photobleached spot contains a significant number of fluorescent molecules that are bound to insoluble scaffolds inside cells, then the recovery curve can be utilized to estimate binding (k_{on}) and unbinding (k_{off}) constants of the proteins, in addition to the diffusion coefficients, provided sufficient measurement accuracy is reached. This requires the formulation of mathematical models that can be used to estimate kinetic rate constants for binding of proteins to scaffolds.

We presume a large (infinite) region that is at rest prior to photobleaching a volume Ω. As such, the initial conditions for the concentrations of the fluorescent protein, f, and its bound state, c, are

$$f(x,0) = f_{eq}(1 - \chi_\Omega(x)) \quad \text{and} \quad c(x,0) = c_{eq}(1 - \chi_\Omega(x)), \tag{1}$$

where χ_Ω is the characteristic function of Ω. These initial concentrations then evolve according to the standard coupled diffusion model

$$\frac{\partial f}{\partial t} = D\Delta f + k_{off}c - k_{on}f, \tag{2}$$

$$\frac{\partial c}{\partial t} = k_{on}f - k_{off}c. \tag{3}$$

Now (2)–(3) subject to (1) is a well posed boundary value problem. As a result the boundary values of f,

$$F(x,t) \equiv f(x,t), \quad x \in \partial\Omega \tag{4}$$

are uniquely determined by the diffusivity and two rate constants. This model is overdetermined by the fluorescence recording

$$\phi(t) = \int_\Omega \{f(x,t) + c(x,t)\} \, dx$$

for, on integrating (2)–(3) over Ω it is not hard to see that

$$\phi(t) = D \int_{\partial\Omega} \nabla f(x,t)n(x) \, dx, \tag{5}$$

where n is the unit inner normal to $\partial\Omega$. In the case that Ω is a ball then (4) and (5) constitute Dirichlet and Neumann data for (2)–(3) and we may draw upon the sizable literature devoted to lateral overdetermination of parabolic systems. In particular, the theory is spelled out in [17], while detailed application is carried out in [7, 11, 8, 9]. Although this literature addresses the questions of identifiability, sensensitivity

to noise, and delivers practical algorithms for efficient recovery of values such as D, k_{on} and k_{off}, the current, most quantitative, studies of FRAP, e.g., [34, 32, 33], have focused on exact solution methods in very special geometries.

4.2 Random RNA Bursts, Due to Small Number of Binding Sites

Under experimental conditions such as in [30], the overall number of molecules of the protein binding to the promoters of tandemly arrayed gene copies was so large that deterministic description using differential equations was appropriate. However, under different circumstances, stochastic effects become important. Raj et al. [28] observed stochastic bursts of transcription from a reporter gene inserted into the genome of Chinese Hamster Ovary (CHO) cells. In this case, there were either 1 or 7 DNA binding sites for the transcription factor of a single gene copy. These observations are indirect, based on observed cell-cell variability in level of the total mRNA in single cells. To obtain estimates of the dynamics of gene activation and transcription, Raj et al. [28] build a mathematical model in which they assume, among other, that the transitions from the inactive (I) to the active (A) state are random and occur with intensities λ and γ, respectively (see Figure). By fitting the complete model ([28], Supplement) to distributions of total mRNA, they estimated the expected times of the reporter gene being transcriptionally active and inactive to be equal to $E(T_A) = \gamma^{-1} = 0.8$ hr., and $E(T_I) = \lambda^{-1} = 2.2$ hr., respectively.

4.3 Stochastic Effects Due to Limiting Co-factors

Stochastic effects may be present even in when there is a large number of arrayed promoter sites with large number of bound molecules. As an example, in the paper by Voss et al. [35], the glucocorticoid receptor (GR) was investigated, which dynamically interacts with response elements in the mouse mammary tumor virus (MMTV) promoter to regulate steroid-dependent transcription. In a clonal mammary carcinoma cell line containing a tandem array of MMTV promoter-reporter gene cassettes integrated at a single genomic locus (total of 800–1200 binding sites in 200 tandemly arrayed gene copies [24], direct binding of a green fluorescent protein (GFP-GR) fusion protein to the MMTV regulatory elements can be observed in living cells. A pronounced cell-to-cell variability was observed in RNA FISH signal and GR-MMTV association within treatment groups. The authors of [35] hypothesize that the GR receptors exist in the nucleoplasmic space in a large variety of multiprotein complexes (Fig. in [35]). These complexes are recruited randomly and stochastically to hormone response elements but remain template associated for brief periods of time. The authors conclude that the transcriptional process induced by nuclear receptor activation involves a series of highly stochastic events, with considerable variation in efficiency possible at each stage.

5   Stochastic Models of Transcription Activation

5.1   *Stochastic Model with Multiple Gene Copies and Multiple Binding Sites [26]*

The simple model employed in [28] can be used to draft a slightly more general model applicable to systems with many gene copies and many transcription factor binding sites, which will be applicable for biological models in several papers cited above. Let us consider a system of N serially arrayed genes, each with K functional binding sites in the promoter region. Let us notice the following partial list of possibilities: (i) Deterministic approximation, K and/or N large, and/or λ and γ and μ large. (ii) Stochastic effects due to small numbers of binding sites, K and N small. (iii) Stochastic effects due to limiting co-factors and/or low abundance of transcription factors, λ small. Various intermediate and more complicated variants are possible. For example, the model in Raj et al. [28] has $N = 1$, $K = 1$ or 7, and μ large in one of the versions. The models in Hat et al. [16] involve $N = 1$, 2, or 4 and $K = 1$ with large μ. This variety of options is increased when translation of mRNA into proteins is included in the model.

The 'chemical master equation' of Gillespie [15] provides a well-known algorithm for simulation of chemical systems (SSA, 'Stochastic Simulation Algorithm', also see Aim 2) with random interactions of a finite numbers of particles of many types. This method can be used to model systems in which gene activation triggers transcription and translation [19]. In some cases, it may be used to derive analytically tractable differential equations. As an example, Paszek [26] considered several versions of systems of partial differential equations (PDE) of transport type, which describe stochastic dynamics of the transcription and translation process in a simple model involving one gene. In this model, the probabilities of gene activation $(A(t):\ 0 \rightarrow 1)$ and deactivation $(A(t):\ 1 \rightarrow 0)$ in $(t, t + \Delta t)$ are correspondingly equal to $c\Delta t + o(\Delta t)$ and $d\Delta t + o(\Delta t)$, while those of production $(X(t) \rightarrow X(t) + 1)$ and degradation $(X(t) \rightarrow X(t) - \min[X(t), 1])$ of a mRNA molecule, are correspondingly equal to $A(t)H\Delta t + o(\Delta t)$ and $\Delta t + o(\Delta t)$, whereas those of a protein molecule production and $(Y(t) \rightarrow Y(t) + 1)$ and degradation $(Y(t) \rightarrow Y(t) - \min[Y(t), 1])$ are equal to $X(t)K\Delta t + o(\Delta t)$ and $r\Delta t + o(\Delta t)$. The distributions of the process are described by the PDE system,

$$\frac{\partial F}{\partial t} + [(z-1) - Kz(s-1)]\frac{\partial F}{\partial z} + r(s-1)\frac{\partial F}{\partial s} = -cF + bG, \tag{6}$$

$$\frac{\partial G}{\partial t} + [(z-1) - Kz(s-1)]\frac{\partial G}{\partial z} + r(s-1)\frac{\partial G}{\partial s} = cF + [H(z-1) - b]G, \tag{7}$$

where $F = F(z, s; t)$, $G = G(z, s; t)$, represent the joint probability generating function (pgf) of $(X(t), Y(t))$ when the gene is inactive $(A(t) = 0)$ or active $(A(t) = 1)$. More equations of this type, corresponding to various models are presented in [26]. Solution and qualitative properties of these models constitute considerable mathematical challenge. However, they can be used with ease to generate solvable moment equations.

5.2 *Mixed-Type Equations*

In eukaryotic cells, the stochastic effects primarily originate in regulation of gene activity [18]. In this approach, the ordinary differential equations for mRNA and protein levels in a single cells are driven by a stochastic term related to gene activation

$$\dot{x}(t) = -x(t) + HA(t),$$
$$\dot{y}(t) = -ry(t) + Kx(t),$$

where intuitively, $x(t) \sim X(t)$, $y(t) \sim Y(t)$ whereas the transitions of $A(t)$ are governed by

$$A(t-0) = 0 \xrightarrow{c} A(t) = 1,$$
$$A(t-0) = 1 \xrightarrow{d} A(t) = 0,$$

where rates c and d may depend on continuous state variables. This system is a counterpart of (6-7) when rates H, K, r are large compared to c, d. These equations yield a system of first-order partial differential equations (PDEs) for two-dimensional joint probability density functions $f(x,y,t)$ and $g(x,y,t)$, of $x(t)$ and $y(t)$, with $A(t) = 0$ and $A(t) = 1$, respectively

$$\partial f / \partial t - \partial(xf)/\partial x + \partial[(Kx - ry)f)]/\partial y = byg - cf,$$
$$\partial f / \partial t + \partial[(H-x)g]/\partial x + r\partial[(Kx - ry)f)]/\partial y = -byg + cf.$$

The model can be considered a set of quasi-deterministic equations with jump-process $A(t)$ forcing, with corresponding Fokker-Planck or Chapmann-Kolmogorov equations for distributions. Verifying validity of this mixed-type approximation is one of the Aims of our proposal. Numerical examples indicate the approximation varies from excellent to poor. The paper of Nelson et al. [25] presents an experimental study of responses of individual cells under a variety of activation levels and patterns. We will continue studying dynamics of such responses, as we did in [22] and [21], based on our own experiments.

References

1. Arkin, A., Ross, J., McAdams, H.H.: Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected escherichia coli cells. Genetics 149, 1633–1648 (1998)
2. Blake, W.J., Kaern, M., Cantor, C.R., Collins, J.J.: Noise in eukaryotic gene expression. Nature 422, 633–637 (2003)

3. Bonizzi, G., Karin, M.: The two NF-kB activation pathways and their role in innate and adaptive immunity. Trends in Immunology 25, 280–288 (2004)

4. Bossisio, D., Marazzi, I., Agresti, A., Shimizu, N., Bianchi, M.E., Natoli, G.: A hyper-dynamic equilibrium between promoter bound and nucleoplasmic dimers controls NF-κB-dependent gene activity. European Molecular Biology Organization Journal 25, 798–810 (2006)

5. Brasier, A.R.: The NF-κB regulatory network. Cardiovascular Toxicology 6 (2006) (in press)

6. Caamano, J.H., Hunter, C.A.: NF-kB family of transcription factors: Central regulators of innate and adaptive immune functions. Clinical Microbiological Review 15, 414–429 (2002)

7. Cox, S.J.: A new method for extracting cable parameters from input impedance data. Mathematical Biosciences 153, 1–12 (1998)

8. Cox, S.J.: Estimating the location and time course of synaptic input from multi-site potential recordings. Journal of Computational Neuroscience 17, 225–243 (2004)

9. Cox, S.J., Wagner, A.: Lateral overdetermination of the FitzHugh-Nagumo system. Inverse Problems 20, 1639–1647 (2004)

10. Davies, M., Krogsgaard, M., Huppa, J.B., Sumen, C., Purbhoo, M.A., Irvine, D.J., Wu, L.C., Ehrlich, L.: Dynamics of cell surface molecules during T cell recognition. Annual Review of Biochemistry 72, 717–742 (2003)

11. Farrell, B., Cox, S.J.: The rate of pore expansion during the spike phase of exocytotic release in mast cells of the beige mouse. Bulletin of Mathematical Biology 64(5), 979–1010 (2002)

12. Fujarewicz, K., Kimmel, M., Lipniacki, T., Swierniak, A.: Adjoint systems for models of cell signalling pathways and their application to parametr fitting. IEEE/ACM Transactions on Computational Biology and Bioinformatics (2006) (accepted)

13. Geva-Zatorsky, N., Rosenfeld, N., Itzkovitz, S., Milo, R., Sigal, A., Dekel, E., Yarnitzky, T., Liron, Y., Polak, P., Lahav, Y., Alon, U.: Oscillations and variability in the p53 system. Molecular Systems Biology 2, 0033 (2006)

14. Ghosh, S., May, M.J., Kopp, E.B.: NF-kappa B and Rel proteins: evolutionarily conserved mediators of immune responses. Annual Review of Immunology 16(60), 225–260 (1998)

15. Gillespie, D.T.: A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. Journal of Computational Physics 22, 403–434 (1976)

16. Hat, B., Paszek, P., Kimmel, M., Piechor, K., Lipniacki, T.: How the number of alleles influences gene expression. Journal of Statistical Physics 128, 511–533 (2007)

17. Isakov, V.: Inverse Problems for Partial Differential Equations. Springer, New York (1998)

18. Kepler, T.B., Elston, T.C.: Stochasticity in transcriptional regulation: origins, consequences, and mathematical representations. Biophysics Journal 81, 3116–3136 (2001)

19. Kierzek, A.M., Zaim, J., Zielenkiewicz, P.: The effect of transcription and translation initiation frequencies on the stochastic fluctuations in prokaryotic gene expression. Journal of Biological Chemistry 276, 8165–8172 (2001)

20. Lele, T.P., Ingber, D.E.: A mathematical model to determine molecular kinetic rate constants under non-steady state conditions using fluorescence recovery after photobleaching (FRAP). Biophysical Chemistry 120, 32–35 (2006)

21. Lipniacki, T., Paszek, P., Brasier, A.R., Luxon, B., Kimmel, M.: Stochastic regulation in early immune response. Biophysics Journal 90, 725–742 (2006)

22. Lipniacki, T., Paszek, P., Marciniak-Czochra, A., Brasier, A.R., Kimmel, M.: Transcriptional stochasticity in gene expression. Journal of Theoretical Biology 238, 348–367 (2006)

23. Liu, P., Choudhary, S., Jamaluddin, M., Li, K., Garofalo, R., Casola, A., Brasier, A.R.: Respiratory syncytial virus activates interferon regulatory factor-3 in airway epithelial cells by upregulating a RIG-I-Toll like receptor-3 pathway. Journal of Virology (2006) (in press)

24. McNally, J., Muller, G.W., Walker, D., Wolford, R., Hager, G.L.: The glucocorticoid receptor: Rapid exchange with regulatory sites in living cells. Science 287, 1262–1265 (2000)

25. Nelson, D.E., Ihekwaba, A.E.C., Elliot, M., Johnson, J.R., Gibney, C.A., Foreman, B.E., Nelson, G., See, V., Horton, C.A., Spiller, D.G., Edwards, S.W., McDowell, H.P., Unitt, J.F., Sullivan, E., Grimley, R., Benson, N., Broomhead, D., Kell, D.B., White, M.R.H.: Oscillations in NF-κB signaling control the dynamics of gene expression. Science 306, 704–708 (2004)

26. Paszek, P.: Modeling stochasticity in gene regulation: Characterization in the terms of the underlying distribution function. Bulletin of Mathematical Biology (2006) (to appear)

27. Peccoud, J., Ycart, B.: Markovian modeling of gene-product synthesis. Theory of Population Biology 48, 222–234 (1995)

28. Raj, A., Peskin, C.S., Tranchina, D., Vargas, D.Y., Tyagi, S.: Stochastic mRNA synthesis in mammalian cells. Public Library of Science Biology 4(10), e309 (2006)

29. Raser, J.M., O'Shea, E.K.: Control of stochasticity in eukaryotic gene expression. Science 304, 1811–1814 (2004)

30. Sharp, Z.D., Mancini, M.G., Hinojos, C.A., Dai, F., Berno, V., Szafran, A.T., Smith, K.P., Lele, T.P., Ingber, D.E., Mancini, M.A.: Estrogen-receptor-α exchange and chromatin dynamics are ligand- and domain-dependent. Journal of Cell Science 119, 4365 (2006)

31. Sigal, A., Milo, R., Cohen, A., Geva-Zatorsky, N., Klein, Y., Liron, Y., Rosenfeld, N., Danon, T., Perzov, N., Alon, U.: Variability and memory of protein levels in human cells. Nature 444, 643–646 (2006)

32. Sprague, B.L., McNally, J.G.: FRAP analysis of binding: proper and fitting. Trends in Cell Biology 15(2), 84–91 (2005)

33. Sprague, B.L., Müller, F., Pego, R.L., Bungay, P.M., Stavreva, D., McNally, J.: Analysis of binding at a single spatially localized cluster of binding sites by fluorescence recovery after photobleaching. Biophysics Journal 91, 1169–1191 (2006)

34. Sprague, B.L., Pego, R.L., Stavreva, D.A., McNally, J.G.: Analysis of binding reactions by fluorescence recovery after photobleaching. Biophysics Journal 86, 3473–3495 (2004)

35. Voss, T.C., John, S., Hager, G.L.: Single-cell analysis of glucocorticoid receptor action reveals that stochastic post-chromatin association mechanisms regulate ligand-specific transcription. Molecular Endocrinology 20, 2641–2655 (2006)

36. White, R.J.: Gene Transcription: Mechanisms and Control. Blackwell Science Limited, Oxford (2001)

37. Zhang, G., Ghosh, S.: Toll-like receptor-meduated NFκB activation: A phylogenetically conserved paradigm in innate immunity. Journal of Clinical Investment 107, 13–19 (2001)

Rough-Granular Computing in Human-Centric Information Processing

Andrzej Jankowski and Andrzej Skowron

Abstract. In the area of ubiquitous computing, users will continuously interact with computing devices by suggesting strategies, hypothesis, by communicating some new facts from domain knowledge, explaining untypical cases in dialogs with devices (agents), etc. Hence, compound vague concepts used by humans should be understandable, at least in an approximate sense, by these devices. We discuss some basic issues of interactive computations in the framework of rough-granular computing for approximation of complex concepts. Among these issues are hierarchical modeling of granule structures and interactions between granules of different complexity. Interactions between granules on which computations are performed are among the fundamental concepts of Wisdom Technology (Wistech). Wistech is encompassing such areas as interactive computations, multiagent systems, cognitive computation, natural computing, complex adaptive and autonomous systems, or knowledge representation and reasoning about knowledge. We outline current results on approximation of compound vague concepts which are based on rough-granular computing. In particular, hierarchical methods are used for approximation of domain ontologies of vague concepts. The developed methodology has been tested on real-life problems related to such areas as unmanned area vehicle control, robotics, predicting of risk patterns from temporal medical and financial data, sun spot classification, bioinformatics.

Keywords: rough sets, granular computing, rough-granular computing, judgment, interaction, wisdom technology (Wistech).

Andrzej Jankowski
Institute of Decision Processes Support and AdgaM Solutions Sp. z o.o.
Wąwozowa 9/64, 02-796 Warsaw, Poland
e-mail: `andrzejj@adgam.com.pl`

Andrzej Skowron
Institute of Mathematics, The University of Warsaw,
Banacha 2, 02-097 Warsaw, Poland
e-mail: `skowron@mimuw.edu.pl`

1 Introduction

The radical changes in Knowledge Technology depend on the further advancement of technology to acquire, represent, store, process, discover, communicate and learn wisdom. We call this technology *wisdom technology* (or Wistech, for short). The term *wisdom* commonly means *rightly judging*. This common notion can be refined. By *wisdom*, we understand an adaptive ability to make judgments correctly to a satisfactory degree (in particular, correct decisions) having in mind real-life constraints. The intuitive nature of wisdom understood in this way can be expressed by the so called *wisdom equation* [18], metaphorically shown as follows.

$$wisdom = knowledge + adaptive\ judgment + interactions. \tag{1}$$

It is worthwhile mentioning that the wisdom concept was intensively discussed by many famous philosophers starting form ancient times. For example, in [58] one can find the following sentences:

Aristotle's man of practical wisdom, the phronimos, does not ignore rules and models, or dispense justice without criteria. He is observant of principles and, at the same time, open to their modification. He begins with nomoi – established law – and employs practical wisdom to determine how it should be applied in particular situations and when departures are warranted. Rules provide the guideposts for inquiry and critical reflection.

Wisdom can be treated as a special type of knowledge processing. In order to explain the specificity of this type of knowledge processing, let us assume that a control system of a given agent *Ag* consists of a society of agent control components interacting with the other agent *Ag* components and with the agent *Ag* environments. Moreover, there are special agent components, called as the agent coordination control components which are responsible for the coordination of control components. Any agent coordination control component mainly searches for answers for the following question: *What to do next*? or, more precisely: *Which of the agent's* Ag *control components should be activated now*? Of course, any agent control component has to process some kind of knowledge representation. In the context of agent perception, the agent *Ag* itself (by using, e.g., interactions, memory, and coordination among control components) is processing a very special type of knowledge reflecting the agent perception of the hierarchy of needs (objectives, plans, etc.) and the current agent or the environment constraints. This kind of knowledge processing mainly deals with complex vague concepts (such as risk or safety) from the point of view of the *selfish* agent needs. Usually, this kind of knowledge processing is not necessarily logical reasoning in terms of proving statements (i.e., labeling statements by truth values such as TRUE or FALSE). This knowledge processing is rather analogous to the judgment process in a court aiming at recognition of evidence which could be used as an argument *for* or *against*. Arguments *for* or *against* are used in order to make the final decision which one of the solutions is the best for the agent in the current situation (i.e., arguments are labeling statements by

judgment values expressing the action priorities). The evaluation of currents needs by agent *Ag* is realized from the point of view of hierarchy of agent *Ag life* values/needs). Wisdom type of knowledge processing by the agent *Ag* is characterized by the ability to improve quality of the judgment process based on the agent *Ag* experiences. In order to emphasize the importance of this ability, we use the concept of *adaptive judgment* in the wisdom equation instead of just *judgment*. An agent who is able to perform adaptive judgment in the above sense, we simply call as a *judge*.

The adaptivity aspects are also crucial from the point of view of interactions [17, 27, 33, 54]. The need for adaptation follows, e.g., from the fact that complex vague concepts on the basis of which the judgment is performed by the agent *Ag* are approximated by classification algorithms (classifiers) which are very often drifting in time following changes in data and represented knowledge.

An important aspect of Wistech is that the complexity and uncertainty of real-life constraints mean that in practice we must reconcile ourselves to the fact that our judgments are based on non-crisp concepts (i.e., concepts with borderline cases) and also do not take into account all the knowledge accumulated and available to us. This is why our judgments are usually imperfect. But as a consolation, we also learn to improve the quality of our judgments via observation and analysis of our experience during interaction with the environment. Satisfactory decision-making levels can be achieved as a result of improved judgments.

Thus wisdom is directly responsible for the focusing of an agents attention (see Aristotle tetrahedron in Fig. 1) on problems and techniques of their solution which are important in terms of the agent judgment mechanism. This mechanism is based on the Maslow hierarchy of needs (see Fig. 2) and agent perception of ongoing interactions with other agents and environments. In particular, the agent's wisdom can be treated, as the control at the highest level of hierarchy of the agent's actions and reactions and is based on concept processing in the metaphoric Aristotle tetrahedron (Fig. 1). One can use the following conceptual simplification of agent wisdom. Agent wisdom is an efficient and an on-line agent judgment mechanism making it

Fig. 1 Relationships between imagination, judgment, perception and psyche
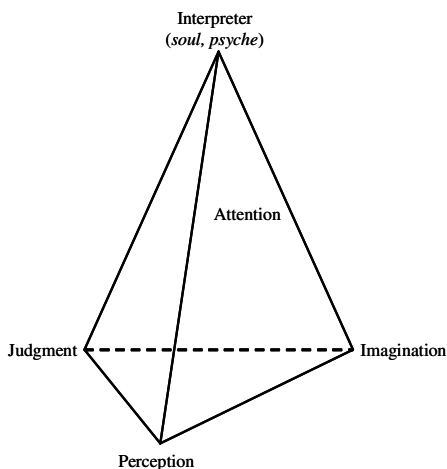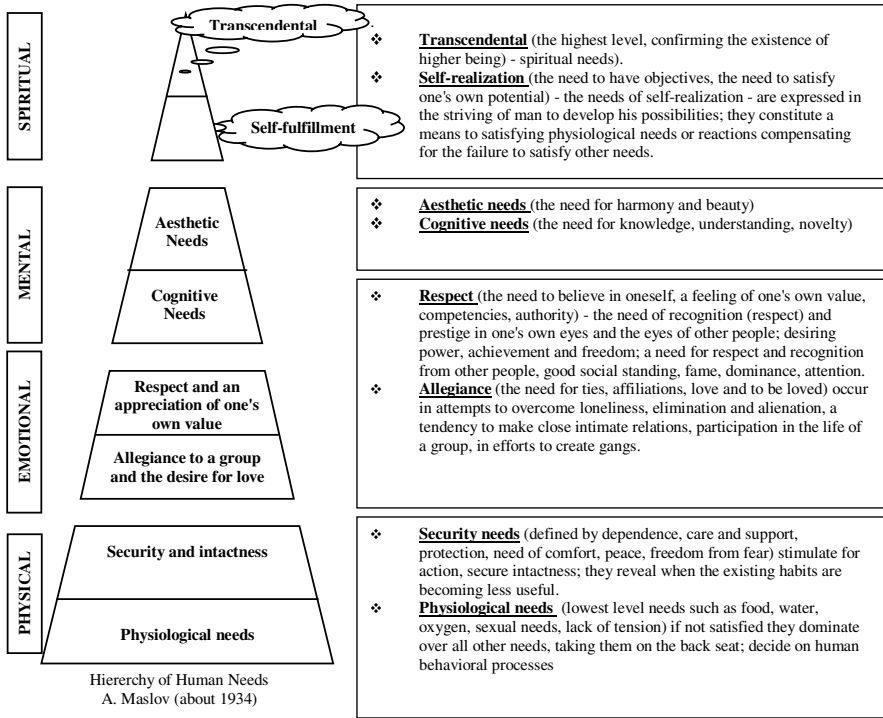
Fig. 2 The Maslov Hierarchy of human needs (about 1934) as an example of judge hierarchy of habit controls

possible for agent to answer the following questions: (i) How to currently construct the most important priority list of problems to be solved? (ii) How to solve the top priority problems under real life constraints? (iii) What to do next?

One of the main barriers hindering an acceleration in the development of Wistech applications lies in developing satisfactory computational models implementing the functioning of *adaptive judgment*. This difficulty primarily consists in overcoming the complexity of integrating the local assimilation and processing of changing non-crisp and incompletely specified concepts necessary to make correct judgments. In other words, we are only able to model tested phenomena using local (subjective) models and interactions between them. In practical applications, usually, we are not able to give perfect global models of analyzed phenomena. However, we can only approximate global models by integrating the various incomplete perspectives of problem perception.

Wisdom techniques include approximate reasoning by agents or teams of agents about vague concepts concerning real-life dynamically changing, usually distributed, systems in which these agents are operating. Such systems consist of other autonomous agents operating in highly unpredictable environments and interacting with each other.

Wistech is based on techniques of reasoning about knowledge, information and data which helps apply the current knowledge in problem solving in real-life highly unpredictable environments and autonomous multiagent systems. This includes such methods as identification of the current situation on the basis of interactions or dialogs, extraction of relevant fragments of knowledge from knowledge networks, judgment for prediction for relevant actions or plans in the current situation, or judgment of the current plan reconfiguration.

In [18, 19, 20, 21] Wisdom Technology (Wistech) is discussed as one of the main paradigms for development of new applications in intelligent systems.

Gottfried Wilhelm Leibniz should be considered a precursor of modern *Granular Computing* (GC) understood as a calculus of human thoughts [23, 24, 20, 21]. Through centuries mathematicians have been developing tools to deal with such a calculus. Unfortunately, the tools developed in *crisp* mathematics, in particular, in classical mathematical logic do not yet allow for the understanding natural language used by humans to express thoughts and reasoning about theses thoughts, an understanding which will allow us to construct truly intelligent systems.

One of the reasons is that humans, capable of efficiently solving many real-life problems, are able to express their thoughts by means of vague, uncertain, imprecise concepts and reason with such concepts. Lotfi Zadeh proposed to base the calculus of thoughts using fuzzy logic to move from computing with numbers to computing with words, from manipulations of measurements to manipulations of perceptions, and further to Granular Computing. This idea has been developed by Lotfi Zadeh himself in a number of papers (see, e.g., [63, 64, 65, 62, 67]) and by other researchers, also using rough set methods (see, e.g., [35, 42]).

Solving complex problems, e.g., by multi-agent systems requires new approximate reasoning methods based on new computing paradigms. One such recently emerging computing paradigm is *Rough Granular Computing* RGC (see, e.g., [42]).

The research on the foundations on RGC is based on the rough set approach. The rough set concept, due to Pawlak [38, 39, 41] is based on classical two valued logic. The rough set approach has been developed to deal with uncertainty and vagueness. The approach makes it possible to reason about the approximations of vague concepts. These approximations are temporary, subjective, and change adaptively with changes in environments [11, 47, 49].

In this paper, we discuss some basic issues on RGC emphasizing the role of hierarchical modeling of granular structures (Sections 2-6) and, in particular, some issues on interactive granular computations (Sect. 6).

2   Granular Computing and Rough-Granular Computing in Hierarchical Learning

The hierarchical learning approach takes advantage of additional domain knowledge provided by human experts. In order to best employ this knowledge, it relies on the observation that human thinking and perception in general, and their reasoning while performing classification tasks in particular, can:

- inherently comprise different levels of abstraction,
- display a natural ability to switch focus from one level to another,
- operate on several levels simultaneously.

Such processes are natural subjects for the GC paradigm, which encompasses theories, methods, techniques and tools for such fields as problem solving, information processing, human perception evaluation, analysis of complex systems and many others.

The concept of information granules is closely related to the imprecise nature of human reasoning and perception. GC therefore provides excellent tools and methodologies for problems involving flexible operations on imprecise or approximated concepts expressed in natural language.

One of the possible approaches in developing methods for compound concept approximations can be based on the layered (hierarchical) learning [55, 12]. Inducing concept approximation should be developed hierarchically starting from concepts that can be directly approximated using sensor measurements toward compound target concepts related to perception. This general idea can be realized using additional domain knowledge represented in natural language. For example, one can use some rules of behavior on the roads, expressed in natural language, to assess from recordings (made, e.g., by camera and other sensors) of actual traffic situations, if a particular situation is safe or not (see, e.g., [29, 9, 8, 14]). The hierarchical learning has been also used for identification of risk patterns in medical data and extended for therapy planning (see, e.g. [7, 6]). Another application of hierarchical learning for sunspot classification is reported in [32]. To deal with such problems one should develop methods for concept approximations together with methods aiming at approximation of reasoning schemes (over such concepts) expressed in natural language. The foundations of such an approach, called Rough-Granular Computing, creating a core of perception logic, are based on rough set theory [39, 41, 14] and its extension rough mereology [44, 48, 35]. The (approximate) Boolean reasoning methods can be scaled to the case of compound concept approximation.

RGC is an approach to the constructive definition of computations over objects, called granules, aiming at searching for solutions of problems which are specified using vague concepts. Computations in RGC are performed on granules representing often vague, partially specified, and compound concepts delivered by agents engaged in tasks such as knowledge representation, communication with other agents, and reasoning. Granules are obtained through the process of granulation (degranulation). Granulation can be viewed as a human way of achieving data compression and it plays a key role in implementing the divide-and-conquer strategy in human problem-solving [64, 67]. The approach combines rough set methods with other soft computing methods, and methods based on granular computing. RGC is used for developing one of the possible Wistech foundations based on approximate reasoning using vague concepts. The RGC approach combines rough set methods with methods based on granular computing [2, 42, 67], borrowing also from other soft computing paradigms.

Let us observe that hierarchical modeling employs some general mechanisms emphasized in [19] dealing with a kind of 'interplay' between syntax and semantics. The

key observation is that the syntax on one level is used to define semantical structures (or their clusters) on the next level of hierarchy. One can interpret them in the framework of the Bairwise classifications [3] as operations on such classifications or as a kind of sums of information systems [50]. They allow us gradually to model structures of granules representing *wider* context of perceived objects. In this way, it is possible to construct more compound granules interpreted, e.g., as patterns representing properties of, e.g., time windows of states, sequences of such time windows, sets of such sequences, etc.

3 Hierarchical Modeling of Granule Structures

Modeling relevant granules such as patterns, approximation spaces, clusters or classifiers starts from relational structures corresponding to their attributes. On can distinguish two kinds of attributes. The attributes of the first kind are like sensors, their values are obtained as the result of interaction of (the agent possessing them) with the environment. The attribute of the second kind are defined over already defined attributes. For any attribute (feature) a we consider a relational structure $\mathscr{R}_a = (V_a, \{r_i\}_{i \in I})$, where V_a is a set of values of the attribute a. Examples of such relational structures defined over the attribute-value set V_a are: $(V_a, =), (V_a, \leq)$, where $\leq$ is a linear order on V_a, or $(V_a, \leq, +, \cdot, 0, 1)$, where $V_a = \mathbb{R}$ and $\mathbb{R}$ is the set of reals. Certainly, V_a may consists complex values, e.g., relational structures. By L_a we denote a set of formulas interpreted over $\mathscr{R}_a$ as subsets of V_a. It means that if $\alpha \in L_a$ then its semantics (an object corresponding to its meaning) $\|\alpha\|_{\mathscr{R}_a}$ is a subset of V_a. Let us note that one can define attributes by sets of formulas. To explain this idea, we assume that $\mathscr{F}_a \subseteq L_a$ is a set of formulas satisfying the following two conditions: (i) for any $v \in V_a$ there exits $\alpha \in \mathscr{F}_a$ true for v; and (ii) for any two different formulas from $\mathscr{F}_a$ their conjunction is false for any $v \in V_a$. Hence, for any $v \in V_a$ there is a unique formula in $\mathscr{F}_a$ true on v. One can consider an example of discretization of $\mathbb{R}$ by formulas $\alpha_1, \ldots, \alpha_k$ with interpretation over $\mathscr{R}_a = (\mathbb{R}, \leq, +, \cdot, 0, 1)$, where $\|\alpha_i\|_{\mathscr{R}_a}$ for $i = 1, \ldots, k$ create a partition of $\mathbb{R}$ into intervals.

If $\mathscr{A} = (U, A)$ is an information system and $a \in A$ then $\|\alpha\|_{\mathscr{R}_a}$ can be used to define semantics of α over $\mathscr{A}$ by assuming $\|\alpha\|_{\mathscr{A}} = \{x \in U : a(x) \in \|\alpha\|_{\mathscr{R}_a}\}$. Hence, any formula α can be treated as a new binary attribute of objects from U (see Fig. 3). If $\mathscr{A}^* = (U^*, A^*)$ is an extension of $\mathscr{A} = (U, A)$, i.e., $U \subseteq U^*$, $A^* = \{a^* : a \in A\}$, and $a^*(x) = a(x)$ for $x \in U$, then $\|\alpha\|_{\mathscr{A}} \subseteq \|\alpha\|_{\mathscr{A}^*}$.

In the next step of modeling, relational structures corresponding to attributes can be fused. Let us consider an illustrative example. We assume $\mathscr{R}_{a_i} = (V_{a_i}, r_{\mathscr{R}_{a_i}})$ are relational structures with binary relation $r_{\mathscr{R}_{a_i}}$ for $i = 1, \ldots, k$. Then, by $\mathscr{R}_{a_1} \times \cdots \times \mathscr{R}_{a_k}$ we denote their fusion defined by a relational structure over $(V_{a_1} \times \cdots \times V_{a_k})^2$ consisting of relation $r \subseteq (V_{a_1} \times \cdots \times V_{a_k})^2$ such that for any $(v_1, \ldots, v_k), (v'_1, \ldots, v'_k) \in V_{a_1} \times \cdots \times V_{a_k}$ we have $(v_1, \ldots, v_k) r (v'_1, \ldots, v'_k)$ if and only if $v_i r_{\mathscr{R}_{a_i}} v'_i$ for $i = 1, \ldots, k$. One can extend this example by imposing some additional constraints.

In the process of searching for (sub-)optimal approximation spaces, different strategies may be used. Let us consider an example of such strategy presented in [53]. In this example, $DT = (U, A, d)$ denotes a decision system (a given sample of
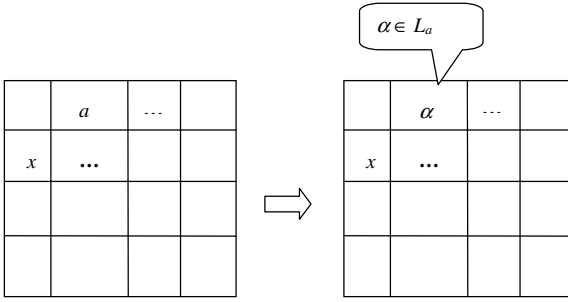
Fig. 3 New attribute defined by a formula α from L_a
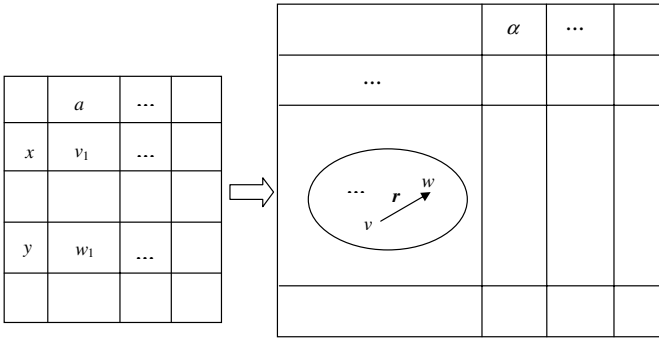
Fig. 4 Granulation to tolerance classes. r is a similarity (tolerance) relation defined over signatures of objects

data), where U is a set of objects, A is a set of attributes and d is a decision. We assume that for any object $x \in U$, only partial information equal to the A-signature of x (object signature, for short) is accessible, i.e., $Inf_A(x) = \{(a, a(x)) : a \in A\}$. Analogously, for any concept we are only given a partial information about this concept by means of a sample of objects, e.g., in the form of decision table. One can use object signatures as new objects in a new relational structure $\mathscr{R}$. In this relational structure $\mathscr{R}$ some relations between object signatures are also modeled, e.g., defined by the similarities of these object signatures (see Fig. 4).

Discovery of relevant relations between object signatures is an important step in searching for relevant approximation spaces. In this way, a class of relational structures representing perception of objects and their parts is constructed. In the next step, we select a language $\mathscr{L}$ consisting of formulas expressing properties over the defined relational structures and we search for relevant formulas in $\mathscr{L}$. The semantics of formulas (e.g., with one free variable) from $\mathscr{L}$ are subsets of object signatures. Note, that each object signature defines a neighborhood of objects from a given sample (e.g., decision table DT) and another set on the whole universe of objects being an extension of U. Thus, each formula from $\mathscr{L}$ defines a family of sets of objects over the sample and also another family of sets over the universe of all
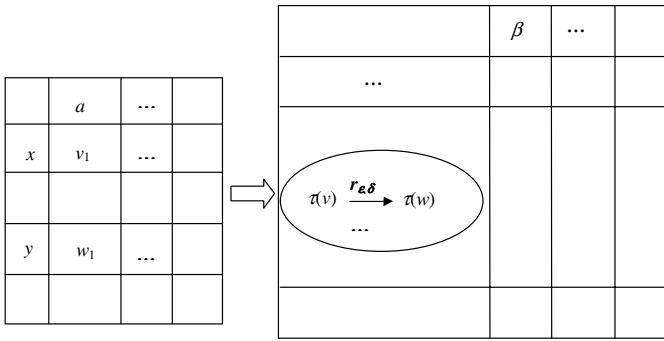
Fig. 5 Granulation of tolerance relational structures to clusters of such structures. $r_{\varepsilon,\delta}$ is a relation with parameters ε, δ on similarity (tolerance) classes

objects. Such families can be used to define new neighborhoods for a new approximation space by, e.g., taking their unions. In the process of searching for relevant neighborhoods, we use information encoded in the available sample. More relevant neighborhoods make it possible to define more relevant approximation spaces (from the point of view of the optimization criterion). Following this scheme, the next level of granulation may be related to clusters of objects (relational structures) for a current level (see Fig. 5).

In Fig. 5 τ denotes a similarity (tolerance) relation on vectors of attribute values, $\tau(v) = \{u : v \ \tau \ u\}$, $\tau(v) \ r_{\varepsilon,\delta} \ \tau(w)$ iff $\mathrm{dist}(\tau(v), \tau(w)) \in [\varepsilon - \delta, \varepsilon + \delta]$, and $\mathrm{dist}(\tau(v), \tau(w)) = \inf\{\mathrm{dist}(v', w') : (v', w') \in \tau(v) \times \tau(w)\}$ where dist is a distance function on vectors of attribute values.

One more example is illustrated in Fig. 6, where the next level of hierarchical modeling is created by defining an information system in which objects are time windows and attributes are (time-related) properties of these windows.

It is worth mentioning that quite often this searching process is even more sophisticated. For example, one can discover several relational structures (e.g., corresponding
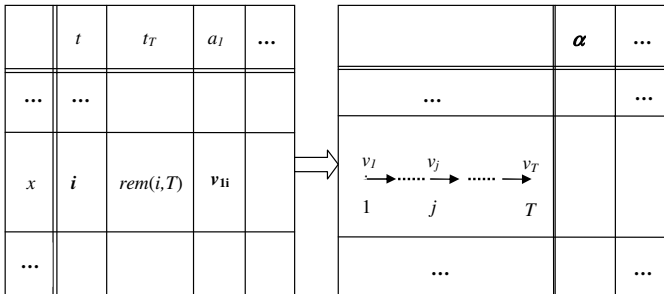
Fig. 6 Granulation of time points into time windows. T is the time window length, $v_j = (v_{1j}, \ldots, v_{Tj})$ for $j = 1, \ldots, T$, $rem(i, T)$ is the remainder from division of i by T, α is an attribute defined over time windows
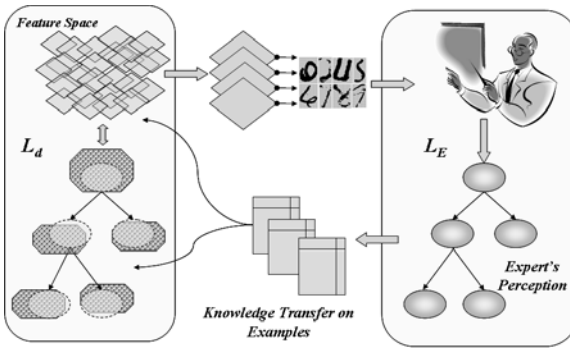
Fig. 7 Expert's knowledge elicitation

to different attributes) and formulas over such structures defining different families of neighborhoods from the original approximation space. As a next step, such families of neighborhoods can be merged into neighborhoods in a new, higher degree approximation space.

The proposed approach is making it possible to construct information systems (or decision tables) on a given level of hierarchical modeling from information systems from lower level(s) by using some constraints in joining objects from underlying information systems. In this way, structural objects can be modeled and their properties can be expressed in constructed information systems by selecting relevant attributes. These attributes are defined by means of a language that makes use of attributes of systems from the lower hierarchical level as well as relations for defining constraints [50, 48, 3]. In some sense, the objects on the next level of hierarchical modeling are defined using the syntax from the lover level of the hierarchy. Domain knowledge is used to aid the discovery of relevant attributes (features) on each level of hierarchy. This domain knowledge can be provided, e.g., by concept ontology together with samples of objects illustrating concepts from this ontology. Such knowledge is making it feasible to search for relevant attributes (features) on different levels of hierarchical modeling (see Sect. 4). In Fig. 7 we symbolically illustrate the transfer of knowledge in a particular application. It is a depiction of how the knowledge about outliers in handwritten digit recognition is transferred from expert to a software system. We call this process *knowledge elicitation*. Observe, that the explanations given by expert(s) are expressed using a subset of natural language limited by using concepts from provided ontology only. Concepts from higher levels of ontology are gradually approximated by the system from concepts on lower levels. This kind of approach is typical for hierarchical modeling [8]. This is, in particular, the case when we search for a relevant approximation space for objects composed from parts for which some approximation spaces, relevant to components, have already been found. We find that hierarchical modeling is required for approximation of complex vague concepts, as in [33, 43].

4   Ontologies as Complex Granules and Their Approximation in RGC

Approximation of complex, possibly vague concepts requires a hierarchical modeling and approximation of more elementary concepts on subsequent levels in the hierarchy along with utilization of domain knowledge. Due to the complexity of these concepts and processes on top levels in the hierarchy one can not assume that fully automatic construction of their models, or the discovery of data patterns required to approximate their components, would be straightforward. We propose to include in this process the discovery of approximations of complex vague concepts, performed interactively with co-operation of domain experts. Such interaction allows for more precise control over the complexity of discovery process, therefore making it computationally more feasible. Thus, the proposed approach transforms a typical data mining system into an equivalent of experimental laboratory (in particular, for *ontology approximation*) in which the software system, aided by human experts, attempts to discover: (i) approximation of complex vague concepts from data under some domain constraints, (ii) patterns relevant to user (researcher), e.g., required in the approximation of vague components of complex concepts.

The research direction aiming at interactive knowledge construction has been pursued by our team, in particular, toward the construction of classifiers for complex concepts (see, e.g., [8, 10, 7, 6, 5, 4] and also [14, 29, 30, 31, 32]) aided by domain knowledge integration. Advances in recent years indicate a possible expansion of the research conducted so far into discovery of models for processes involving complex objects from temporal or spatio-temporal data.

The novelty of the proposed RGC approach for the discovery of approximations of complex concepts from data and domain knowledge lies in combining, on one side, a number of novel methods of granular computing developed using the rough set methods and other known approaches to the approximation of vague, complex concepts (see, e.g., [8, 10, 11, 7, 6, 5, 4, 20, 29, 30, 31, 32, 38, 39, 41, 42, 66, 67]) with, on the other side, the discovery of structures from data through an interactive collaboration with domain experts (see, e.g., [8, 10, 11, 7, 6, 5, 4, 20, 29, 30, 31, 32, 42]). The developed methodology based on RGC was applied, to various extent, in real-life projects including: unmanned area vehicle control, robotics, prediction of risk patterns from temporal medical and financial data, sunspot classification, and bioinformatics. For technical details please refer to [8, 10, 11, 7, 6, 5, 4] and [20, 29, 30, 31, 32, 42]).

5   Toward RGC for Process Mining

The rapid expansion of the Internet has resulted not only in the ever growing amount of data therein stored, but also in the burgeoning complexity of the concepts and phenomena pertaining to those data. This issue has been vividly compared in [16] to the advances in human mobility from the period of walking afoot to the era of jet travel. These essential changes in data have brought new challenges to the development

of new data mining methods, especially that the treatment of these data increasingly involves complex processes that elude classic modeling paradigms. Types of datasets currently regarded 'hot', like biomedical, financial or net user behavior data are just a few examples. Mining such temporal or complex data streams is on the agenda of many research centers and companies worldwide (see, e.g., [1, 46]). In the data mining community, there is a rapidly growing interest in developing methods for *process mining*, e.g., for discovery of structures of temporal processes from observations (recorded data). Works on process mining, e.g., [13, 26, 59, 61] have recently been undertaken by many renowned centers worldwide[1]. This research is also related to functional data analysis (cf. [45]), cognitive networks (cf. [37]), and dynamical system modeling in biology [15].

In [28, 27] we outlined an approach to discovery of processes from data and domain knowledge which is based on RGC philosophy.

In Sect. 6, we discuss some issues related to granule interactions also in process mining.

6   Toward Interactive RGC

Interactions between granules are rudimentary for understanding the nature of *interactive computations* [17]. In the RGC framework, it is possible to model interactive computations performed on granules of different complexity aiming at construction of approximations of complex vague concepts. Approximations of such concepts are capable of adaptive adjustment with the changes of underlying data and domain knowledge. Hence, the decision making algorithm based on the approximation of such vague concepts is also adaptively changing. Hence, our decision making algorithms are different from the classical algorithms which 'are metaphorically dump and blind because they cannot adapt interactively while they compute' [60].

In this section, we discuss some examples of interactions of granules showing the richness and complexity of granule interactions which should be modeled in RGC. The first example is related to discovery of concurrent systems from information systems.

Back in 1992, Zdzisław Pawlak (cf. [40]) proposed to use data tables (information systems) as specifications of concurrent systems. In this approach, any information system can be considered as a representation of a (traditional) concurrent system: attributes are interpreted as local processes of the concurrent system, values of attributes – as states of these local processes, and objects – as global states of the considered system. Several methods for synthesis of concurrent systems from data have been developed (see, e.g., [36, 51, 52, 57]). These methods are based on the following steps. First, for a given information system S we generate its (formal) theory $Th(S)$ consisting of a set of selected rules over descriptors defined by this system.

---

[1] http://www.isle.org/~langley/,
http://soc.web.cse.unsw.edu.au/bibliography/discovery/
index.html

These rules describe the coexistence constraints of local states within global states specified by *S*. Next, we define a maximal extension *Ext(S)* of *S* consisting of all objects having descriptions consistent with all rules in *Th(S)*. Finally, a Petri net with the set of reachable markings equal to *Ext(S)* is generated. There have been also developed methods for synthesis of Petri nets from information systems based on decomposition of information systems into the so called components defined by reducts. This approach is making it possible to represent a given information system by a set of interacting local processes defined by some functional dependencies extracted from data. Interactions between local processes are represented by rules over descriptors extracted from data too. It is worth mentioning that the ability to produce from an information system a structure that is essentially (is similar to) a Petri net brings significant profits. Petri nets and similar structures have been studied for decades, and nowadays we have quite potent collection of tools that make use of these notions, at our disposal.

Our second example is related to learning of state changes for agents interacting with dynamically changing environments. One possible approach can be analogous to modeling by differential equations. However, instead of assuming the definition of the functions describing these changes we propose to approximate these functions from experimental data using domain knowledge [28, 27].

Let us assume that changes of the environment state $e(t)$ and agent state $s(t)$ interacting over time *t* are described by the following scheme of equations:

$$\Delta s(t) = F(t, \Delta t, s(t), e(t)), \tag{2}$$
$$\Delta e(t) = G(t, \Delta t, s(t), e(t)).$$

Because the functions F, G are often highly nonlinear one can hardly expect that assuming some linear models one can obtain satisfactory solutions by tuning parameters in these models [15, 37, 22].

Due to uncertainty of information about states $s(t), e(t)$ one can only look for approximations of functions F, G from available data. When we learn approximations of functions F, G it is necessary to develop methods for computing approximations of trajectories of solutions based on interaction of approximations of functions F, G with granules representing uncertain information about states. We couple of function approximations with descriptions of indiscernibility (similarity) classes in which the current state is included in order to identify indiscernibility (similarity) classes for the next state(s). This requires some special interaction of granule representing uncertain information about the current state and the granule represented by approximation of functions describing changes between consecutive states. First, the granule of object is interacting with components of function approximation. This step is, in some sense, analogous to fuzzification in fuzzy control. In the case of rule based classifier, this step involves search for inclusion degrees of object granule and patterns represented by the left hand sides (antecedents) of rules. This may be perceived as matching membership degrees in fuzzy controller. Finally, the results of the interaction are fused to form a granule representing the next state. Again, this step is analogous to defuzzification in fuzzy controller. In the case of rule based

classifier, this step is based on the conflict resolution strategy or voting strategy making it possible to select or construct the final decision granule in presence of possibly contradictory, partially matching rules. We perceive the idea described above as very important direction for further research on methods for discovery of process trajectory approximation from data and domain knowledge.

More advanced interaction of processes may occur if we consider the situation when each path in a given process is represented by a vector of attribute values. Such a situation may occur when, for instance, paths from the lower level undergo clustering. Then, some additional constraints can be related to paths of the resulting process constructed from paths of interacting, lower-level processes. They may represent results of synchronization of two or more processes. For example, in any path of the process obtained as a result of interaction between two lower-level processes states with a certain distinguished property should separate (appear in-between) states with another specific property.

It should be noted that in practical approaches to modeling it is often necessary to use relevant names (labels) for the constructed processes, tantamount to their position and rôle in concept hierarchy (or corresponding ontology). To answer to this requirement one may use methods of inducing, e.g., Petri nets from examples of paths (see, e.g., [26]).

Another way of looking at modeling of interactions is by employing the agent-oriented framework. The depiction of agents' interactions with environment(s) is essentially based on observation, that each agent perceives only a partial (and possibly vague) information about environment. On the basis of the perceived information and its own state the agent derives (creates) some granules, with the goal of changing the state of environment to its favor. These granules are involved in interactions with the environment and granules originating in other agents. Using either competitive or cooperative strategies (coalitions of) agents involved in interactions form a resulting action which changes the environment(s) in a way that is in some accordance with components (agent-specific granules). The approaches that use elements of such interactive agent co-operation are nowadays popular in multiagent systems [25, 56].

In the following, final example we describe an application of domain knowledge in modeling of interactions. We use sentences from (a limited subset of) the natural language coupled with so called *behavioral graphs* [8] to define relationships (interactions) that occur between parts of a complex object. In this example we show such description for the task of recognizing whether at a given moment the observed road situation leads to imminent danger or not. The modeling of the system that ultimately is capable of recognizing the extremely compound concept of *dangerous situation* on the basis of low-level measurements, is indeed hierarchical. In Fig. 8 we present a behavioral graph for a single object-vehicle on a road. This behavioral graph appears in between the lowest level (sensor measurements) and the highest level (dangerous situation) in the hierarchy of concepts.

A composition of behavioral graphs, appearing on lower level in the hierarchy, can be used to represent behavior (and interaction) of a more compound part consisting of, e.g., two vehicles involved in the maneuver of overtaking (see Fig. 9).
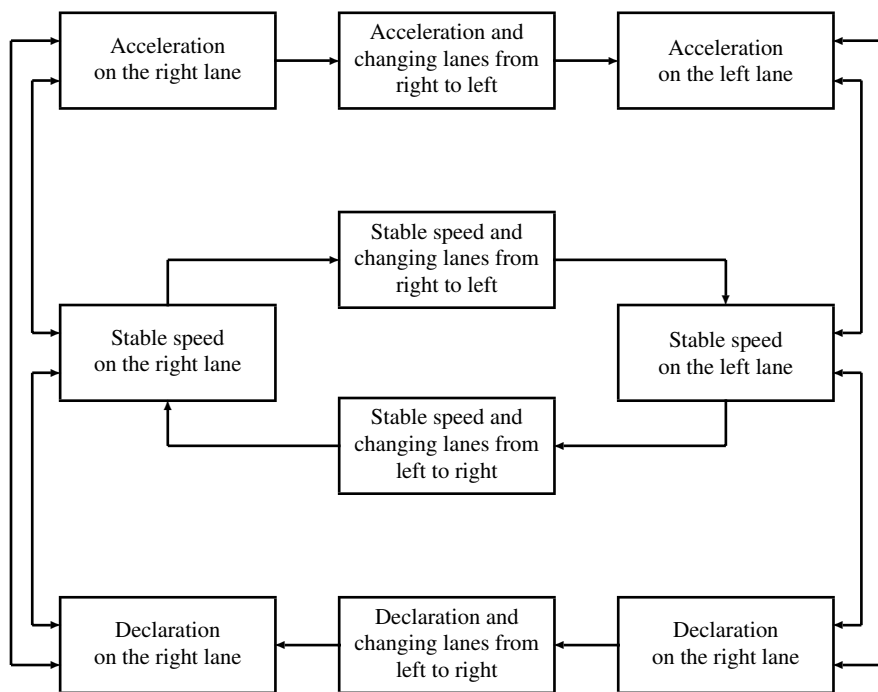
Fig. 8 A behavioral graph for a single object-vehicle

Please note, that the concept of *overtaking* is built of components which at some point were also approximated from the lower level concepts. This is a case of, e.g., *changing lane* or *A passing B* (refer to Fig. 9).

The identification of the behavioral pattern of a complex object on the basis of sensory data cannot go forward without (approximation of) ontology of concepts. It is this ontology that makes it possible to link the low level measurements (sensory concepts) with the high level description of behavioral patterns [8, 10, 11, 7, 5, 4, 20, 42]. By means of this ontology we establish that – following our road example – in order to know what the *overtaiking* is, one has to define a concept of *A passing B*, as well as link both *A* and *B* to an object-vehicle structure (see Fig. 8).

An example of behavioral graphs for medical application [4] is presented in Fig. 10. Behavioral graphs based on domain knowledge were also used in risk analysis for medical data [8, 10, 11, 7, 5, 4, 20, 42].

Models of behavioral patterns can also be represented by Petri nets, differential equations or using relevant formal theories. Such patterns are used in further steps for modeling more compound processes obtained by interaction of patterns representing local processes. It is important to note that one can hardly expect to discover such models fully automatically from data, without cooperation with experts.

The *RoughIce* platform for hierarchical modeling, in particular for modeling of interactions is available at `logic.mimuw.edu.pl/~bazan/roughice/`.

| 1. Vehicle A is behind B on the right lane | | 6. Vehicle A is before B on the right lane |
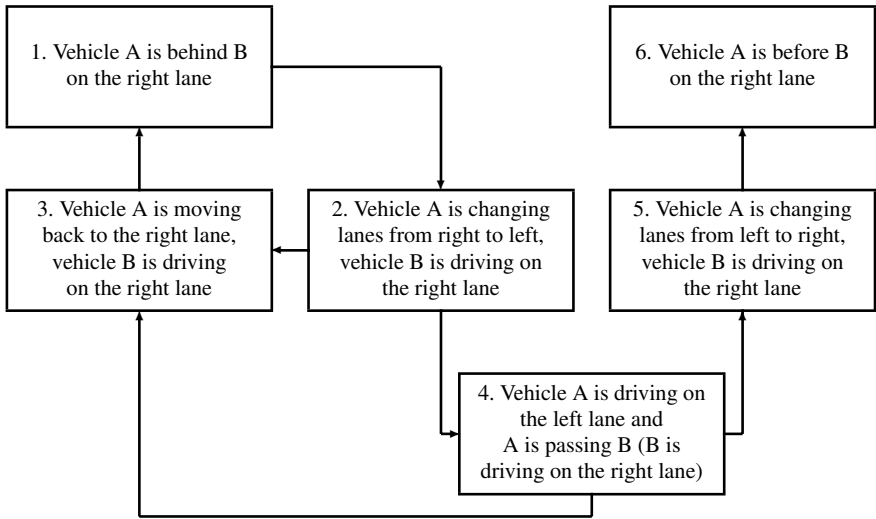
Fig. 9 A behavioral graph for the maneuver of overtaking
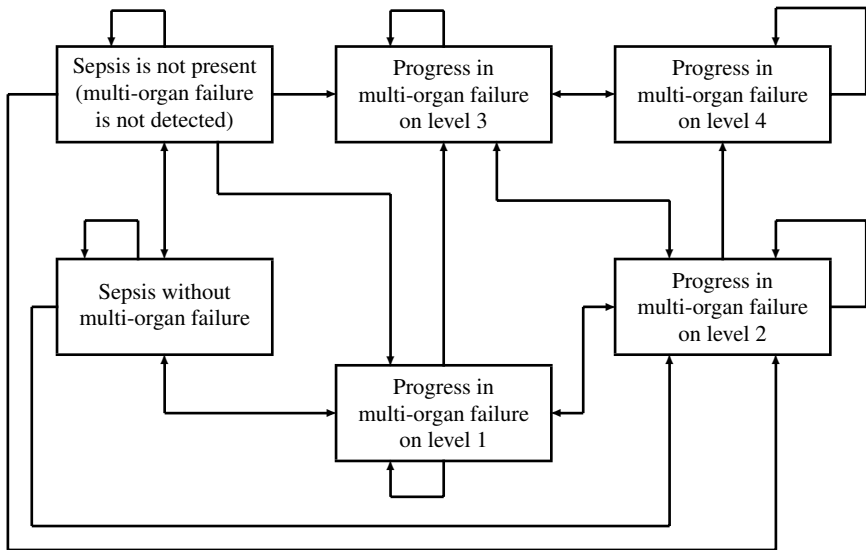
Fig. 10 A behavioral graph of sepsis by analyzing the multi-organ failure

7 Conclusions

We discussed some issues closely related to research directions within the *Wisdom Technology* (Wistech) research programme, as outlined recently in [18, 19, 20, 21]. There are possible different ways to build computational models that are based on

Wistech philosophy. We outlined some steps of such modeling for just one of them, which is based on the RGC approach. The approach is promising for developing new methods based on human-machine interactions.

References

1. Aggarwal, C. (ed.): Data Streams: Models and Algorithms. Springer, Berlin (2007)
2. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Dordrecht (2003)
3. Barwise, J., Seligman, J.: Information Flow: The Logic of Distributed Systems. Cambridge University Press, Cambridge (1997)
4. 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)
5. Bazan, J.: Rough sets and granular computing in behavioral pattern identification and planning. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Handbook of Granular Computing, pp. 777–800. John Wiley & Sons, New York (2008)
6. Bazan, J., Kruczek, P., Bazan-Socha, S., Skowron, A., Pietrzyk, J.J.: Automatic planning of treatment of infants with respiratory failure through rough set modeling. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 418–427. Springer, Heidelberg (2006)
7. Bazan, J., Kruczek, P., Bazan-Socha, S., Skowron, A., Pietrzyk, J.J.: Risk pattern identification in the treatment of infants with respiratory failure through rough set modeling. In: Proceedings of Information Processing and Management under Uncertainty in Knowledge-Based Systems, vol. 3, pp. 2650–2657. Editions E.D.K., Paris (2006)
8. Bazan, J., Peters, J.F., Skowron, A.: Behavioral pattern identification through rough set modelling. In: Ślęzak, D., Szczuka, M., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3642, pp. 688–697. Springer, Heidelberg (2005)
9. Bazan, J., Skowron, A.: Classifiers based on approximate reasoning schemes. In: Dunin-Kęplicz, B., Jankowski, A., Skowron, A., Szczuka, M. (eds.) Monitoring, Security, and Rescue Techniques in Multiagent Systems. Advances in Soft Computing, pp. 191–202. Springer, Heidelberg (2005)
10. Bazan, J., Skowron, A.: On-line elimination of non-relevant parts of complex objects in behavioral pattern identification. In: Pal, S.K., Bandoyopadhay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 720–725. Springer, Heidelberg (2005)
11. Bazan, J., Skowron, A., Swiniarski, R.: 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)
12. Behnke, S.: Hierarchical Neural Networks for Image Interpretation. LNCS, vol. 2766. Springer, Heidelberg (2003)
13. Borrett, S.R., Bridewell, W., Arrigo, P.L.K.R.: A method for representing and developing process models. Ecological Complexity 4(1-2), 1–12 (2007)
14. Doherty, P., Łukaszewicz, W., Skowron, A., Szałas, A.: Knowledge Representation Techniques: A Rough Set Approach. Studies in Fuzziness and Soft Computing, vol. 202. Springer, Heidelberg (2006)

15. Feng, J., Jost, J., Minping, Q.: Network: From Biology to Theory. Springer, Heidelberg (2007)
16. Friedman, J.H.: Data mining and statistics. What's the connection? - Keynote address. In: Proceedings of the 29th Symposium on the Interface: Computing Science and Statistics, Houston, US (1997)
17. Goldin, D., Smolka, S., Wegner, P.: Interactive Computation: The New Paradigm. Springer, Heidelberg (2006)
18. Jankowski, A., Skowron, A.: A wistech paradigm for intelligent systems. 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. 94–132. Springer, Heidelberg (2007)
19. 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)
20. Jankowski, A., Skowron, A.: Wisdom granular computing. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Handbook of Granular Computing, pp. 329–346. John Wiley & Sons, New York (2008)
21. Jankowski, A., Skowron, A.: Wisdom technology: A rough-granular approach. In: Festschrift dedicated to Leonard Bolc, pp. 1–40. Springer, Heidelberg (2009) (in print)
22. Kleinberg, J., Papadimitriou, C., Raghavan, P.: A microeconomic view of data mining. Data Mining and Knowledge Discovery 2, 311–324 (1998)
23. Leibniz, G.: Dissertio de Arte Combinatoria. Lepzig, Germany (1666)
24. Leibniz, G.: New Essays on Human Understanding, Cambridge, UK (1982); Written in 1705, translated and edited by Remnant, P., Bennett, J.
25. Luck, M., McBurney, P., Preist, C.: Agent technology. Enabling next generation computing: A roadmap for agent based computing (2003), www.agentlink.org
26. de Medeiros, A.K.A., Weijters, A.J.M.M., van der Aalst, W.M.P.: Genetic process mining: An experimental evaluation. Data Mining and Knowledge Discovery 14, 245–304 (2007)
27. Nguyen, H.S., Jankowski, A., Skowron, A., Stepaniuk, J., Szczuka, M.: Discovery of process models from data and domain knowledge: A rough-granular approach. In: Yao, J.T. (ed.) Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation, pp. 1–30. IGI Global, Hershey (2008) (in print)
28. Nguyen, H.S., Skowron, A.: A rough granular computing in discovery of process models from data and domain knowledge. Journal of Chongqing University of Post and Telecommunications 20(3), 341–347 (2008)
29. 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(1). LNCS, vol. 3100, pp. 187–208. Springer, Heidelberg (2004)
30. Nguyen, T.T.: Eliciting domain knowledge in handwritten digit recognition. In: Pal, S.K., Bandoyopadhay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 762–767. Springer, Heidelberg (2005)
31. Nguyen, T.T.: Outlier and exception analysis in rough sets and granular computing. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Handbook of Granular Computing, pp. 823–834. John Wiley & Sons, New York (2008)
32. Nguyen, T.T., Paddon, C.P.W.D.J., Nguyen, S.H., Nguyen, H.S.: Learning sunspot classification. Fundamenta Informaticae 72(1-3), 295–309 (2006)

33. Nguyen, T.T., Skowron, A.: Rough-granular computing in human-centric information processing. In: Bargiela, A., Pedrycz, W. (eds.) Human-Centric Information Processing Through Granular Modelling. Studies in Computational Intelligence, vol. 182, pp. 1–30. Springer, Heidelberg (2009)

34. Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.): PReMI 2005. LNCS, vol. 3776. Springer, Heidelberg (2005)

35. Pal, S.K., Polkowski, L., Skowron, A. (eds.): Rough-Neural Computing: Techniques for Computing with Words. Cognitive Technologies. Springer, Heidelberg (2004)

36. Pancerz, K., Suraj, Z.: Discovering concurrent models from data tables with the ROSECON. Fundamenta Informaticae 60(1-4), 251–268 (2004)

37. Papageorgiou, E.I., Stylios, C.D.: Fuzzy cognitive maps. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Handbook of Granular Computing, pp. 755–774. John Wiley & Sons, New York (2008)

38. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)

39. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data, System Theory. In: Knowledge Engineering and Problem Solving, vol. 9. Kluwer Academic Publishers, Dordrecht (1991)

40. Pawlak, Z.: Concurrent versus sequential the rough sets perspective. Bulletin of the EATCS 48, 178–190 (1992)

41. Pawlak, Z., Skowron, A.: Rudiments of rough sets; Rough sets: Some extensions; Rough sets and boolean reasoning. Information Sciences 177(1), 3–73 (2007)

42. Pedrycz, W., Skowron, A., Kreinovich, V. (eds.): Handbook of Granular Computing. John Wiley & Sons, New York (2008)

43. Poggio, T., Smale, S.: The mathematics of learning: Dealing with data. Notices of the AMS 50(5), 537–544 (2003)

44. Polkowski, L., Skowron, A.: Rough mereology: A new paradigm for approximate reasoning. International Journal of Approximate Reasoning 51, 333–365 (1996)

45. Ramsay, J.O., Silverman, B.W.: Applied Functional Data Analysis. Springer, Heidelberg (2002)

46. Roddick, J.F., Hornsby, K., Spiliopoulou, M.: An updated bibliography of temporal, spatial and spatio-temporal data mining research. In: Roddick, J.F., Hornsby, K. (eds.) TSDM 2000. LNCS (LNAI), vol. 2007, pp. 147–163. Springer, Heidelberg (2001)

47. Skowron, A.: Rough sets and vague concept. Fundamenta Informaticae 64, 417–431 (2005)

48. Skowron, A., Stepaniuk, J.: Information granules and rough-neural computing. In: Pal, S.K., Polkowski, L., Skowron, A. (eds.) Rough-Neural Computing: Techniques for Computing with Words, Cognitive Technologies, pp. 43–84. Springer, Heidelberg (2003)

49. Skowron, A., Stepaniuk, J., Peters, J., Swiniarski, R.: Calculi of approximation spaces. Fundamenta Informaticae 72(1-3), 363–378 (2006)

50. Skowron, A., Stepaniuk, J., Peters, J.F.: Rough sets and infomorphisms: Towards approximation of relations in distributed environments. Fundamenta Informaticae 54(2-3), 263–277 (2003)

51. Skowron, A., Suraj, Z.: Rough sets and concurrency. Bulletin of the Polish Academy of Sciences 41, 237–254 (1993)

52. Skowron, A., Suraj, Z.: Discovery of concurrent data models from experimental tables: A rough set approach. In: Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining, pp. 288–293. AAAI Press, Menlo Park (1995)

53. Skowron, A., Synak, P.: Complex patterns. Fundamenta Informaticae 60(1-4), 351–366 (2004)

54. 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) (in print)

55. Stone, P.: Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer. MIT Press, Cambridge (2000)

56. Sun, R. (ed.): Cognition and Multi-Agent Interaction. From Cognitive Modeling to Social Simulation. Cambridge University Press, Cambridge (2006)

57. Suraj, Z.: Rough set methods for the synthesis and analysis of concurrent processes. In: Polkowski, L., Lin, T., Tsumoto, S. (eds.) Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems. Studies in Fuzziness and Soft Computing, vol. 56, pp. 379–488. Springer, Heidelberg (2000)

58. Thiele, L.P.: The Heart of Judgment: Practical Wisdom, Neuroscience, and Narrative. Cambridge University Press, Edinburgh (2006)

59. Unnikrishnan, K.P., Ramakrishnan, N., Sastry, P.S., Uthurusamy, R. (eds.): Proceedings of the 4th Workshop on Temporal Data Mining: Network Reconstruction from Dynamic Data, Philadelphia, US (2006), http://people.cs.vt.edu/~ramakris/kddtdm06/

60. Wegner, P.: Why interaction is more powerful than algorithms. Communications of the ACM 40, 80–91 (1997)

61. Wu, F.X.: Inference of gene regulatory networks and its validation. Current Bioinformatics 2(2), 139–144 (2007)

62. Zadeh, L.: Foreword. In: Pal, S.K., Polkowski, L., Skowron, A. (eds.) Rough-Neural Computing: Techniques for Computing with Words, pp. IX–XI. Springer, Heidelberg (2004)

63. Zadeh, L.A.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems, Man and Cybernetics 3, 28–44 (1973)

64. Zadeh, L.A.: Fuzzy sets and information granularity. In: Gupta, M., Ragade, R., Yager, R. (eds.) Advances in Fuzzy Set Theory and Applications, pp. 3–18. North-Holland Publishing Co., Amsterdam (1979)

65. Zadeh, L.A.: Outline of a computational approach to meaning and knowledge representation based on the concept of a generalized assignment statement. In: Thoma, M., Wyner, A. (eds.) Proceedings of the International Seminar on Artificial Intelligence and Man-Machine System, pp. 198–211. Springer, Heidelberg (1986)

66. Zadeh, L.A.: A new direction in AI - toward a computational theory of perceptions. Artificial Intelligence Magazine 22(1), 73–84 (2001)

67. Zadeh, L.A.: Generalized theory of uncertainty (GTU)-principal concepts and ideas. Computational Statistics and Data Analysis 51, 15–46 (2006)

Discovering Affinities between Perceptual Granules

L_2 Norm-Based Tolerance Near Preclass Approach

James F. Peters

> *The effectiveness of the pattern recognition process depends mainly on proper representation of these patterns, using the set of characteristic features.*
>
> *– K.A. Cyran and A. Mrozek, 2001.*

Abstract. This paper proposes an approach to detecting affinities between perceptual objects contained in perceptual granules such as images with tolerance near preclasses. A *perceptual object* is something perceptible to the senses or knowable by the mind. Perceptual objects that have similar appearance are considered perceptually near each other, i.e., perceived objects that have perceived affinities or, at least, similar descriptions. A *perceptual granule* is a finite, non-empty set containing sample perceptual objects with common descriptions. Perceptual granules originate from observations of the objects in the physical world. Similarities between perceptual granules are measured within the context of what is known as a tolerance near space. This form of tolerance space is inspired by C.E. Zeeman's work on visual perception and Henri Poincaré's work on the contrast between mathematical continua and the physical continua in a pragmatic philosophy of science that laid the foundations for tolerance spaces. The perception of nearness or closeness that underlies tolerance near relations is rooted in Maurice Merleau-Ponty's work on the phenomenology of perception during the mid-1940s, and, especially, philosophical reflections on description of perceived objects and the perception of nearness. Pairs of perceptual granules such as images are considered near each other to the extent that tolerance near preclasses of sufficient magnitude can be found. The contribution of this paper is the introduction of L_2 norm-based tolerance near preclasses in detecting affinities between images.

Keywords: affinities, image, L_2 norm, perceptual granule, tolerance near preclass.

James F. Peters
Computational Intelligence Laboratory, Department of Electrical & Computer Engineering, University of Manitoba
Winnipeg, Manitoba R3T 5V6 Canada
e-mail: jfpeters@ee.umanitoba.ca

1 Introduction

This paper introduces a tolerance near preclass approach to solving the image correspondence problem, i.e., where one uses image tolerance preclasses to detect affinities between pairs of images. Recently, it has been shown that near sets can be used in a perception-based approach to discovering correspondences between images (see, e.g., [6, 7, 8, 16, 17]). Sets of perceptual objects where two or more of the objects have matching descriptions are called near sets. Detecting image resemblance and image description are part of the more general pattern recognition process enunciated by Krzysztof Cyran and Adam Mrózek in 2001 [2]. Work on a basis for near sets began in 2002, motivated by image analysis and inspired by a study of the perception of the nearness of perceptual objects carried out in cooperation with Z. Pawlak in [12]. This initial work led to the introduction of near sets [14], elaborated in [13, 16, 18]. A perception-based approach to discovering resemblances between images leads to a tolerance class form of near sets [16] that models human perception in a physical continuum viewed in the context of image tolerance spaces. A tolerance space-based approach to perceiving image resemblances hearkens back to the observation about perception made by Ewa Orłowska in 1982 [10] (see, also, [11]), i.e., classes defined in an approximation space serve as a formal counterpart of perception.

The term *tolerance space* was coined by E.C. Zeeman in 1961 in modeling visual perception with tolerances [25]. A tolerance space is a set X supplied with a binary relation $\simeq$ (i.e., a subset $\simeq \subset X \times X$) that is reflexive (for all $x \in X$, $x \simeq x$) and symmetric (i.e., for all $x, y \in X$, $x \simeq y$ implies $y \simeq x$) but transitivity of $\simeq$ is not required [23]. For example, it is possible to define a tolerance space relative to subimages of an image. This is made possible by assuming that each image is a set of fixed points. Let O denote a set of perceptual objects (e.g., gray level subimages) and let $\overline{gr}(x)$ = average gray level of subimage x. Then define the tolerance relation

$$\simeq_{\overline{gr},\varepsilon} = \{(x,y) \in O \times O \mid \; |\overline{gr}(x) - \overline{gr}(y)| \leq \varepsilon\},$$

for some tolerance $\varepsilon \in \Re$ (reals). Then $(O, \simeq_{\overline{gr},\varepsilon})$ is a sample tolerance space. Formulation of a tolerance relation is at the heart of the discovery process in searching for affinities between perceptual granules. The basic idea is to find objects such as images that resemble each other with a tolerable level of error. Sossinsky [23] observes that main idea underlying tolerance theory comes from Henri Poincaré [19]. Physical continua (e.g., measurable magnitudes in the physical world of medical imaging [4]) are contrasted with the mathematical continua (real numbers) where almost solutions are common and a given equation has no exact solution. An *almost solution* of an equation (or a system of equations) is an object which, when substituted into the equation, transforms it into a numerical 'almost identity, i.e., a relation between numbers which is true only approximately (within a prescribed tolerance) [23]. Equality in the physical world is meaningless, since it can never be verified either in practice or in theory. The study

of image tolerance near spaces is directly related to recent work on tolerance spaces (see, e.g., [1, 3, 4, 15, 16, 17, 20, 21, 22, 26]). The contribution of this paper is the introduction of L_2 norm-based tolerance near preclasses useful in detecting affinities between images.

This paper is organized as follows. Section 2 presents the basic framework used to define L_2 norm-based tolerance near relations (Sect. 2.2), tolerance near sets (Sect. 2.3), tolerance near preclasses (Sect. 2.4), and image resemblance measurement (Sect. 2.5).

2 L_2 Norm-Based Tolerance Near Relations

This section gives a brief review of tolerance near sets [16, 17] and introduces tolerance near preclasses. The notation used in this paper is summarized in Table 1.

Table 1 Relation Symbols

Symbol	Interpretation
$\mathbb{R}$	Set of real numbers
O	Set of perceptual objects
X	$X \subseteq O$, set of sample objects
x	$x \in O$, sample object
$\mathbb{F}$	A set of functions representing object features
$\mathscr{B}$	$\mathscr{B} \subseteq \mathbb{F}$
ϕ_i	$\phi_i \in \mathscr{B}$, where $\phi_i : O \longrightarrow \mathbb{R}$, probe function
$\phi(x)$	$\phi(x) = (\phi_1(x), \ldots, \phi_i(x), \ldots, \phi_L(x))$, description
$\langle X, \mathbb{F} \rangle$	$\phi(x_1), \ldots, \phi(x_{|X|})$, i.e., perceptual information system
ε	$\varepsilon \in [0,1]$
$\simeq_{\mathscr{B},\varepsilon}$	$= \{(x,y) \in O \times O \mid \parallel \phi(x) - \phi(y) \parallel \leq \varepsilon\}$, perceptual tolerance relation
$\cong_{\mathscr{B},\varepsilon}$	L_2 norm-based weak tolerance nearness relation
$\bowtie_{\mathscr{B},\varepsilon}$	generic nearness relation
C^x	$\{(x,y) \in X \times Y \mid x \bowtie y\}$, tolerance near preclass
$\mathbb{C}^x$	$\{C_1^x, \ldots, C_i^x, \ldots, C_k^x\}$, collection of tolerance near preclasses
$x_{/\simeq_{\mathscr{B},\varepsilon}}$	$= \underset{i}{\mathrm{argmax}}\{|C_i^x| \mid C_i^x \in \mathbb{C}^x\}$, maximal tolerance near preclass
$C_i^{x,th}$	$\underset{i}{\mathrm{argmax}}\{|C_i^x| \geq th| \mid C_i^x \in \mathbb{C}^x\}$, threshold tolerance near preclass

2.1 L_2 Norm for Images

This section introduces an L_2 norm for images based on the measurment of the length of each vector of feature-value differences extracted from pairs of images. The difference d_i between image feature values is obtained using

$$d_i(\phi_i(x), \phi_i(y)) = |\phi_i(x) - \phi_i(y)|, \phi_i \in B, (x,y) \in X \times Y, i \leq |B| \,,$$

where d_i denotes the ith difference between image feature values. Let $\mathbf{d}^T, \mathbf{d}$ denote row and column vectors of features value differences, respectively, i.e.,

$$\mathbf{d}^T = (d_1, \ldots, d_k), \mathbf{d} = \begin{bmatrix} d_1 \\ \ldots \\ d_k \end{bmatrix}.$$

Finally, the overall distance is the L_2 norm $\| \mathbf{d} \|_2$ for a vector $\mathbf{d}$ of features value difference measurements, i.e.,

$$\| \mathbf{d} \|_2 = (\mathbf{d}^T \mathbf{d})^{\frac{1}{2}} = \sqrt{\sum_{i=1}^{k} d_i^2}. \tag{1}$$

In general, $\| \cdot \|_2$ denotes the length of a vector in L_2 space [9]. The particular $\| \mathbf{d} \|_2$ norm in (1) provides what can be viewed as a concrescence (gathering together of features value difference measurements in vector $\mathbf{d}$) used to measure resemblances between images. This overall distance metric provides a formal basis for a variety of similarity measures that have exhibited good performance in the image retrieval experiments reported in this paper. An important, direct benefit of (1) is that it makes it possible to test which combinations of features provide good or poor indices of image similarity or degree of dissimilarity. Obviously, some combinations of features work better than others.

2.2 Perceptual Tolerance Relation

In this section, tolerance near sets are defined within the context of a perceptual information system.

Definition 1 (Perceptual Information System). A *perceptual information system* $\langle O, \mathbb{F} \rangle$ or, more concisely, *perceptual system*, is a real valued total deterministic information system where O is a non-empty set of *perceptual objects*, while $\mathbb{F}$ a countable set of *probe functions*.

A *perceptual tolerance relation* is defined in the context of perceptual systems in (2).

Definition 2 (L_2 Norm-Based Perceptual Tolerance Relation [16, 17]). Let $\langle O, \mathbb{F} \rangle$ be a perceptual system and let $\varepsilon \in \mathfrak{R}$ (set of all real numbers). For every $\mathscr{B} \subseteq \mathbb{F}$ the perceptual tolerance relation $\simeq_{\mathscr{B}, \varepsilon}$ is defined as:

$$\simeq_{\mathscr{B}, \varepsilon} = \left\{ (x, y) \in O \times O : \| \mathbf{d} \|_2 \leq \varepsilon \right\}, \tag{2}$$

where $\| \cdot \|_2$ is the L_2 norm in (1).

Example 1 (Image Tolerance Classes). Figure 1 shows a pair of images, their tolerance class coverings (Fig. 1b, Fig. 1e) and one selected tolerance class relative to a particular image region (Fig. 1c, Fig. 1f, i.e., left eye). Let $\langle O, \mathbb{F} \rangle$ be a perceptual
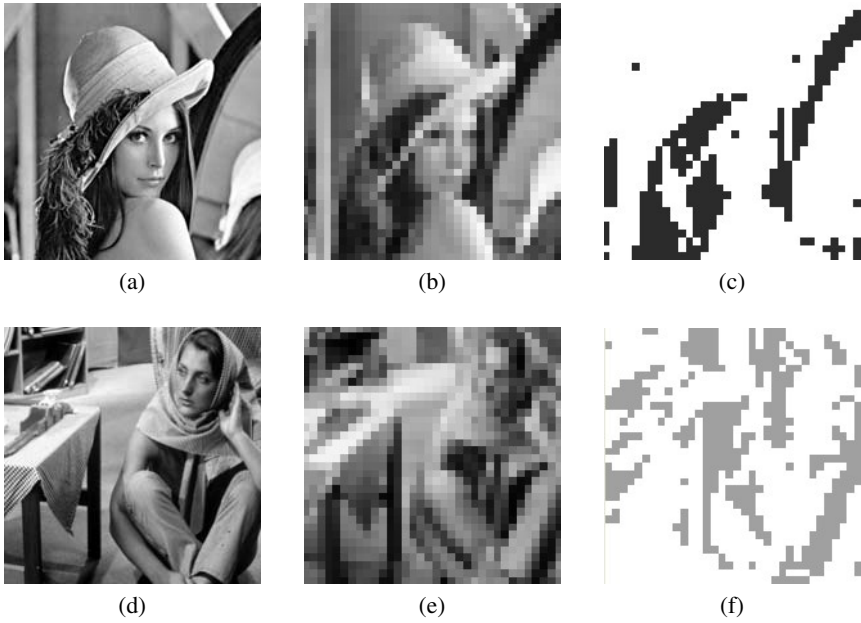
Fig. 1 Sample images and their tolerance classes. **a** Lena (L) **b** Lena Classes **c** L Eye Class **d** Barbara (B) **e** Barb Classes **f** B Eye Class

system where O denotes the set of 25×25 subimages. The image is divided into 100 subimages of size 25×25 and can be shown as a set $X = O$ of all the 100 subimages. Let $\mathscr{B} = \{\phi_1(x)\} \subseteq \mathbb{F}$, where $\phi_1(x) = \overline{gr}(x)$ is the average gray scale value of subimage x between 0 and 255. Let $\varepsilon = 25.5(10\%)$. Observe, for example, the sample tolerance class and containing subimages in Fig. 1c corresponding to Lena's left eye. Again, for example, observe the sample tolerance class and containing subimages in Fig. 1f corresponding to Barbara's left eye. Relative to the subimage containing Lena's eye and Barbara's eye, each tolerance class contains subimages where the difference between average gray scale values of the subimages and the selected subimage are within the prescribed tolerance level ε. Separate image tolerance class coverings for each image provide a basis for measuring the degree that pairs of image resemble each other.

It is now possible to define a weak tolerance nearness relation (see Def. 5), first introduced in [15].

Definition 3 (L_2 **Norm-Based Weak Tolerance Nearness Relation** [15]). Let $\langle O, \mathbb{F} \rangle$ be a perceptual system and let $\varepsilon \in \Re$ (reals). For every $\mathscr{B} \subseteq \mathbb{F}$ the L_2 norm-based weak tolerance nearness relation $\cong_{\mathscr{B},\varepsilon}$ is defined as:

$$\cong_{\mathscr{B},\varepsilon} = \left\{ (x,y) \in O \times O \mid \exists \phi_i \in \mathscr{B}.\ \| \mathbf{d} \|_2 \le \varepsilon \right\}, \tag{3}$$

where $\| \cdot \|_2$ is the L_2 norm in (1).

2.3   Tolerance Near Sets

Perceptual systems and tolerance near sets provide a feature-based solution of the image correspondence problem. The basic idea is to discover tolerance classes containing images with descriptions that differ from each other within a preset tolerance. Pairs of images X, Y with partitions defined by a tolerance relation resemble each other in the case where $X \bowtie_{\mathbb{F}} Y$ for some tolerance ε.

Definition 4 (Tolerance Near Sets [15, 16]). Let $\langle O, \mathbb{F} \rangle$ be a perceptual system and let $X \subseteq O$. A set X is a tolerance near set iff there is $Y \subseteq O$ such that $X \bowtie_{\mathbb{F}} Y$.

In effect, tolerance perceptual near sets are those sets that are defined by the nearness relation $\bowtie_{\mathbb{F}}$.

Definition 5 (Weak Tolerance Nearness [15]). Let $\langle O, \mathbb{F} \rangle$ be a perceptual system and let $X, Y \subseteq O, \varepsilon \in \mathbb{R}$. The set X is perceptually near to the set Y within the perceptual system $\langle O, \mathbb{F} \rangle$ ($X \bowtie_{\mathbb{F}} Y$) iff there exists $x \in X, y \in Y$ and there is a $\phi \in \mathbb{F}, \varepsilon \in \mathbb{R}$ such that $x \cong_{\mathscr{B}} y$. If a perceptual system is understood, then we say shortly that a set X is perceptually near to a set Y in a weak tolerance sense of nearness.

2.4   Tolerance Near Preclasses

Tolerance preclasses were introduced in [20] and elaborated in [1]. Let $\simeq$ denote a tolerance relation. A subset $C \subseteq \simeq$ defined on a set X is a preclass of the tolerance relation $\simeq$ if, and only if, $\forall \{x, y\} \subseteq C \Rightarrow x \simeq y$. A maximal preclass of a tolerance relation $\simeq$ is called a class of $\simeq$ (clique induced by $\simeq$ [24]). The main difference between classes in an equivalence relation $\sim$ and tolerance classes in $\simeq$ is the equivalence classes form a pairwise disjoint covering of X. A tolerance near preclass C^x is defined as

$$C^x = \{(x, y) \in X \times X \mid x \cong_{\mathscr{B}, \varepsilon} y\}.$$

Let $\mathbb{C}^x$ denote a collection of preclasses, i.e.,

$$\mathbb{C}^x = \{C_1^x, \ldots, C_i^x, \ldots, C_k^x\}.$$

Proposition 1. *Let $\cong_{\mathscr{B}, \varepsilon}$ be a tolerance near relation defined a set X. A subset $C^x \subseteq X$ if, and only if C^x is contained in some tolerance near class in the covering defined on X by $\cong_{\mathscr{B}, \varepsilon}$.*

We can now define a tolerance near class $x_{/\cong_{\mathscr{B}, \varepsilon}}$ in terms of a maximal tolerance near preclass, i.e.,

$$x_{/\cong_{\mathscr{B}, \varepsilon}} = \underset{i}{\operatorname{argmax}} \{|C_i^x| \; : \; C_i^x \in \mathbb{C}^x\}.$$

In terms of computational efficiency, it makes sense to introduce a threshold-based tolerance near preclass in solving image retrieval and video segmentation problems.

Definition 6 (Threshold-Based Tolerance Near Preclass). Let $\langle O, \mathbb{F} \rangle$ be a perceptual system and let $X \subseteq O$. Let $\mathbb{C}^x$ denote a collection of preclasses in a covering defined on O by a tolerance near relation $\cong_{\mathscr{B}, \varepsilon}$. Then $C_i^{x,th} \in \mathbb{C}^x$ denotes a threshold-based tolerance near preclass that is defined as:

$$C_i^{x,th} = \operatorname*{argmax}_{i,th}\{|C_i^x| \geq th| \ : \ C_i^x \in \mathbb{C}^x\}. \tag{4}$$

The assertion that images X, Y resemble each other is limited to cases where one finds a tolerance near preclass $C_i^{x,th}$ with cardinality above threshold th and stopping short of searching for maximal tolerance near class. This convention assures a more conservative approach to concluding that images are near each other at the $C_i^{x,th}$ preclass level rather concluding that images resemble each other in the minimal preclass cardinality case when $|\mathbb{C}^x| = 2$.

Example 2 (Tolerance Near Preclasses). Assume that we start with a query image X viewed as a set of subimages (denoted by $\circ$ in Fig. 2). The basic approach in detecting pairs of images that sufficient affinity to be classified as near each other (i.e., the degree of resemblance is sufficient), is to define a covering for the pair of images using the tolerance relation $\cong_{\mathscr{B}, \varepsilon}$ defined in (3) and then compare a template image Y with the query image X. In other words, we are looking for tolerance near classes containing subimages from X, Y that are within the prescribed tolerance. However, instead of searching for maximal preclasses, the search for images that resemble each other ends whenever we find one or more tolerance near preclasses $C_i^{x,th}$ defined in (4) that have sufficiently high cardinality relative to threshold th. In effect, it is asserted here that the discovery of a tolerance near preclass $C_i^{x,th}$ makes it possible to classify a pair of images as belonging to the same species. The sample scenarios in the bipartite graph in Fig. 2 illustrates this approach, where

$$C_i^{x,1} = \operatorname*{argmax}_{i,1}\{|C_i^x| \geq 1 \ : \ C_i^x \in \mathbb{C}^x\},$$

$$C_i^{x,2} = \operatorname*{argmax}_{i,2}\{|C_i^x| \geq 2 \ : \ C_i^x \in \mathbb{C}^x\},$$

$$C_i^{x,3} = \operatorname*{argmax}_{i,3}\{|C_i^x| \geq 3 \ : \ C_i^x \in \mathbb{C}^x\},$$

$$C_i^{x,4} = \operatorname*{argmax}_{i,4}\{|C_i^x| \geq 4 \ : \ C_i^x \in \mathbb{C}^x\}.$$

Fig. 2 Near Preclasses
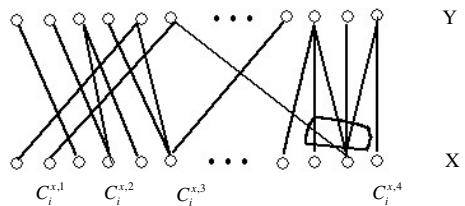
Table 2 Sample Image Nearness Measurements

Features	subimage $n \times n$	tolerance ε	$NM_{\cong_{\mathscr{B},\varepsilon}}(X,Y)$
$\overline{gr}$	25×25	0.01	0.6509 (medium nearness of subimages)
$\overline{gr},H_S$	25×25	0.01	0.2738
$\overline{gr},H_S,H_P$	25×25	0.01	0.2299
$\overline{gr}$	25×25	0.10	0.7942 (medium nearness of subimages)
$\overline{gr},H_S$	25×25	0.10	0.5835
$\overline{gr},H_S,H_P$	25×25	0.10	0.5830

2.5 *Image Resemblance Measurement*

A complete image nearness measurement system is available at [5]. For example, a nearness measure $NM_{\cong_{\mathscr{B}}}(X,Y)$ with tolerance relation $\cong_{\mathscr{B},\varepsilon}$ [4, 7] (for simplicity, we write $\cong_{\mathscr{B}}$ instead of $\cong_{\mathscr{B},\varepsilon}$) and with weights $|z_{/\cong_{\mathscr{B}}}|$ is:

$$
NM_{\cong_{\mathscr{B}}}(X,Y) = \left(\sum_{z_{/\cong_{\mathscr{B}}} \in Z_{/\cong_{\mathscr{B}}}} |z_{/\cong_{\mathscr{B}}}| \right)^{-1} \times
$$
$$
\times \sum_{z_{/\cong_{\mathscr{B}}} \in Z_{/\cong_{\mathscr{B}}}} |z_{/\cong_{\mathscr{B}}}| \frac{\min(|[z_{/\cong_{\mathscr{B}}}]_X|,|[z_{/\cong_{\mathscr{B}}}]_Y|)}{\max(|[z_{/\cong_{\mathscr{B}}}]_X|,|[z_{/\cong_{\mathscr{B}}}]_Y|)} . \tag{5}
$$

Let $\overline{gr},H_S,H_P$ denote $n \times n$ subimage average grey level, Shannon entropy, and Pal entropy, respectively. Table 2 gives a summary of measurements carried out on the pair of sample images in Fig. 2.2 and Fig. 2.2. These sample measurements do match our intuition after visual inspection of the images. Not surprisingly, the degree of image nearness decreases as the number of measured features increases. Also, changing ε from very small to larger values tends to inflate the nearness measurements as shown in Table 2.

3 Conclusion

This article introduces an approach to measuring the resemblance between pairs of images using an L_2 norm-based tolerance nearness relation. For the first time, tolerance near preclasses are introduced. It is conjectured that measuring image resemblance can be simplified and computation time reduced by stopping short of comparing maximal tolerance near preclasses made possible with the use of a threshold on preclass size. One obvious benefit of the proposed approach is a more conservative estimation of the resemblance between images. That, if all threshold-based tolerance near preclasses have cardinality below a preset threshold for a sample pair of images, then one concludes *absence of resemblance* for the sample images. This approach has promising implications for segmenting videos, especially

in applications where grouping images in a video depends on very refined measurements over many separate images contained in a video.

References

1. Bartol, W., Miró, J., Pióro, K., Rosselló, F.: On the coverings by tolerance classes. Information Sciences 166(1-4), 193–211 (2004)
2. Cyran, K.A., Mrózek, A.: Rough sets in hybrid methods for pattern recognition. International Journal of Intelligent Systems 16, 149–168 (2001)
3. Gerasin, S.N., Shlyakhov, V.V., Yakovlev, S.V.: Set coverings and tolerance relations. Cybernetics and Systems Analysis 44(3), 333–340 (2008)
4. Hassanien, A.E., Abraham, A., Peters, J.F., Schaefer, G., Henry, C.: Rough sets and near sets in medical imaging: A review. IEEE Transactions on Information Technology in Biomedicine (in press, 2009)
5. Henry, C.: Image nearness measurement system (2009),
   http://wren.ee.umanitoba.ca
6. Henry, C., Peters, J.F.: Image pattern recognition using approximation spaces and near sets. In: An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.) RSFDGrC 2007. LNCS (LNAI), vol. 4482, pp. 475–482. Springer, Heidelberg (2007)
7. Henry, C., Peters, J.F.: Near set index in an objective image segmentation evaluation framework. In: GEOgraphic Object Based Image Analysis: Pixels, Objects, Intelligence, pp. 1–6. University of Calgary, Alberta (2008)
8. Henry, C., Peters, J.F.: Perception-based image analysis. International Journal of Bio-Inspired Computation 2(2) (to appear, 2009)
9. Jänich, K.: Topology. Springer, Berlin (1984)
10. Orłowska, E.: Semantics of vague concepts. Applications of rough sets. Tech. Rep. 469, Institute for Computer Science, Polish Academy of Sciences (1982)
11. Orłowska, E.: Semantics of vague concepts. In: Dorn, G., Weingartner, P. (eds.) Foundations of Logic and Linguistics. Problems and Solutions, pp. 465–482. Plenum Pres, London (1985)
12. Pawlak, Z., Peters, J.F.: Jak blisko (how near). Systemy Wspomagania Decyzji I 57, 109 (2002)
13. Peters, J.F.: Near sets. General theory about nearness of objects. Applied Mathematical Sciences 1(53), 2609–2629 (2007)
14. Peters, J.F.: Near sets. Special theory about nearness of objects. Fundamenta Informaticae 76, 1–27 (2007)
15. Peters, J.F.: Discovery of perceptually near information granules. In: Yao, J.T. (ed.) Novel Developements in Granular Computing: Applications of Advanced Human Reasoning and Soft Computation. Information Science Reference, Hersey (in press, 2009)

16. Peters, J.F.: Tolerance near sets and image correspondence. International Journal of Bio-Inspired Computation 1(4), 239–245 (2009)
17. Peters, J.F., Ramanna, S.: Affinities between perceptual granules: Foundations and perspectives. In: Bargiela, A., Pedrycz, W. (eds.) Human-Centric Information Processing Through Granular Modelling SCI 182, pp. 49–66. Springer, Heidelberg (2009)
18. Peters, J.F., Wasilewski, P.: Foundations of near sets. Information Sciences. An International Journal (in press, 2009)
19. Poincaré, H.: The topology of the brain and the visual perception. In: Fort, K.M. (ed.) Topology of 3-manifolds and Selected Topics, pp. 240–256. Prentice Hall, New Jersey (1965)
20. Schroeder, M., Wright, M.: Tolerance and weak tolerance relations. Journal of Combinatorial Mathematics and Combinatorial Computing 11, 123–160 (1992)
21. Shreider, Y.A.: Tolerance spaces. Cybernetics and Systems Analysis 6(12), 153–758 (1970)
22. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27(2-3), 245–253 (1996)
23. Sossinsky, A.B.: Tolerance space theory and some applications. Acta Applicandae Mathematicae: An International Survey. Journal on Applying Mathematics and Mathematical Applications 5(2), 137–167 (1986)
24. Street, A.P., Wallis, W.D.: Combinatorics: A first course. The Charles Babbage Research Centre, Winnipeg (1982)
25. Zeeman, E.C.: The topology of the brain and the visual perception. In: Fort, K.M. (ed.) Topology of 3-manifolds and Selected Topics, pp. 240–256. Prentice Hall, New Jersey (1965)
26. Zheng, Z., Hu, H., Shi, Z.: Tolerance relation based granular space. In: Ślęzak, D., Wang, G., Szczuka, M.S., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS, vol. 3641, pp. 682–691. Springer, Heidelberg (2005)

Part II
Human-Computer Interactions

A Psycholinguistic Model of Man-Machine Interactions Based on Needs of Human Personality

Adrian Horzyk and Ryszard Tadeusiewicz

Abstract. Fast development of internet services together with a need to automate maintains of internet services in order to reduce expenses force to use some artificial intelligence solutions that are able to interact between a man and a machine. Such interactions can be carried out using internet chatbots that are able to communicate with a human in natural language supplemented with voice synthesizers. A main drawback of today systems is that they do not recognize nor understand and weakly react to human needs. A conversation will satisfy a human if some implemented algorithms will be able to passively recognize and classify human needs and adjust a conversation and reactions to these needs. This paper describes a new personality model that can be successfully used by chatbots to achieve this goal. The presented personality model figures out words, phrases and sentence constructions that can be recognized in a conversation, describes personality needs and suitable intelligent reactions to these needs in order to provide a human with satisfaction.

Keywords: human personality model, human needs, personality recognition, tuning of chatbot reactions, chatbot, chatterbot, psychology, biopsychology, artificial intelligence, psycholinguistics, computer linguistics.

1 Introduction to Chatbot Communication

Chatbot reactions would be more intelligent if it will be able to automatically and passively recognize human needs coming from personality, physiology, intelligence and a spirit and appropriately react to them. Nowadays chatbots can mainly watch

Adrian Horzyk · Ryszard Tadeusiewicz
Department of Automatics, AGH University of Science and Technology,
Mickiewicza Av. 30, 30-059 Cracow, Poland
e-mail:{horzyk, rtad}@agh.edu.pl
http://home.agh.edu.pl/~horzyk,
http://www.uci.agh.edu.pl/uczelnia/tad/dossier.php

and analyze only linguistic behaviours of a human. The behaviours can be divided into some actions and reactions. The probably most interesting human reactions comes from personality which can be defined, recognized and used to fulfill human personality needs during chatbot talks to people. Biophysiologists and biopsychologists show up that our personality is probably dependent on our brain construction and various neurotransmitters and their various synaptic receptors which differ quite widely from person to person [11]. There is more than a hundred known types of various neurotransmitters and more than a sixty known various synaptic receptors which various subsets appear in our brains in various configurations and quantities. These physiological features of our brains are hereditary, so we can observe similar personality behaviours in our ancestors and descendants, especially in our children, parents and grandparents.

There are many psychological models of human personality types, traits and models [1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14] that try to describe it on various points of view and for various applications. This paper introduces a special personality model that can be implemented in various linguistic systems like chatbots in order to automatically and passively recognize personality types and traits of a talking person. This model lets a chatbot to functionally describe and predict most probable personality actions and reactions after words, phrases, an inflection and sentence constructions that are used by an analyzed person. This personality model uses some groups of major personality behaviours that are easy to recognize. If some behaviours of a certain group occur then it is very probable that all other behaviours of these group can occur in human actions and reactions as well. Moreover, these groups of personality behaviours also define which actions and reactions are liked and disliked by a human of a certain personality type. Furthermore, various human personality needs (HPNs) can be recognized using this model. This model lets a chatbot to predict some of human actions and reactions and modify its way of talking to a human using properly tuned words, phrases and sentence construction to fulfill personality needs of an analyzed person. The groups of personality traits will be called Human Personality Types (HPTs). Each HPT can be variously intensive for each person so the HPTs should be characterized using a continuous range of intensities. Each HPT is associated with some personality needs which fulfillment is very important for each person. Many negotiations fail if personality needs are not fulfilled!

This paper describes many various needs of some introduced HPTs, e.g., will to decide and choose, will to harmonize, will to systematize and order, will to find inspiration, will to reveal and discover, will to be assured. HPTs are in various degree intensive and important to satisfy various people.

All HPTs and their needs can be automatically and passively recognized listening to words, phrases, an inflection and sentence constructions of an analyzed chatting person. These needs can be recognized also in some expressions of a body language and behaviours but they are not so easy to watch for chatbots as linguistic expressions of personality. In order to make a use of human personality for a communication of chatbots there is necessary to be able to:

- define useful HPTs,
- define HPNs associated with each HPT,

- define positive and negative actions and reactions for which each HPT is sensitive and reactive,
- find out what actions and reactions each HPT likes and dislikes,
- find out words, phrases, an inflection and sentence constructions that characterize linguistic actions and reactions to each HPT,
- define relationship between HPT intensities and quantity of used words, phrases, an inflection and sentence constructions,
- automatically and passively recognize HPTs and their intensities for a given person after linguistic expressions of HPTs during a talk,
- define predictable human reactions dependently of recognized HPTs, their intensities and various possible combinations,
- define 'magic' words, phrases, an inflection and sentence constructions that have a strong influence for each HPT and aptly act or react to human linguistic statements and questions in order to fulfill HPNs during a talk regardless of a topic of a conversation or talks,
- define chatbot reactions to various HPTs to release prospective profitable, beneficial and positive reactions of a considered person of the known HPTs.

This paper focuses and systematizes only the most representative features, needs, actions, reactions, words, phrases, an inflection, sentence constructions of the introduced HPTs that are interesting for chatbots and their psycholinguistic engines. The sets of words and phrases can be expanded after the idea of each HPT presented in this paper.

2 Description of a Personality Model

A psycholinguistic model of human personality described in this paper consists of 11 human personality types (HPTs) that represent special groups of human behaviours and human personality needs (HPNs) [10]. The HPTs and their intensities are hereditary and usually strongly control human behaviour from the whole beginning of a human life and determine directions of human development, actions and aspirations. They influence on personal criterions and decide against something that is inconsistent to personal HPTs. The ability to recognize HPTs and their needs allows a chatbot engine constructor to formulate and use appropriate algorithms to conduct negotiations, talks or more probably persuade or ask somebody to do something. The HPTs can be treated as a special kind of a human psychological operating system that makes a human to head in certain directions. The HPNs together with human physiological needs are the main reasons and motivators for a human in an early stage of life. Human intelligence gradually develops during a whole life and takes control of human behaviour. Intelligence can develop faster or slower dependent on an ability of an individual hereditary brain structure and its development and on a individual will and a human aspiration. Intelligence is closely related to ability to associate various actions, objects and their features. The associations can be more or less extended and enable people to elaborate sentences in different ways.

Table 1 The needs and likes for the HPTs

DOM	decide, choose, boss [sb] around, dominate, manage, speak and decide on behalf of himself and other people, select, rule, govern, order, make sb to do sth, determine, restrain, have got own opinion, control sb or sth, steer, drive, manipulate, conduct, handle, lead, influence, operate, persuade, teach, advise, instruct, regulate, designate, appoint, recommend, force, want, will, convince sb of his point of view; have assets, original, intact, virgin and untouched sth/sb; be kindly asked, thanked, bowed, curtsied, given way, yielded by sb
MAX	have, get, own, possess, win, obtain, observe, watch, read, imagine, create, focus, make known, do sth extreme, e.g.: unique, rare, unusual, enormous, huge, super, great, maximal or the best
INS	seek for inspiration and inspire sb with sth; look for some new associations, things, features and parameters of things, arrangements, layouts, systems, relations, feelings, experiences, behaviours, ideas, conceptions; speak about with what is he inspired, at what is he surprised or by what is he enthralled; surprises and to spring a surprise on sb; act spontaneously, differently from others and by intuition; create sth new because it supplies him with inspiration; watch, listen, touch, sniff and taste in order to observe, hear, feel or smell new things and people that can supply him with a new portion of inspiration
DIS	discover, examine, penetrate, integrate various associations and understand unknown things, actions, reactions, behaviours, relations; ask frequently himself and other people about things, actions and behaviours that he does not know nor understand; watch, listen, touch, sniff and taste in order to discover, get known and understand; research, think about sth, ponder over sth and associate facts with other facts; when something unusual, new, unknown and unordinary happens all the time; news, new technologies and their products
VER	watch, observe, listen, feel and smell in order to compare and distinguish sth from other things, actions, behaviours, differentiate between things, features of things, people, actions, relations, behaviours and discriminate them; look for distinguish marks, even the smallest differences and he watches, checkes and verifies all things and people around in order to find differences; speak, talk and write about noticed differences; precise, exact and accurate description of things, actions and tasks
SYS	organize, systematize, order, sort, arrange, classify, cluster, form a sequence or chronology, enumerate, list schedule everything, act only after planned actions or reactions; create plans, models, standards, norms, schedules, rules, regulations and to act in accordance with them; everything have its right place and time; do everything in some order; enumerate or list tasks and calculate time necessary to perform them; enumerate things, features, actions etc. emphasizing their order and sequence
ASS	protect, assure, warrant, guarantee, insure, safeguard, prevent, warn others, limit risks and threats and make precautions against them and avoid them; purchase proven and reliable products; expects and demands guarantees from others; insures himself and other people against risks and threats; safeguard and protect himself and other people; use preventive, protective and precautionary measures and recommends them to others; be prepared and know what to do when difficulties or troubles occur; have some reserves, provisions and stores
HAR	harmonize, conciliate and reconcile people, appease disputes and conflicts, resolves and tones down conflicts, make peace with others, make symbiosis between things, adapt, accommodate, adjust, tune, fit and match things together; avoids quarrels with sb, disagreements among sb, rows, brawls, flights, conflicts; alleviate stress and nervous situations, relieves, reduces, calms, appeases, assuage, eases and moderates all situations
EMP	empathy, compassionate, understand other people and their emotions, be understood by other people, be sensitive, tender, affectionate, give himself to know, bringing himself closer to somebody else, grow closer together and degrade a distance, personal conflicts; listen to secrets and intentions of other people
TAO	be subjective, practical, sensible, businesslike and matter-of-fact, quick execution and solution of tasks, come straight to the point, get to the heart of the matter or hit the nail on the head; like completed tasks, examinations and work
BAL	look after balance, counterbalance, compensation for sth, exact measuring, assessing, weighting, weighting out, leveling, calibrating, sizing, grading according to size and to keep things in balance; arbitrate, adjudicate and work out or decide which is right; act appropriately and proportionately to sth, looks for appropriately balanced solutions; when things fit each other and are appropriate to sth; measured and balanced judgments; expect balance, reciprocity and mutual concessions

Intelligence can support and reinforce or suppress and temper HPT actions and reactions and lets a human to much more efficiently and easily fulfill needs, avoid increase of their intensities or decrease of their fulfillment. Intelligence can also

Table 2 The dislikes for the HPTs

DOM	be dominated, managed, determined, restrained, controlled, forced, ordered, steered, manipulated, persuaded, admonished, advised, instructed, when somebody decides or chooses something on behalf of him, when somebody criticizes his opinion, choices, decisions, things, thought, ideas and proposals; admit to defeats, failures, mistakes, lost, inability, incapacity, powerlessness, helplessness and everything else what debases or demeans him before others
MAX	minor, lesser, small or insignificant things, goals, aims, targets, actions, features, parameters, experiences or feelings; when somebody discourages him or tries to persuade him that his goals are impossible to achieve
INS	simple, normal, usual and ordinary things, features, actions; established procedures, actions, behaviours and schedules and to act according to them; repetition of uninspiring actions; stereotypes, stiff regulations, plans, procedures, sequences
DIS	simple, normal, usual and ordinary things, actions; unsolved or unexplained questions; repetitions and established procedures, actions, behaviours and schedules if he cannot get known something new
VER	inconsistence, mistakes, errors, bugs, lacks, a negligence, oversights, shortcomings, faults, flaws, vices, lies, contradictions, conflicts, misreadings, miscalculations, misconceptions, carelessness, imprecise or inaccurate things and actions; underestimation or underration of his notices and remarks
SYS	act without planning and scheduling things and actions; act in a hurry because he needs time for planning and scheduling; when somebody is late or unorganized
ASS	ignoring or disregarding his warnings, rules, steps and means of caution, safety and security or when people do not want to see or think about threats
HAR	conflicts, quarrels with sb, disagreements among sb, rows, brawls, fights, wars, stress, stress out, nervous situations, be under pressure from sb
EMP	when somebody does not tell about his intentions and reasons of doing something to him; when somebody escapes personal contact; when somebody cut his personal reflections
TAO	handle him with kid glove, pull wool over his eyes, beat about the bush, mince one's words; unfinished tasks, examinations or work; digressions, talking about personal reasons and intentions
BAL	unbalanced things or judgments; inconsistent acting or when something is inconsistent with something else; when somebody disregards, violates or breaks the law or when somebody neither observe nor follow rules and regulations nor does something contrary to the regulations

make a human to behave independently of personality and physiology. Consciously modified intelligent behaviour becomes automatic after some time. Each need can be fulfilled in some level so a person usually opposes to lose an achieved level and tries to achieve a higher level of its fulfillment (Fig. 2). Each HPT triggers a will to behave as specified in Table 1. Each HPT does not like, refuses or escapes things or actions described in Table 2. The HPTs can be recognized using words, phrases, an inflection and sentence construction described in Tables 3–4 and should be treated in the way described in Table 5 to trigger positive personality reactions. This paper focuses on actions and reactions coming from personality and on its linguistic expressions and constructions. These paper introduces 11 Human Personality Types (HPTs):

- *Dominant* (DOM) – He likes to dominate, choose and act after his own opinion. The ways of linguistic expressions of domination are partially dependent on an individual intelligence. More intelligent individuals will try to encourage or ask somebody to do something etc. Less intelligent individuals will force, order or

Table 3 The words and phrases which enable to recognize the HPTs

DOM	I, my, we, us, our, want, need, decide, choose, select, elect, control, steer, drive, conduct, program, order, book, recommend, independent, original, intact, untouched, virgin, have sth, own sth, owner, possess sth; [on] my own, unaided; I/we [would] like, my/our opinion, my/our choice, from my/our point of view, to my mind, I/we think, I/we wish; you right, I/we have decided to/on, not to be dependent on sb/sth, believe me, do what I say, I/we recommend, I/we agree with, I/we disagree with/on/about, I/we concur with
MAX	great, large, huge, extreme, super, hiper, enormous, maximal, fast, rapid, speedy, quick, nice, wonderful, exceptional, remarkable, unique, rare, better, the best, more, the most, higher, the highest, big, bigger, the biggest, an extreme size, weight or height of physical or intelectual features or parameters, no problem; I/we manage to do sth, I/we cope with sth; I/we get by/along
INS	idea, surprise, unexpected, unknown, new, revolution, inspiring, inspirational, inspired, remarkable, fantastic, mysterious, magic, super, muse, mood, art, artistic, imaginatively, create, creative, fashion; I have thought about, I have invented/devised/concoct, I have been inspired/enthralled by something, I have been surprised
DIS	why, ask, explanation, discover, reveal, discern, check, check-up, compare, recognize, reconnoitre, examine, penetrate, integrate, understand, new, news, unknown, unusual, unordinary, relation, explore, classify, associate, think over/about, ponder over, make out
VER	not, mis-, in-, im-, dis-, -less, disagree, incorrect, mistake, fault, error, inaccurate, imprecise, inconsistent, misplaced, careless, omitted, dirty, spoiled, risk, show, look at, attention, note, control, quality, details, detailed, precise, inexact, improve, correct, repair, mend, fix, remedy, rectify, redress, neatness, remark, notice, watch, observe, point out, thorough, painstaking, meticulous, punctilious; to be meticulous in sth/in doing sth; put the emphasis on sth
SYS	first, second, third, fourth, fifth, next, last, at last, now, earlier, later, at the beginning, start, start with, at the end, mess, tidiness, untidy, gradually, step, in steps, one by one, one after the other, in order, order, sort, sequence, rank, systematize, level, stage, arrange, classify, cluster, chronology, enumerate, list, map, schedule, appointment, diary, timetable, layout, compose, composition, group, structure, model, organization, organize, think, lay out, plan out, unfold, divide, spread out, time, on time, date, deadline, count
ASS	but, problem, doubt, misgivings, careful, be careful, danger, cautious, reliable, sure, confident, certain, secure, warrant, sure, protect, provisions, stores, stock, prevent, precaution, be assured, guaranteed, warranted, insured, prudent, risk, threat, safeguard, precautionary measure against something, alarm, alert, warn, limit, precaution against risks, just in case, can be necessary, put away something for a rainy day
HAR	O.K., yes, good, no problem, agree, confirm, bit, little, a bit, a little, not so much, not many, small, slight, minor, almost, let off, peace, quiet, it is interesting, I ponder over this, I think about this; I ask my; do you mind if I; excuse me
EMP	nice, I am, children, family, hurt, wound, distress, unpleasantness, tribulation, understand, tell me about your problem/situation, how about you, for you, with you, intend, intention, with a view to doing something, with the purpose of doing something, that is why, because, why, as, since, for, in order to, mean, what for, sympathy, sympathize with somebody, compassion, compassionate, condolence, pity somebody, regret, help, reciprocate, feel sorry for somebody; I am going to do something, what are you going to do, how are you?
TAO	performance, efficient, fast, concrete facts, specifics, hurry, hurry up, hasten, quickly, subject, task, practical, sensible, businesslike, precisely, accurately, exactly, finished, completed, concise, succinct, matter-of-fact, heart of the matter, hit the nail of the head, get on somebody with one's work, let's get down to facts/business/specifics, get down to brass tacks, talk sense, bring something to a conclusion; to get to the bottom of a matter
BAL	balance, balancing, counterbalance, compensation for sth, exact, accurate, precise, measure, measuring, assess, assessing, weight, weighting, weight out, level, calibrate, sizing, sort, grade, according to size, to keep things in balance, compare, compensate for, equal, be consistent, just, justice, fair, fairness, unfair, unfairly, judge, principle, rule, reciprocate, repay, be consistent, compensate, settle a debt, equalize, make something up to somebody

make somebody to do something etc. If a DOM individual is appropriately asked, there is a bigger probability to achieve a positive response or reaction when e.g. conducting negotiation. There is necessary a lot of intelligence, will and

Table 4 The inflection, sentence constructions and topics which enable to recognize the HPTs

DOM	recommendation, commands and orders; speaking on behalf of a group of people; expressions of selections, choices and decisions; giving own opinions and points of view; various expressions of a refuse, a perverse, contrary, revolt, rebel or mutiny
MAX	asking for something more extreme (bigger, taller, faster etc.); describing extreme goals, aims, targets or dreams; using adjectives and adverbs in comparative and superlative forms
ISN	talking about invention, projects, conceptions, style, fashion, inspiration, surprise
DIS	lot's of various questions, asks and requests for explanations of something he would like to know or understand, asks for options and points of view, I would like to know/understand
VER	point out various differences, impreciseness, faults, mistakes, inaccurateness, carelessness
SYS	use various expressions of time and date; enumeration or list something; show temporal or spatial relations and order
ASS	raise doubts, something but something; suspicious and mistrustful tasks, questions and statements
HAR	white lies, lies, telling to seek advice from somebody, diminutives and weaken words; handle somebody with kid glove; pull wool over somebody's eyes; beat about the bush; minces his words, excusing oneself, asking for permission
EMP	white lies to avoid unpleasant situation, diminutives, telling about intentions, reciprocate, weaken words, digressions, intentions, reasons and motivations behind behaviour and decisions
TAO	rare or no intention and a little explanation, short matter-of-fact speech, talk without beating about the bush
BAL	balancing, comparison, weighting, measuring, calibrating

understanding of other people personality to eliminate conflicts between DOM individuals. When a DOM individual is incorrectly treated he usually refuses, cancels, declines, denies, opposes or goes against somebody's orders or is perverse, contrary or revolt against somebody or something. He can come in conflicts with DOM and VER individuals. He prefers cooperation with HAR, EMP, weakly DOM, weakly VER individuals.

- *Maximalist* (MAX) – He focuses and looks for extreme goals, aims, targets, actions, features, parameters of things, experiences or feelings and he aspires to them. Intelligence can facilitate him to achieve his goals. Moreover, this HPT reinforces and heightens all other HPTs. When a MAX individual is simultaneously DOM he likes to challenge somebody and compete for something extreme in order to come into a possession of it. He can come in conflicts with ASS and VER individuals. He complements well with MAX and INS individuals.

- *Inspiring* (INS) – He is rarely consistent because he can suddenly change plans under the influence of a new idea, spur, impression or inspiration. He is usually very creative. He sometimes prefers even disarray, disorder or an artistic chaos to a well-organized orders, systems or arrangements because they can inspire him somehow better. Intelligence can greatly support and reinforce him to create new things and actions or to hit on new ideas and conceptions. He likes to be accompanied by INS, DIS and MAX individuals. He can come in conflicts with SYS, ASS, VER and BAL individuals.

Table 5 The profitable and beneficial treatment of the HPTs that value them

DOM	kindly or obsequious ask him for something (things, actions or opinions), thank him; leave him alternatives, possibilities, a choice, if possible carry out his selection and a choice; do not take him away consciousness of independence; neither make, push, force, drive, persuade, recommend, order, command him nor decide on behalf of him, neither narrow nor limit a selection or alternatives too much; neither admonish nor instruct him; sensitively advise or offer him solutions, products, propositions if he asks for it but leave him to make a final choice or make a final decision; do not show disrespect to his opinion or point of view; neither ignore nor disregard him
MAX	neither discourage him nor try to persuade him that his goals are impossible to achieve, try to show a will to help him to achieve his goals and help him to eliminate dangers, risks and threats on his way; express, reflect and illustrate extremes and splendid goals; demonstrate or offer ideas, ways, actions or things that can make him closer to these goals
INS	talk about inspiring, unknown, mysterious, fantastic things, actions, feelings, behaviours and people; surprise him and do some unexpected things or actions; do not plan nor schedule with him
DIS	talk about discoveries, news and other new or interesting things, ideas, associations, opinions, facts, relations, actions, reactions, behaviours, topics; conduct conversations and discussions on discovering topics; bring something undiscovered up for discussion
VER	listen to his remarks and comments, neither quarrel nor argue over what he said but you can tell him your opinion and substantiate it if you disagree with him; do not underestimate or underrate his notices and treat them as criticism or reject his notices; if possible and profitable use these notices to improve things or actions; weigh your words, choose your words carefully; let him to reveal mistakes, errors, bugs, lacks, a negligence, oversights, shortcomings, inconsistencies, faults, flaws, vices, lies, contradictions, conflicts, misreadings, miscalculations, misconceptions for free, thank him for them and ask for more and more notices, remarks, suggestions and comments until he finishes; let him be sure that you value his remarks and notices
SYS	map, reproduce and copy his order; enumerate or list things and actions; keep things in order and chronology; avoid doing digressions; do not change his plans too often; give him extra time for planning and scheduling before he starts to act or answer; do not hurry him
ASS	assure him that his words of caution and warning and against what he says are taken into consideration; ask for more notices and his recommendation; neither ignore nor shrug off his criticism or warnings; tell him what to do if/when in doubt or in case of doubt; all his doubts should be dispelled
HAR	do not come into conflict; make him sure he can tell what he means and his opinion, ask or suggestions do not cause a conflict; neither push nor drive him if you would like to know his opinion; do everything in peace, quiet and understanding; ask him about his opinion and show him that his answers will not lead to conflicts, quarels etc.; be sensitive to his opinions; help him to harmonize all situations and people; take into consideration that he probably can change his decision in case of somebody's disagreement or in view of having a quarrel with somebody
EMP	tell him about own intentions, reasons, be empathic and let him to understand your intentions, make personal digressions, do not cut his personal reflections, do not come straight to the point
TAO	talk to him in concrete terms, do not mince words; do not beat about the bush; immediately go straight down to business, facts, specifics or brass tacks; get immediately down to facts, business or specifics
BAL	keep balance in everything and be fair, just and self-critical

- *Discovering* (DIS) – He is very inquisitive, inquiring, incisive, very prying, curious about explanations of all mysteries, secrets, incomprehensible, obscure and undiscovered things, features and properties of things, actions, behaviours and relations. Intelligence can help him a lot to explore, penetrate, discern, checkup, recognize, reconnoitre, examine and finally associate facts with other facts and understand them. He is usually very interested in scientific and other

discoveries and keen on them. He likes to be accompanied especially by DIS and INS individuals.

- *Verifying* (VER) – He is precise, exact and consistent. He has a great talent for checking, verifying and noticing some important details that other people cannot notice or do not want to notice. Intelligence can also help him to recognize, associate, differentiate and compare more complicated and intricate mechanisms, relationships, things, actions, behaviours etc. He compares when associating facts with other facts. He can reveal many mistakes, errors, bugs, lacks, a negligence, oversights, shortcomings, inconsistencies, faults, flaws, vices, lies, contradictions, conflicts, misreadings, miscalculations, misconceptions etc. He likes when other people appreciate and value his notices and remarks and thank him for them. It is neither necessary nor obligatory to agree with everything he says. If somebody underestimates or underrates his notices, does not take them into account or rejects his notices he usually moves his focus on this person and starts to check and verify this person and looks for his or her shortcomings and vices. It can transform into a personal criticism and a heated exchange. He is often confrontational and contentious. He can come into conflicts especially with DOM, MAX, INS, HAR, EMP individuals.
- *Systematic* (SYS) – He acts only after he has planned everything. He lives confined in his own plans, systems and schedules. When something changes he has to change his plans before he will continue. This is the reason why he does not like when somebody or something makes him to reschedule or change plans. He focuses on ordering, sorting, arranging, classifying and clustering everything around. He has a tendency to pigeonhole somebody as somebody else from his model after a standard classification of his or her behaviour. An order and a tidiness have a special value for him. Intelligence can help him to associate everything and create a very intricate and sophisticated ordered model of the world. He likes to be accompanied by SYS, ASS, VER and TAO individuals. He can come into conflicts with INS, MAX, DOM and TAO individuals.
- *Assurant* (ASS) – He always expects some troubles, problems and difficulties and talks about them. He pick holes in almost everything. He has many doubts and misgivings about various people, things, procedures etc. He can come into conflicts especially with INS, MAX, weakly SYS, weakly ASS and DOM individuals. He complements other ASS and VER individuals.
- *Harmonious* (HAR) – He can seemingly do what other people want and behave towards others like adapting, adjusting or conforming to them but later he usually does something else if there is no threat of further conflicts. He smiles a lot, reassures and behave as being compliant. A HAR individual rarely says what he means, he can even lie, suppress, withdraw, cancel, reverse, ease off, let off, give something a miss, skip something or hold something back in order to neither lead nor bring about conflict. He usually handles somebody with kid glove, pulls wool over somebody's eyes, beats about the bush and minces his words. He uses many soothing, softening and weaken words, diminutives in order to make sensitive or difficult situations more agreeable, accommodating, friendly, amicable, soft, gentle or harmonious. He intentionally does not come to conflicts with anybody.

He is especially exposed to DOM and DIF individuals who can create confronta-
tional and contentious situations which he tries to avoid. He prefers contacts with
HAR, EMP and BAL people.

- *Emphatic* (EMP) – He is usually very communicative and forthcoming especially
  about private life, things and experiences. He lets other people in on his intentions
  of doing something. In his talks, he puts in many private words and digressions.
  He uses many soothing, softening and weaken words in order to degrade a dis-
  tance to other people. He feels strange in company of TAO and weakly EMP
  individuals. He prefers a company of EMP individuals.
- *Task-Oriented* (TAO) – He passes information on to somebody rather short with-
  out telling about his intentions or private digressions. He neither fraternizes with
  somebody fast nor opens up to somebody nor confides in somebody rapidly. He
  often harries other people and gets on other people with their work. He harries
  others to get down to facts, business or specifics and asks about them. He does
  not like to talk to EMP and HAR individuals because they usually mince their
  words and speak not in concrete terms. He can go well with TAO, SYS and VER
  individuals because they are meticulous and concrete.
- *Balancing* (BAL) – He is very balanced, fair and consistent. He persuades others
  to apply and observe rules, regulations and principles that are balanced, just and
  fair. He makes things on principle. He is capable of being very exact in criticism
  and self-criticism. He can come well with BAL and VER individuals. Conflicts
  are possible with INS, MAX, DOM, HAR and weakly BAL individuals.

On account of a huge number of words and phrases in English the description in
Tables 3–4 cannot contain all words and phrases for each HPT. It describes only
the general and major features of each introduced HPT and presents the main idea
of their recognition and beneficial uses. Each person has got a different mixture of
HPTs and their intensities (Fig. 1). The most intensive HPTs have usually a major
role in giving a person satisfaction and fulfillment. Many times two groups of HPTs
pull a person in opposite directions in accordance with the intensities of their needs,
e.g., let us assume that a DOM (intensity 70%) and DIS (intensity 40%) individual
did not understand a statement. The cognitive part of his personality (with lower
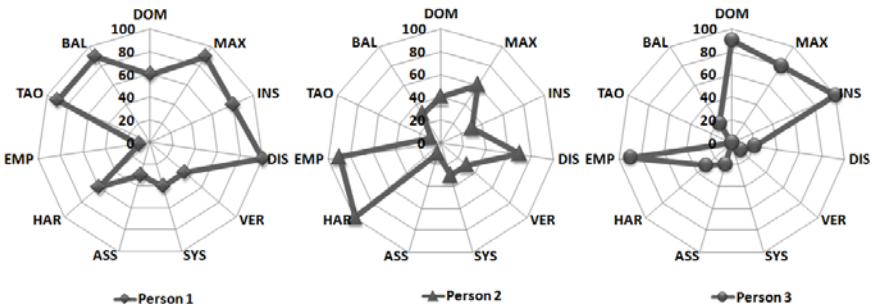
Fig. 1 The personality profiles taking into account intensities of the HPTs

intensity 40%) wants to ask for more details and an explanation, but the dominating part (with higher intensity 70%) of his personality does not want to demean himself before a speaker. This person is torn between asking something or not asking. Usually the stronger sum of intensities of HPTs decides about the final choice if intelligence does not temper this personality natural reaction. If these intensities are reverse, e.g., DOM (intensity 50%) and DIS (intensity 80%) individual will probably ask about more details. The intensities of HPTs can also sum up together, e.g., HAR intensity 60% + EMP intensity 40% can decide for doing something even if SYS intensity 80% (which is stronger than both of them individually) wants something opposite. Intelligence can trigger other reaction that does not result of the stronger sum of intensities of HPTs. Such intelligent reaction, which is incompatible to the needs of recognized HPTs, can be sensible but usually sensed as unpleasant, unnatural or against one's nature.

3 Psycholinguistic Recognition of Personality and Needs

Chatbot engine can automatically count up characteristic words, phrases, an inflection, sentence construction during a talk and weight them up properly for each HPT in order to fix intensities of them and choose a profitable actions or reactions that can trigger positive human reactions. The described personality determination and computations are passive and apart from consciousness of a talking person. Some experiments have been performed in construction of chatbots for internet shops. The contents and the way of presentation of information were adjusted to intensities of recognized HPTs. The testing part of customers that talks to a chatbot which recognized their personality and tuned its expressions and forms of a product presentation to them were about 30% more satisfied and want to bought something than the other testing part of customers that were treated in the same manner regardless of their personality. Even simple counting of the characteristic words and phrases (Table 3) for the defined HPTs used in a talk by a human was enough to achieve this result.

Intelligence can associate and conclude that some actions and behaviours are necessary to achieve some goals because it is reasonable about some actions and wise. Intelligence can also temper some unprofitable personality reactions. Intelligence can eliminate all misunderstanding and many conflicts between different HPTs. Intelligence can also substitute some desirable actions that are missing or weak in a certain personality, e.g.:

- NonSYS individual can learn to be sometimes systematic because he needs it in his work or to achieve certain goals that can come from certain needs.
- NonEMP individual can learn to understand other people and examine their needs and feelings because it helps him to talk and communicate to them or negotiate and cooperate with them.
- NonVER MAX individual can learn to pay his attention on some smaller things that can result in some problems that can prevent him from achieving some desirable extreme goals.

suspense, pressure, stress of decrease or not increase of satisfaction

fear, dread, anxiety, pain, suffer, tribulation, reluctance	hope, expectations, direction, intention, requirements, demands, aspirations
unpleasant experiences, dissatisfied, discontented, unfulfilled, distress	pleasant and nice experiences, satisfaction, fulfillment, happiness, pleasure, relieve onself
Decrease of need satisfaction and fulfillment	Increase of need satisfaction and fulfillment	Tendency to grow in time if satisfied
fulfillment of the need	intensity of the need	

0%   Level of current   100%
                      fulfillment of the need   Target level of the need fulfillment

Fig. 2 The directions and relations between fulfillment and intensity of a need, tendency to grow it in time and feelings accompanying them

People try to get and drive more satisfaction from something, take and try to find more pleasure in doing something (Table 1). On the other hand, people try to defend and secure a possessed fulfillment against decrease or lost (Table 2, Fig. 2). The needs make us living creatures that have reasons to live and fulfill needs. The needs affect on us to create and strive for some targets that fulfill them. The needs also affect a development and an increase of intelligence that supports and reinforces our beneficial and profitable actions and reactions through more sophisticated associating of facts, sequences of facts and groups of facts. Intelligence can give us more pleasure of doing something and enables us to express more satisfaction in various ways. The personality needs can be recognized and classified using the described psycholinguistic model of personality (Table 5) and its needs (Table 1). The needs can be used to qualify expected behaviours. The recognized HPTs can give us better understanding of each other and of what differs us and exclude many misunderstanding. The described groups of personal behaviours (HPTs) can be even used to foresee some individual future actions and reactions in a given context or situation what can result in better understanding of each other, in better possibility to conduct negotiations and in easier cooperation. These knowledge can also help to model and understand human needs by a computer and create better algorithms (Table 5) and interfaces for man-machine interactions.

4 Conclusions

The paper describes the new philosophy to construct intelligent interfaces for man-machine interactions using psychological knowledge of human personality. It has defined and introduced 11 human personality types (Dominating – DOM, Maximalist – MAX, Inspiring – INS, Discovering – DIS, Verifying – VER, Systematic – SYS, Assurant – ASS, Harmonious – HAR, Emphatic – EMP, Task-Oriented – TAO and Balancing – BAL) and shown the way of their recognition. It points out usual ways of behaviour that are characteristic for them. It also describes likes and dislikes for them that can conclude in formation of appropriate algorithms

for more profitable interaction between them and a chatbot. It describes how to tune to them and use more profitable actions and reactions to various persons individually (Table 5). The presented description supplies our consciousness with understanding of needs coming from personality (Table 1) and can influence the creation of more human interfaces that are able to understand and go along with human natural expectations.

References

1. Allen, B.P.: Personality Theories: Development, Growth, and Diversity, 5th edn. Allyn and Bacon, Needham Heights (2006)
2. Cloninger, S.C.: Theories of Personality: Understanding Persons, 4th edn. Prentice Hall, New York (2004)
3. Engler, B.: Personality Theories. Houghton Mifflin, Boston (2006)
4. Engler, B.: Personality Theories: An Introduction, 7th edn. Houghton Mifflin, Boston (2006)
5. Ewen, R.B.: An Introduction to Theories of Personality, 6th edn. Lawrence Erlbaum Associates, New York (2003)
6. Feist, J., Feist, G.J.: Theories of Personality, 6th edn. McGraw-Hill, New York (2006)
7. Friedman, H.S., Schustack, M.W.: Personality: Classic Theories and Modern Research, 3rd edn. Allyn and Bacon, Needham Heights (2006)
8. Gut, J., Haman, W.: Appreciate conflict, 3rd edn. OnePress, Helion (2008)
9. Hergenhahn, B.R., Olson, M.: An Introduction to Theories of Personality, 7th edn. Prentice Hall, New York (2007)
10. Horzyk, A.: Secrets of negotiation with satisfaction. Lectures at AGH University of Science and Technology (2006–2009)
11. Kalat, J.: Biological Psychology. Thomson Learning Incorporated, Wadsworth (2004)
12. Monte, C.F., Sollod, R.N.: Beneath the Mask: An Introduction to Theories of Personality, 7th edn. Harcourt College Publishers, Fort Worth (2003)
13. Ryckman, R.M.: Theories of Personality, 9th edn. Belmont, Cengage Learning (2008)
14. Schultz, D.P., Schultz, S.E.: Theories of Personality, 9th edn. Belmont, Cengage Learning (2009)

Adaptable Graphical User Interfaces
for Player-Based Applications

Łukasz Wyciślik

Abstract. The article presents the architecture concept of systems that have player-based presentation layer and due to collecting user activity statistics is adaptable. Thanks to separation of persistent data, domain logic, presentation logic and view layers it is possible to develop systems exposing multiple versions of GUI to achieve one domain goal. The GUI versioning mechanism combined with tracing of user activities gives opportunity to evolutionary changing of presentation logic and view layers to achieve maximum intuitive and friendly GUI. In the article further research direction is also presented based upon the concept of user profiles gathering. This mechanism enables exposing different versions of GUI to different users groups and is applicable also to disabled users with different types of disabilities.

Keywords: GUI, adaptable interface, user profile, domain logic.

1 Introduction

The concept of graphical user interface (GUI) was introduced in the 1960s by Dr. Douglas Engelbart and is commonly understood as the use of graphic icons and pointing devices to control a computer. Evolution of graphical operating systems (e.g., GEOS, TOS, and Windows) has brought common understanding of GUI controls such as window, menu, check box etc. that gave basics for development of GUI for applications presently known as 'rich application' (or 'fat application'). In the 1990s, due to growing Internet popularity and internet browsers, the World Wide Web capability limited to static information sharing (as WWW pages) appeared insufficient. The need to make also services accessible through Internet brought

Łukasz Wyciślik
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: lwycislik@polsl.pl

the rapid development of internet applications technology presently known as 'thin client' (or 'slim client'). Nowadays, due to continued growing of Internet accessibility and operating systems diversity, there are several key challenges in application development: independence from operating systems and other system components (web browsers, runtime frameworks, etc.) on which the application is running, need of intuitive and friendly GUIs often considered as rich GUI, need of reliability, security, efficiency and many other [10]. Both of technologies have serious lacks to adequate fulfill mentioned above postulates. 'Rich application' technology depends on operating system often even when the virtual machine technology is involved (JAVA VM, .NET CLR) while 'thin client' gives limited ability in rendering 'rich-like GUI' even when DHTML and java-script technology is involved (what's more, the use of DHTML and java-script technology results in limited independence of web browsers). Disadvantages and limitations of technologies mentioned above resulted in completely new approach to GUI building technology that is based on dedicated 'player' component. This component is a specialized piece of software that is responsible for rich GUI rendering and runs on client-side node (within web browser and/or operating system). Due to its specialization the GUI rendering mechanism offers features available in modern graphical operating systems not limited just to simple 'window displaying/managing' but also supports animation effects, vector graphics, 3-D graphics, multimedia streams etc. What's more it easy enables to build custom GUI controls that cross the boundary of graphical operating system standards. It is also 'lightweight' enough to install or upgrade without user interaction (contrary to 'heavy' multi-purpose virtual machines such as JAVA VM or .NET).

The long evolution of input-output computer interfaces to its present form was driven mostly (beside aesthetic aspects) by maximum ergonomy criterion. Ergonomy requirements may be different dependly on user abilities, e.g., system that is used occasionally by concrete user but is dedicated for large population (e.g., photo portal) should be intuitive and self-describing while system that is used intensively but by a limited number of users (e.g., banking system operated by a clerck side) should have compact view and be equipped with 'shortcut keys'. For disabled people ergonomy requirements vary dependly on their disability types, e.g., for color-blinds the proper GUI colouring needs to be provided while for blind users specialized voice supporting systems should be involved. But regardless of methods used to fulfill the ergonomy criterion there is one main goal to be achieved – to enable users achieving their goals efficiently what can be measured directly by taking proper time statistics.

There are two opposite approaches to GUI designing – a 'white box' one, which assuming that GUI designer is an expert of future user population abilities and business processes the system is dedicated for. In such case the GUI model is developed according to an expert knowledge. It's obviously difficult to apply this method on systems dedicated for a large population of people derived from different environments (cultural, profession etc.). The second approach is an evolutionary one and relays on a 'black-box' conception – that it does not assume knowledge about

future system users, but thanks to fact that the GUI ergonomy could by measured it assumes iteration process of GUI developing. This approach enables even completely automatic process of GUI developing – e.g., the GUI model can be continuously being modified by evolutionary algorithms. Of course these two approaches can be combined, e.g., the GUI designer develops the first or even successive GUI models.

2   User Goal Driven Approach for GUI Building

Present-day methods of building complex business applications assume performing several activities that may be grouped in categories such as analyzing users needs, designing, building, testing, deploying, etc. [5] The first step to build such system is to understand the business the system is dedicated for and to understand the needs of future users. As object oriented analysis is the commonly used method the several UML diagrams are created to describe the business domain and user needs. These diagrams form analytical model of the future system. Such model should describe system in structural perspective by actor, class, object diagrams and in behavioral perspective by sequence, colaboration, activity and use case diagrams [4]. Use case diagrams describe relations among use cases.

Use case defined on human-computer interaction level is the flow of interactions between user and system after that one user goal may be achieved. Use cases are often base for functional requirements specification and are part of an analytical model. They have the most influence on GUI designing and later GUI development but unfortunately they are not a part of UML 2.0 specification what results in various approaches to use case specifying [7, 8, 9]. The detailed description of human-computer interactions gives the use case scenario which consists of several steps. One use case may be described by several use case scenarios dependent on use case complexity – the more decision branches – the more use case scenarios. The use case realization may involve several use case scenarios in various order – dependly on starting conditions and later circumstances. Changes in business reality may cause to change system requirements and results in new versions of use cases. Use case version may be implemented in various implementations – it depends on displaying and inputting devices and dependent on different user preferences. One use case must have one main actor so concrete flow of a use case must have one performer (user) similarly one use case involves one object type (that is term from business model) so concrete flow of a use case involves one object (that is instance of term). Use cases may be reusable by other use cases by include or extend relations so concrete flow may also include or be extended by other concrete flows. All of above concepts are presented in the class diagram in Fig. 1.

The concrete flow consists of atomic steps that describe performer or system action. The concrete flow may be finished positively, negatively or be interrupted (Result attribute). Steps in concrete flow are identified by successive numbers and are related to use case scenarios steps by value of scenario step id attribute.
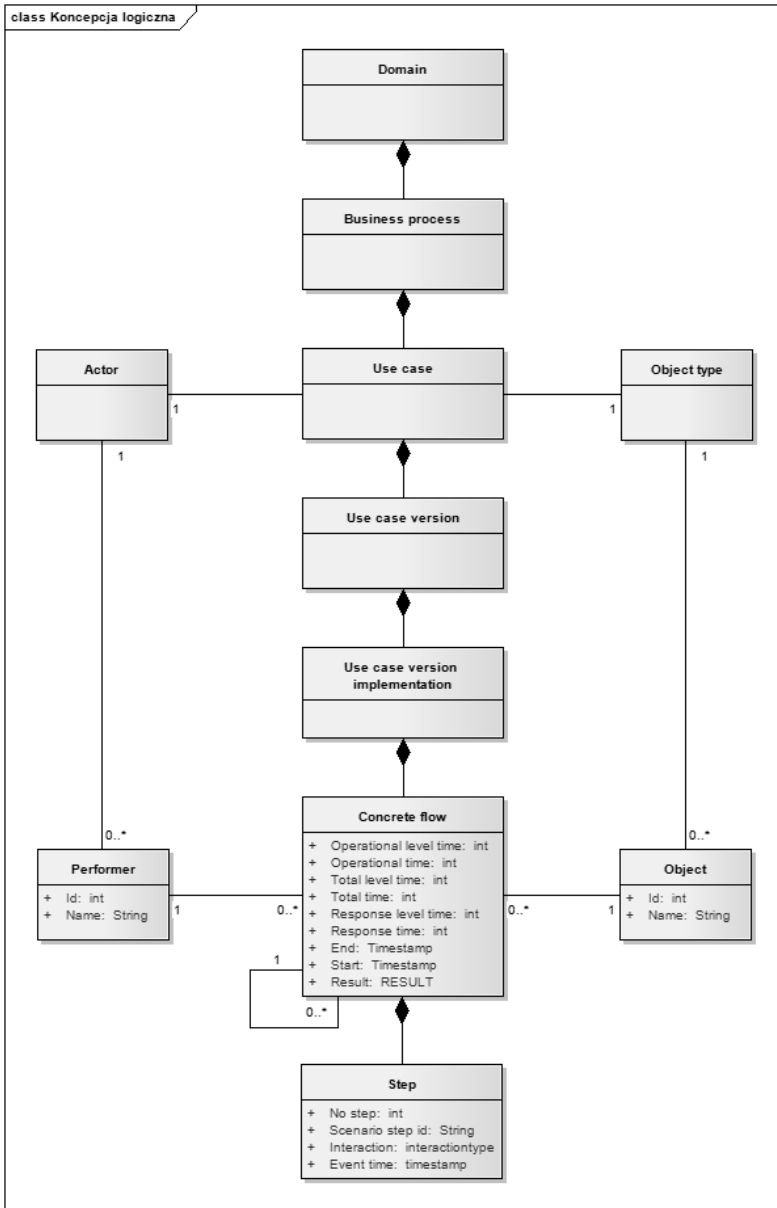
Fig. 1 Model of tracing between analitycal perspective and GUI implementation

Gathering time statistics in each step lets calculating the system response time in case of steps describing system actions (what is the measure of system performance) and user action time that is needed for navigating the GUI (what is the measure of

GUI ergonomy) or for human-human interaction in case of the secondary actor existence in the use case (what is the total time of navigating the GUI and human-human dialog). Because of the above there is need to distinguish this three step types (interaction type attribute in step class). At the concrete flow class level the are six time ratings – the total time, that document the total time of the use case duration including all sub-use cases, the total level time, that documents the total time of the use case duration excluding all sub-use cases, the response time, that documents the time of the use case tie-up including all sub-use cases, the response level time, that documents the time of the use case tie-up excluding all sub-use cases, the operational time that documents the time of navigating the GUI including all sub-use cases and the operational level time that documents the time of navigating the GUI excluding all sub-use cases. All of these ratings are derived from timestamps of steps. It is worth mentioning that for adaptable graphical user interfaces building the operational ratings may be analyzed separatly for each performer and for system performance tuning the response ratings analyzing for each object separately should be considered.

3 Architectural Concept of a System with Player-Based Presentation Layer

The architecture concept of adaptable graphical user interfaces for player-based applications concerns view and presentation logic layer and logging/auditing aspect. Other system layers are not affected. The concept and advantages comming out of system layering were described, e.g., in [3]. On a client side there are view and presentation logic layers and a proxy [11] component for a data logging aspect. They are running on an Adobe player (Flash for webbrowsers, AIR for desktop applications) and are commonly known as Flex [2] applications. On a server side there are BlazeDS [1] and Logback [6] components. Basic relations between these pieces of architecture are shown in deployment diagram in Fig. 2.

The communication between clients and server is done with the aid of AMF3 binary network protocol implemented in BlazeDs component. Thanks to it exposing java methods to Flex is only the matter of proper system configuration at deployment time. This communication chanel is used by proposed architecture solution, but may be used also for regular business method calls as well. On the server side the general logging mechanism Logback is used but for simplifying and adapting its interface the dedicated audit facade is implemented. This facade is accessible directly from the client side Flex technology.

Gathering use case flow data is based on audit facade methods calling from the presentation logic layer of Flex application. These calls are done by proxy class that passing them directly to the server or caching them and sending to the server less frequently but in bigger packets what provides better system efficiency. Fig. 3 presents the class diagram of proposed solution.
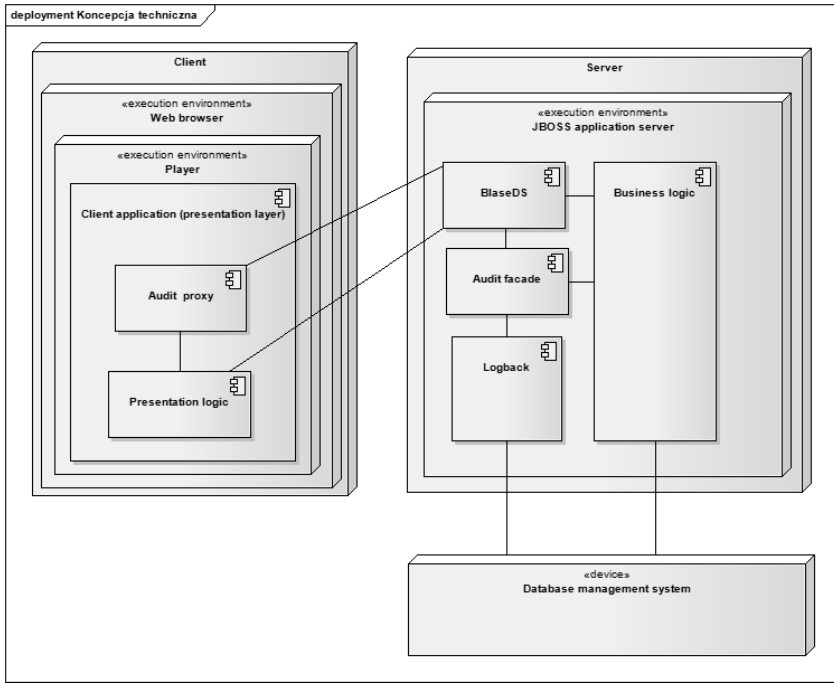
Fig. 2 Deployment diagram of adaptable graphical user interfaces for player-based applications infrastructure
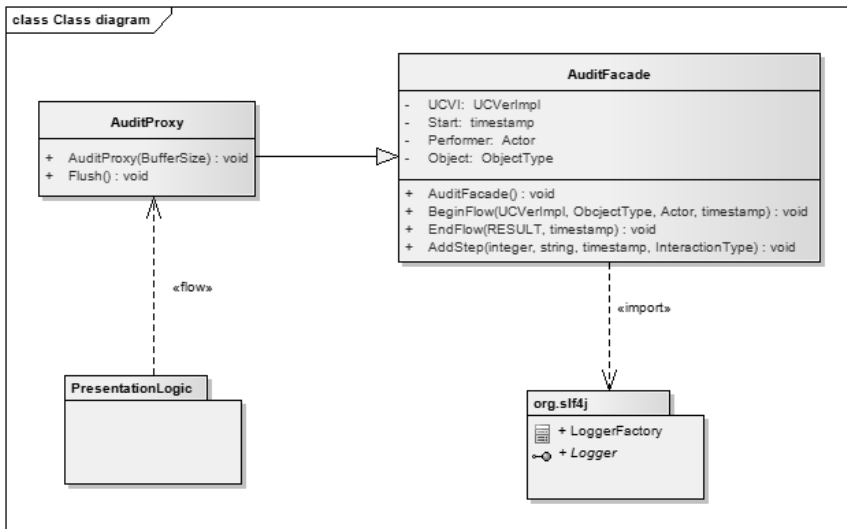
Fig. 3 Class diagram of adaptable graphical user interfaces for player-based applications infrastructure

4   Conclusions

The article presents the architecture concept of systems that have player-based presentation layer and due to collecting user activity statistics and supporting GUI versioning is adaptable. Thanks to separation of persistent data, domain logic, presentation logic and view layers it is possible to develop systems exposing multiple versions of GUI to achieve the same goal. The GUI versioning mechanism combined with tracing of user activities gives opportunity to evolutionary changing of presentation logic and view layers to achieve maximum intuitive and friendly GUI.

Of course gathering reliable data requires building 'user-goal centric' contrary to 'persisten-data-model centric' GUIs, which means:

- for each use-case implementation it must be known when and what use case is started,
- for each use-case implementation it must be known when and what use case is finished,
- user has no possibility to branch use case accept the branches defined in use case model by extends and include stereotypes,
- every step of scenario has defined the interaction type,
- human-human interaction step should not require GUI navigation if possible.

Proposed architecture helps to discover inefficiency/performance bottlenecks not only derived from a GUI implementation defects but also resulted from lower system layers implementation defects (e.g., business layer, object-relational mapping layer, persistent data layer). The future research will concern the concept of clustering user statistics data in order to determine the minimal number of parallel GUI versions to be served to achieve the same user goal. This mechanism would enable exposing different versions of GUI to different users groups and will be applicable also to disabled users with different types of disabilities.

References

1. Adobe: Blazeds page, http://opensource.adobe.com/wiki/display/blazeds/BlazeDS/
2. Adobe: Flex3 page, http://www.adobe.com/products/flex/
3. Fowler, M., Rice, D.: Patterns of Enterprise Application Architecture. Addison-Wesley, Reading (2003)
4. Jacobson, I., Christerson, M., Jonsson, P., Overgaard, G.: Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, Reading (1992)
5. Larman, C., Basili, V.R.: Iterative and incremental development: A brief history. Computer 36(6), 47–56 (2003)
6. Logback: Home page, http://logback.qos.ch/
7. Metz, P.: Merging the use case textual and graphical use case worlds. Ph.D. thesis, Department of Mathematics and Computing, Cork Institute of Technology (2003)
8. Metz, P., O'Brien, J., Weber, W.: Specifying use case interaction: Types of alternative courses. Journal of Object Technology 2(2), 111–131 (2003)

9. Metz, P., O'Brien, J., Weber, W.: Specifying use case interaction: Claryfying extension points and rejoin points. Journal of Object Technology 3(5), 87–102 (2004)
10. Myers, B.A.: Why are human-computer interfaces difficult to design and implement? Cornegie Mellon University, Computer Science Department, CMU-CS-93-183 (1993)
11. Tichy, W.F.: A catalogue of general-purpose design patterns. In: Proceedings of Technology of Object-Oriented Languages and Systems. TOOLS, vol. 23. IEEE Computer Society, Los Alamitos (1998)

Case-Based Reasoning Model in Process of Emergency Management

Jiri Krupka, Miloslava Kasparova, and Pavel Jirava

Abstract. In this paper a modelling of a system, in the conventional structure, of the emergency service is presented. The model is based on the case-based reasoning (CBR) algorithm, and it focuses on the fire rescue service. The algorithm has been realized in MATLAB, and their outputs represent effective recommendations for a commander of a fire protection unit. In the first part of this paper, information about CBR algorithm, and a decision process in the tactical level of fire rescue service, are briefly introduced. The second part is aimed at the system approach definition of the formation of 'cases' of decision making, by the fire protection unit commander in charge during the handling of emergency situations, on the basis of guidelines in the fire protection unit field manual. The manual contains a set of methodical sheets; every sheet is described by a set of attributes. It exploits characterization of a set of cases. Generally this real decision is executed under pressured time, contains a lot of information, and it is unique for each situation; it is very important to compare the decision of new cases with the base decisions of previous cases. In the third part, the possibility of applying CBR algorithm to new real cases is analysed. We modified the guidelines of cases on the basis of commander's know how, and on expert recommendations. We analyzed some metrices into a comparison of cases, and we realized CBR algorithm as a graphic user interface. Finally we discussed the possibility of using Soft CBR, it means using theories of fuzzy logic and rough sets for a description of vagueness and uncertainty during the description of knowledge in the fire rescue service. We will consider an implementation of CBR algorithm on digital signal processors.

Keywords: fire rescue service, knowledge, cases, case-based reasoning, uncertainty, soft case-based reasoning, fuzzy logic, rough sets theory.

Jiri Krupka · Miloslava Kasparova · Pavel Jirava
Institute of System Engineering and Informatics, Faculty of Economics and Administration, University of Pardubice,
Studentska 84, 532 10 Pardubice, Czech Republic
e-mail:{Jiri.Krupka,Miloslava.Kasparova,Pavel.Jirava}@upce.cz

1 Introduction

Concepts such as 'Emergency management' and 'Emergency service' are very broad and their understanding is different worldwide (e.g. we can mention the US conception [19], an Australian conception [4], or a French conception [1]). In the Czech Republic (CR) it is referred to as the Integrated Rescue System (IRS), which consists of the main units and other units. One of the main units of IRS [7] is the Fire Rescue Service (FRS), which is one of the basic bodies of the IRS. The primary mission of the FRS [5], [6] is to protect life, health, and property of citizens against fire, and to provide effective help in emergencies. The FRS participates in solving crisis situations (they are defined in the legislation). The crisis state is defined mainly by Constitutional Law No. 110/1998 and Law No. 239/2000 in IRS [7]. The crisis management (CM) deals with the problems in handling crisis situations as they happen, and in theoretical situations. The CM [8] represents a logically neat set of information about crisis, their causes and consequences, as well as their principles, methods, and procedures in solving them. The CM is an interdisciplinary science which pursues management as a purposeful activity of people, and its role is to create a methodology of management emphasized on achieving efficiency related to the set aim (i.e., to protect human society and material values against the effects of a crisis [8]). At the same time it is a kind of activity, or structure of activities, which is used by the managers to achieve the aim, including a decision making process.

If we pursue the decision making process mentioned above [2, 10, 13, 18, 21] then we will not only cope with a method 'Common Sense', when a human banks on his or her experience or opinions with solving everyday problems, or uses the principle Occam's razor (when the easiest solution is the best one), but we also have to use an artificial intelligence [16]. E.g., there is a creative modelling in which there are methods of natural conclusion and explanation (i.e., using computers for typical 'human' activities – theorem argumentation, understanding the natural texts, medical or technical diagnostics, or knowledge engineering [16]). The main topics of knowledge engineering are methods and techniques for: getting, formulating, coding, keeping, testing, and maintenance of knowledge. Knowledge is either taken from top specialists, experts, or it is intuitively deduced from the data file. One of the methods for obtaining information automatically is the CBR. The key field in the CBR is creating a 'classic case'.

The CBR [20] is based on an expert's decision made in an unknown field according to the similarities of previously solved cases, and not based on the system of principles. It is supposed that, what was done in one situation is also likely suitable for similar situations [11]. This procedure matches the Anglo-Saxon conception of the law based on precedents. The precedent is a factor, which could be defined as a previous decision used for future solutions of analogous cases. The knowledge is saved in the form of the previously solved situations. The result of this is a database where each written record has its own case, and individual entries are characteristics of the case (i.e., inquiries which the system asks during the counsel). When making a decision in a new situation, the database looks for the most similar situation. During the run of the algorithm the base of cases could be 'refilled', this means it can
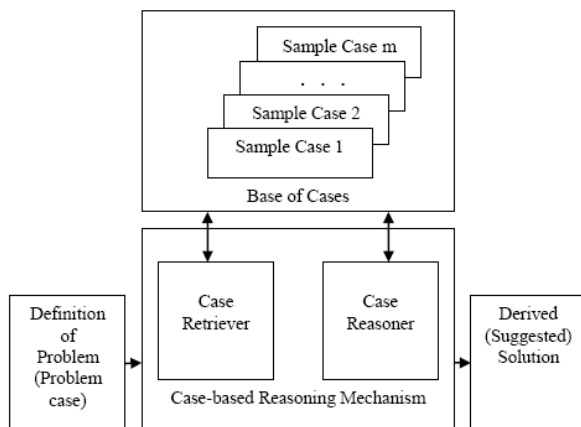
Fig. 1 Model of CBR System

save 'new' cases into the database. The CBR uses an effective index and searching algorithm and metrics [20], which can evaluate the similarities between the cases.

The stages of problem solving in CBR [3, 9, 14] could be seen in the model R4 [20], which are: Retrieve, Reuse, Revise, and Retain the cases. The CBR mechanism could be defined as a 'black box' in the model system of CBR (seen in Fig. 1), where the input procedure is 'Case Retrieve' and the output procedure is 'Case Reason'.

2 Model of CBR for Fire Protection Unit

According to the discussions at the Department of Crisis in the County Council of Pardubice, and the Department of Prevention and Planning of the Fire Brigade in Pardubice's Region, an orientation of the practical usage of the CBR method was chosen, it is aimed at the tactical level of the Fire Brigade, this means the actions taken for each action by commanders of fire protection units (FPU) [12]. The commander of the action is physically at the location where the crisis situation is, and makes many decisions, usually under a time constraint and under the stress brought by the information acquired from the neighbourhood. The FPU Field Manual was developed in order to help and make decisions easier, which involves methodical sheets[1] (MSs).

MSs from the field manual are those which describe each crisis situation, and have a role to help the commander solve the situations, which were used for the compilation of the 'sample' cases. The MS contains: characteristics of each situation, presumptive occurrence of the situation (or in what circumstances this situation

---

[1] It is available at URL: http://www.hzscr.cz/clanek/bojovy-rad-jednotek-pozarni-ochrany-v-dokumentech-491249.aspx. (in Czech)

might occur), tasks and procedures of the FPU activities, and prospective unusualness of the situations.

There were nine MSs chosen from the FPU Field Manual for a practical example. They have the same attribute, 'general threat'. The chosen MSs are: accidents on the road (D1), accidents with a huge amount of casualties (D3), the action with the occurrence of dangerous substances (L1), the breakdown of menacing water supplies and oil supplies (L3), the action with the occurrence of explosives and explosive materials and their impetuses (L12), the danger of corroding (N3), the danger of intoxication (N8), the danger on roads (N10), and the danger of explosions (N16).

The main part of each MS was the incipient situation for a choice of attributes, were the characteristics of the cases are described. The other criterion was the evidence of attributes at the first probe of the location where the accident happened, made by the commander of the action.

Seventeen attributes were forecasted, but later on they were reduced to twelve a1, a2, ..., a12 (after the discussion with a commander of the Chemical Department of the Fire Brigade). The attributes are: the amount of crashed vehicles (a1), the quality of transported cargo (a2), meteorological conditions (a3), propulsive and operational liquid (a4), marking of the vehicles (a5), an unusual flame (a6), special coverings (a7), vegetation changes in the surrounding area (a8), menacing of water supplies (a9), the danger of explosions and chemical reactions (a10), an emergence of warmth, steam and acoustical anomaly while burning (a11), and health problems of the people involved (a12).

Three steps were used when assigning the values of the MS attributes on the basis of Table 1. The first step was the main meaning of each MS, and the table of attributes was made only according to the instructions listed in the MS. These symbols are marked 'A'. The second step was done in order to make a table of attributes, which was added to the original table of attributes. The added information was the knowledge which was gained through a discussion with expert 1, the commander of Pardubice's Fire Brigade Station. The symbols for these are marked 'B'. The third step in the table of attributes was added to the table as long-lasting experience of an expert 2, the commander of the Chemical Department of the Fire Brigade. The symbols of the added table are marked 'C'. Due to these steps, the final table of attributes of the MS was obtain. The lines are formed by each MS and the columns are formed by the values of attributes describing every situation contained in the chosen MS. Every symbol has a value from 1 to 3, where 3 is the big value. Empty cells in this table have values equal 0.

The input procedure 'Case Retrieve' among the MSs usually uses [11, 20] the nearest neighbour and inductive retrieval methods. One of them, takes advantage of Euclidean metrics, the Euclidean distance ρ is the ordinary distance between two points $X = (X_1, X_2, \ldots, X_n)$ and $Y = (Y_1, Y_2, \ldots, Y_n)$ in Euclidean n-space, on R^n, is defined as:

$$\rho(X,Y) = \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2 + \cdots + (X_n - Y_n)^2} \,, \qquad (1)$$

Table 1 Attribute Values of Cases

Type of MS	Attributes of Cases											
	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12
D1	A	A	A									
D3	A	A	A	A								
L1		C	C	C	A	A	A	A	C	A	C	C
L3			C	C	C		C	C	A			
L12	C	C			B	C	B	B		A	B	
N3		C			B		B	B	C	C	C	A
N8		C	C		B		B	B	C	C	A	A
N10	A	A		A	A							
N16		C			C	A	A			A	A	C

where $X, Y \in K$ and K is a set $K \neq \emptyset$, and ρ is a relation from $K \times K$ to R, and $\rho(X,Y) \geq 0$, and $\rho(X,Y) = 0 \Leftrightarrow X = Y$, and $\rho(X,Y) = \rho(Y,X)$.

3 Application of CBR Algorithm

The CBR algorithm for this application is defined as the algorithm with a feedback. Due to this feedback, the Base of our program's cases is refilled with new solutions and the system is updated. The new solutions are the cases which were solved in the past. The algorithm was suggested according to the general method CBR [19] and its pseudo code is in Fig. 2.

The algorithm begins with loading base of cases and output vectors. Base of cases is a matrix $(m \times n)$, where cases are in rows (i-case, $i = 1, 2, \ldots, m$) and attributes are in columns (j-attribute, $j = 1, 2, \ldots, n$). The size of the output vector is $(m \times 1)$, where every row contains a proposed solution from base of cases. Loading a new case is the next (second) step. The new case is composed of n-items vector, where n is the number of attributes. Loading could be realized in two ways, manually or by import from the file. This file must contain semicolon delimited matrix $(1 \times n)$. Correctness control of new case is the third step. This control is realized by checking the number of base rules attributes and new case attributes. When the number of attributes is not equal to 12, the program returns to the second step.

New cases are compared with cases in the base of cases for similarities in the fourth step. The nearest neighbour method is used in this step. It means that for new case X_{new} elements from the base of cases X_r are searched. For X_r the items distances is defined in the following way:

$$\| X_r - X_{\text{new}} \| = \min_{i=1,2,\ldots,m} \| X_i - X_{\text{new}} \|, \tag{2}$$

where $\| \bullet \|$ is Euclidean distance.

Input   : Base of cases X_r; Set of new cases X; Solution S,
        where X_r is the matrix $(X_m \times A_n)$, X_m is the set of cases, m is the number of
        'sample' cases, A_n is the set of attributes, n is number of attributes,
        and $X = (X_1, X_2, \ldots, X_i, \ldots, X_q)$
Output: Methodical sheet *MS* // *S* is subset of set MSs
begin
    load X_r
    repeat
     | **load** X_m
    until *Attributes check* ;
    do Search nearest case X_i `//use the nearest neighbour method`
    do Propose solution S
    while not *S accepted* **do**
     | **do** Repair S
    end while
    do *MS*
    do *MS*
end

Fig. 2 Algorithm of CBR Application

In CBR we used two ways for computing distance. Distance can be computed using Euclidean metrics. In the next step, the distance vector is normed to have the same norm. We could norm the vector by dividing every vector item by the sum of distance vector. The sum of the absolute values of the difference of attributes values in base of cases and attributes values of new case is the second method for computing distance, where output distance is normed by the maximum-possible case occurrence.

After finding the closest case, the algorithm offers two solutions which might be accepted by the expert, and then the program finishes and shows the solution and saves the solved case into the Base of Cases, or the expert corrects the offer. In this case, the solution is added into the Base of Cases and is shown as the solution, the program finishes again. Suggested solutions are: the first solution, which is taken from the basic Base of Cases which matches the precondition of similarity (a smaller distance than the threshold value of sensitivity), the second suggested solution is the case taken from the solved cases; it is the case which is the most similar to the new case.

The algorithm mentioned above was realized in MATLAB as the graphic user interface (GUI).

4   Conclusion

This article is focused on CBR based modelling. It concentrates on the area of fire protection. In this area it is usually very difficult to choose a correct solution. This is because the solution often depends on specific or unforeseeable factors and on the

experience of the commander of the action. We proposed a model for this activity and also we introduced the application. This application could be used as a solution for crisis situation.

Is the suggested solution correct? How can we represent experience and knowledge in the physical world we live in? What methods should we use for working with knowledge? These are the questions we might ask when looking for an answer during the problem solving activities in the field of the CM. Every model of decision making process is simplified, it is not precise, but fortunately some of them are serviceable.

It is obvious from Table 1 that the commanders of the action do not only follow the MSs, but long-lasting experience when solving similar situations also plays a big role in solving crisis situations. In the practical situations it could be easier to get knowledge from the expert in the form of a model rather than in the form of rules that are used in a standard expert system.

The CBR method offers the possibilities on how to record knowledge when defining 'sample' cases from the chosen field, and it allows the specification of 'new' cases with the aim to create their 'base' of general knowledge. It is represented by expert's knowledge on one side and the knowledge obtained from the data of that field on the other side.

As mentioned above, an application in the MATLAB environment was proposed in this paper. The GUI of this application is in the Czech language at this time because it is primarily designed for the Czech fire brigade. The algorithm described in Fig. 2 is implemented in the application and testing of this application proved that the suggested CBR method is functional. The mentioned CBR application could be used, e.g., when training a crew during the analysis of the check, or directly at the place of the crisis situation if the CBR application was implemented into specialized hardware (realized on the signal processor).

The future work could be focused on using of fuzzy logic (FL) and rough sets theory (RST) [14, 15, 17, 22]. The FL can be used for an expression of uncertainty into attributes values and RST for a definition 'sample' cases in the base of cases.

References

1. Barrier, G.: Emergency medical services for treatment of mass casualties. Critical Care Medicine 17, 1062–1067 (1989)
2. Carlsson, C., Fuller, R.: Fuzzy Reasoning in Decision Making and Optimization. Physica Verlag, New York (2002)
3. Fagin, D., et al.: Reasoning about Knowledge. Massetschussets Institute of Technology Press, Cambridge (1996)
4. Fire, Authority, E.S.: Annual report-state emergency services (2003), http://www.fesa.wa.gov.au/internet/upload/1746862705/docs/ses.pdf
5. Fire Rescue Service of the Czech Republic: Law no. 133 on fire protection (1985); In latter wording

6. Fire Rescue Service of the Czech Republic: Law no. 238 on fire rescue service of cr and on the modification of certain codes (2000); In latter wording
7. Fire Rescue Service of the Czech Republic: Law no. 239 on integrated rescue system and on the modification of certain codes (2000); In latter wording
8. Fire Rescue Service of the Czech Republic: Law no. 240 on crisis management and on the modification of certain codes (crisis code) (2000); In latter wording
9. Halpern, J.Y.: Reasoning about Uncertainty. Massetschussets Institute of Technology Press, Cambridge (2003)
10. Han, J., Kamber, M.: Data Mining. Elsevier, San Francisco (2006)
11. Kolodner, J.: Case-Based Reasoning. Morgan Kaufman, San Mateo (1993)
12. Krupka, J., Mohout, J.: Method of case-based reasoning and the fire protection unit field manual. In: Proceedings of the Conference on Crisis Management. Vitkovice v Krkonosich, Czech Republic (2006) (in Czech)
13. Nollke, M.: Entscheidungen treffen: schnell, sicher, richtig. Rudolf Haufe Verlag GmbH and Co. KG, München (2002)
14. Pal, S.K., Shiu, S.C.K.: Foundation of Soft Case-Based Reasoning. John Wiley & Sons, Incorporated, New Jersey (2004)
15. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publisher, Dordrecht (1991)
16. Rusell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, Englewood Cliffs (2002)
17. Slowinski, R.: Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory. Kluwer Academic Publisher, Dordrecht (1992)
18. Turban, E., Aronson, J.E., Liang, T.P.: Decision Support Systems and Intelligent Systems, 7th edn. Pearson Prentice Hall, New Jersey (2005)
19. United States Office of Personnel Management: Federal manager's – decision maker's emergency guide (2005),
http://www.opm.gov/emergency/pdf/ManagersGuide.pdf
20. Watson, I.: Applying Case-Based Reasoning: Techniques for Enterprise System. Morgan Kaufmann Publisher Incorporated, San Francisco (1997)
21. Witten, I.H., Frank, E.: Data Mining – Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann Publisher, Amsterdam (2005)
22. Zadeh, L.A.: Outline of a new approach to the analysis of complex systems and decision process. IEEE Transaction on Systems, Man, and Cybernetics 3(1), 28–44 (1973)

Enterprise Ontology According
to Roman Ingarden Formal Ontology

Jan Andreasik

Abstract. In the paper, original ontology is defined. This ontology is based on the concept apparatus of theory of object proposed by Roman Ingarden – a Polish philosopher. The author presents a definition of individual object according to this theory. In this definition, the following concepts are included: essence, form, way of existence, constitutive nature, subject of properties, endowment of properties, properties, state of thing, relation. By analogy, the author defines one's own concepts of the enterprise ontology: enterprise essence, form of competence system, content of assessments, enterprise vision, enterprise competence system, competence assessment, assessment criteria, score trajectories, grammar of trajectories, enterprise condition, competence potential condition, competence gap condition, act of explanation. According to the presented enterprise ontology, there are created descriptions of cases in the intelligent system for recognition of enterprise condition designed on the basis of the CBR (Case-Based Reasoning) methodology.

Keywords: ontology, enterprise, ERP, Roman Ingarden theory, case-based reasoning.

1 Introduction

In the design of business intelligence systems, especially the Executive Information Systems (EIS), the structure of Knowledge-Based Systems is being used. The basis for creating knowledge about an enterprise is ontology of the enterprise. There are many different approaches to this concept. Models proposed until now mostly

Jan Andreasik
Zamość University of Management and Administration
Akademicka 4, 22-400 Zamość, Poland
e-mail: `jandreasik@spp.org.pl`

focused on conceptualization of the processes as well as on transactions realized by an enterprise. This type of ontology [27] is used in enterprise information systems such as Enterprise Resource Planning (ERP). The knowledge about an enterprise is collected through a process of data mining [30] or by concluding from business cases using the Case Based Reasoning (CBR) methodology [22].

The main question raised by the stakeholders is about the current and future economic condition of an enterprise. The most common models used to answer that question are prediction models of enterprise bankruptcy. The knowledge obtained from expert systems lies only within a discrimination function [1], which has been worked out in most cases in a learning process based on financial ratios. The financial analysis focuses only on ratios and results, presented in the basic financial reports such as the balance sheet or profit and loss account. These ratios analyze only a process of the capital and asset management as well as focus on calculation of a net profit. More information can be obtained through strategic analysis using the Balanced scorecard (BSC) method [17]. This approach focuses on operational, financial, marketing and development processes.

The author of this paper, when constructing the system of enterprise features, is following J.M. Bocheński approach [7]. He defines an industrial enterprise as a set of six elements: capital, labor, customers, region, country and invention. The author presents his original concept of enterprise ontology incorporated in R. Ingarden formal ontology [18]. This ontology defines three types of objects: individual object, intentional object and object determined in time. The author suggests that an enterprise can be defined using concepts that determine individual objects: Constitutive nature, Properties, Essence, State of object, Subject of properties, Relation, Endowment of properties. Using R. Ingarden formal ontology in defining an enterprise as an individual object, there is a need to answer following questions: What will be the subject of properties? How to define properties? How to determine state of thing with reference to an enterprise and how many states are needed to distinguish? How to define the relation among a given object and other objects in order to clarify their features? What makes constitutive nature of an enterprise? How to take hold of enterprise essence?

The author presents his own definitions that form original enterprise ontology, which is used to create a library of enterprise cases (descriptions of an enterprise) for building the intelligent system for recognition of enterprise condition designed on the basis of the CBR methodology.

2 Definition of Object in Roman Ingarden Formal Ontology

R. Ingarden has defined three types of objects: Individual object, Intentional object, Object being in time. Every object has extensive definitions. In this section, the author disregards for definitions of a philosophical nature and put together a concept system in order to go to analogy between an individual object defined by R. Ingarden and an enterprise defined in the next section.

An individual object is a pair $PI = \langle F, M \rangle$, where F is a form and M is a matter. R. Ingarden gave nine definitions of pairs: a form and a matter, respectively. The first form determines a subject of properties (features). A matter is the content, essence represented by a constitutive nature of an object. A constitutive nature of an object makes up a concept completely and originally assigned to only one object. It has to determine a subject of properties in the utter way. It can be admitted that there exists a number of the so-called quasi-natures of the same object. There is determined association of unity between a form and a matter. There are distinguished four types of unity: actual unity, essential unity, functional unity, and harmonic unity. For each subject of properties, there exists a set of properties $PS = \{P_i\}$. An object property has not only a form connected with determining a feature F, but it has also a matter called its endowment E: $P = \langle F, E \rangle$.

R. Ingarden showed that there exists a subjective form of an object, which is called state of thing ST. State of thing is a function r called a 'relation' determined by the external subject of action. For an object represented by a subject of properties PS, there are assigned properties: $r : PS \rightarrow P$. In a set of states of things there are possible positive states of things as well as negative states of things. R. Ingarden defines a relation as multi-object state of thing. The thing is that in the comparison process among an object and other objects a given property is assigned to the object. There are distinguished the following kinds of properties: relative features, properties conditioned externally, properties acquired, properties of an individual object strictly itself own. R. Ingarden defines also a concept of essence of an individual object. Essence of an object includes three elements: form, matter, and a way of existence connected with them. One can distinguish: radical essence, strict essence, essence purely material, simple essence, a set of characteristic features.

Figure 1 shows a graphical diagram of concepts making up definition of an individual object. The author presents this diagram on the basis of analysis of R. Ingarden's paper [18].

The theory of object emphasizes strongly a problem of determining properties (features). Therefore, a key concept is determining a subject of properties. Until now, in enterprise ontologies a subject was an enterprise architecture (TOVE ontology) [14], processes of exchange of resources (REA ontology) [11], patterns of transactions (Ψ-theory) [10], knowledge resources forming an enterprise memory (BusCO) [19].

Fig. 1 A concept system of definition of object according to R. Ingarden's theory of object [18]
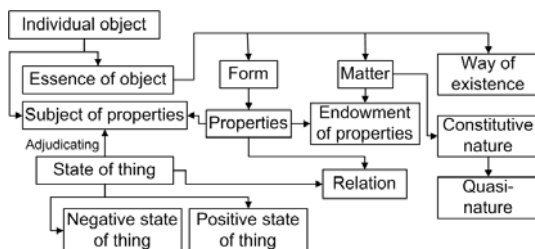
In this paper, the author presents the original enterprise ontology, in which a subject is a competence system. Modern enterprise theories are the theoretical basis of this approach. These theories are based on conceptions of competence, knowledge resources, and intellectual capital [9]. Until now, ontologies and description languages focus on an internal structure of an enterprise delivering concept apparatus concerning organization, planning and management of human resources [16, 25]. The author's approach is oriented to recognition of enterprise condition disregarding for its architecture. An assessment of condition is made by experts on the basis of information from data warehouses, which are prepared by the so-called agents according to fixed procedures.

3   Enterprise Ontology

Below, concepts of defined enterprise ontology are given.

3.1   Enterprise Competence System

A choice of a subject of properties is a key issue for a process of defining enterprise ontology. Ontology in computer science is defined as a specification of conceptualization. Therefore, a subject is the representative of an object. It identifies its conception. In order to make assessment (this is a goal of conceptualization) it is needed to define a subject in a way that it is possible to assign to a subject a set of properties (features). In the author's conception there exist two classes of properties: assessment of competence potential and assessment of competence gap. Competence potential of an enterprise is assessed by the expert in a given range, which is convergent with a strategy analysis according to the SWOT method [31] including strengths and opportunities. A competence gap is assessed by the expert in a given range, which is convergent with a strategy analysis according to the SWOT method including weaknesses and threats (risks). A subject of properties is defined by two taxonomies of ranges of competence potential and ranges of competence gap. A definition of a subject is given in the BNF (Backus-Naur Form) notation [20]:

```
<subject>::=<Enterprise Competence System ECS>
<ECS>::=<competence potential>,<competence gap>
<competence potential>::=<type of competence potential>
   <kind of competence potential>
   <range of competence potential>
<competence gap>::=<type of competence gap>
   <kind of competence gap><range of competence gap>
```

According to the definition of an enterprise given by J.M. Bocheński [7] the author proposes to distinguish five spaces of assessment of the enterprise competence system:

```
<type of competence potential>:=<capital potential CP> |
  <innovation and investment potential IP> |
  <key stakeholder potential SP> |
  <neighborhood potential NP> |
  <environment potential EP>
<type of competence gap>:=
  <in the range of risk of keeping capital CG> |
  <in the range of risk of project realization PG> |
  <in the range of risk of key stakeholder requirements SG>|
  <in the range of risk from neighborhood NG> |
  <in the range of risk from environment EG> |
```

For each type of competence potential and for each type of competence gap, respectively, there exists an assigned list of kinds. For each kind of competence potential in a given type, there exists an assigned list of detailed ranges. For each kind of competence gap in a given type, there exists an assigned list of detailed ranges [12].

3.2 Competence Assessment (Properties)

In each range of competence potential and in each range of competence gap there are determined: elementary competence potential and elementary competence gap. One can determine two kinds of properties:

- weights assigned to each type, kind, and range of competence potential and competence gap, respectively,
- comparative assessments of enterprise competence in each range.

```
<competence assessment>::=<weights>|<comparative assessments>
```

According to R. Ingarden formal ontology, weights make up expert assessments of the significance of elementary competencies and elementary competence gaps in the assessment of the whole enterprise. However, comparative assessments are made by the expert on the basis of current multi-criteria analysis of a competence system in relation to the target state of a competence system determined in the enterprise strategy. Criteria in multi-criteria analysis are elements of properties of endowment. Comparative assessments are determined like assessments in the AHP method [28], which is recommended in performing state analysis.

3.3 Assessment Criteria (Endowment of Properties)

In order to make comparative assessment of enterprise competence in a given range, it is needed to perform multi-criteria analysis. A system of criteria makes up the so-called material endowment of properties according to R. Ingarden formal ontology. The following definition of endowment is proposed:

```
<endowment of properties>:=
   <possibility of using employee resources UR>,
   <qualifications of employee resources QR>,
   <abilities of employee resources AR>,
   <experience of employee resources ER>,<technology T>
<UR>:=<available own resources>,
   <partly available own resources>,
   <available outside resources>,
   <partly available outside resources>,
   <hardly available outside resources>
<QR>:=<basic qualifications>,
   <specialist, certified qualifications>,
   <unique qualifications>
<AR>:=<team work>,<ability of cooperation>,<creativity>,
   <responsibility>,<decision making>,<commitment>,
   <solution searching>,<articulatity>
<ER>:=<existence of successes>,<the length of employment>,
   <recommendations>,<rewards and honorable mentions>
<T>:= <procedures>,<methods and processes>,
   <procedures of machine and device services>,
   <computer aided systems>,<unique recipes>
```

3.4   Enterprise Condition (State of Thing)

Enterprise condition, according to Ingarden's conception, is determined by an external subject of action, which is, in our case, the expert making assessment of enterprise competence. Condition of an enterprise is a set of competence assessments made by the expert in given ranges and according to proper criteria.

```
<assessment of enterprise competence>::=
   <range of competence potential>|<range of competence gap>|
   <assessment value>
```

An assessment value is a number from the interval $[0, 1]$, which is determined on the basis of multi-criteria analysis by means of the AHP methods. One can use a computer aided system EXPERT CHOICE. The AHP method is used commonly in the assessment process [8, 13, 21, 23, 29]. According to Ingarden's conception, the author of this paper makes division of assessment of the enterprise competence system into two states. A positive state which includes assessment of competence potential and a negative state which includes assessment of competence gap. The presented enterprise ontology makes up a basis for construction of an intelligent system for recognition enterprise condition according to the CBR (Case-Based Reasoning) methodology [24]. In this methodology, there is created a case base. Each case is defined according to this enterprise ontology. A case includes a set of states of an enterprise defined by different experts in different time. Owing to assessment of competence potential and competence gap in a large set of ranges, there is a need to aggregate assessments. Aggregation algorithms use different similarity

measures. The author in his papers [2, 4] showed aggregation methods based on theory of multi-criteria decision making. The effect of aggregation is a set of identifying rates, which make up identifier of enterprise condition. In the retain process of a case base (in the CBR methodology) one can use identifiers for retrieve the most similar case to a given one.

```
<identifier of enterprise condition>::=
    <a set of identifying rates>
```

3.5 Explanation Act

R. Ingarden defines state of thing, in which more than one object takes part. In knowledge discovery processes on an enterprise, we deal with generalization of enterprise states in order to make classification. Assigning a given enterprise to a given class is connected with the knowledge discovery on a current or future enterprise behavior. A lot of approaches concerns classification of enterprises into two classes: enterprises in good condition and enterprises being in a bankruptcy state. There exists a lot of methods of classification. Methods which use theory of multi-criteria decision making are presented in [33]. In present ontology, a relation of a considered enterprise to a group of enterprises making up a cluster is defined.

```
<explanation act of enterprise condition>::=
    <identifier of enterprise condition>
    <cluster characteristic>
```

A relation defined following R. Ingarden as multi-subject state of thing makes up the explanation act of an enterprise condition. An explanation act procedure of an enterprise condition is presented by the author in [5].

3.6 Enterprise Score Trajectories (Constitutive Nature of Object)

The best characteristic of an enterprise is made by financial reports: a profit and loss account and a balance. Interpretation of these reports is made by financial analysis. There exists difficulty in making assessment of the enterprise condition on the basis of financial rates. It is needed to build a benchmark system, in which values of rates are comparable with suitable average, minimum, or maximum values in a group of enterprises. Such a system has been constructed in Zamość University of Management and Administration for the SME enterprises from Podkarpacie and Lublin Provinces as a part of the e-barometer system [12]. However, in order to make deeper analysis of enterprise scores it is needed to draw up functions of scores in time and to determine extrema, intervals of increasing and decreasing, etc. An interesting construction of analysis of enterprise scores has been presented by J. Argenti [6]. He distinguishes three characteristic types of elementary score trajectories. Other authors [26] determined four characteristic types of elementary score trajectories by means of specific metaphors: bullfrog, drowned frog, boiled frog, tadpole.

These characteristic types have been connected with an enterprise profile expressing its constitutive nature. In defined ontology, constitutive nature of an enterprise is defined by means of grammar expressing notation of a generalized score trajectory of an enterprise. This generalized trajectory arises from multi-criteria analysis of component trajectories, which are time functions of net profit, investor's capital, assets, operating capital, EBIDTA, income. Nonterminal symbols are characteristic score trajectories, however terminal symbols are four segments of score increasing or decreasing in three consecutive years.

```
<score trajectory>::=<grammar G>
```

$G = \langle T, N, P, ST \rangle$, where: T – a set of terminal symbols: $\{s1$ – strong increasing, $s2$ – weak increasing, $s3$ – strong decreasing, $s4$ – weak decreasing$\}$, N – a set of nonterminal symbols (a set of characteristic score trajectories), P – a set of production, ST – a generalized score trajectory (an initial symbol). The author presents an exact definition of the grammar G in other papers.

3.7 Enterprise Vision

In addition to a form, R. Ingarden determines a way of existence in definition of essence of object. This way of existence is connected with a process of occurring some states determined by, among others, other objects of the same domain of being. An important point of reference in determining properties (features) of an enterprise, in our case enterprise competence, is a strategy. In categories of R. Ingarden's theory of object a strategy can be captured by means of concept apparatus of an intentional object. J. Wojtysiak [32] presents an attempt to depiction of ontology of money by means of concept apparatus of an intentional object. An original conception of depiction of an enterprise strategy through defining its vision has been presented by G. Hamel and C.K. Prahalad [15]. In present ontology, enterprise vision is depicted by means of the same structure of ranges of competence potential and competence gap like in a case of defining the enterprise competence system. It allows the expert to make assessment in relation to elaborated vision in strategy documents. Definitions of concepts presented above make up original enterprise ontology based on R. Ingarden's theory of object. A graphical diagram of this ontology is shown in Fig. 2.
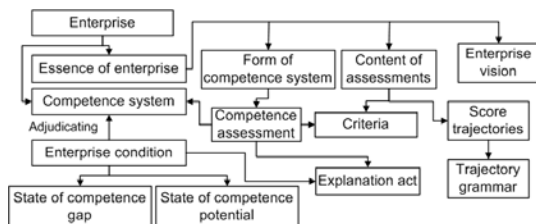
Fig. 2 A graphical diagram of enterprise ontology modeled on R. Ingarden's formal ontology

4 Conclusions

In the present paper, a concept system of enterprise ontology elaborated by the author is shown. This system refers to Roman Ingarden's formal ontology. A goal of this approach is to determine a way of formalization of the knowledge of an enterprise. Defined enterprise ontology makes up a basis for constructing an intelligent system of recognition of enterprise condition based on the Knowledge-Based System (KBS). The author presents a structure of the system in [2, 3, 4, 5]. In preparation of presented enterprise ontology, the author used papers of Polish philosophers: R. Ingarden [18], J.M. Bocheński [7], and J. Wojtysiak [32].

References

1. Altman, E.I.: Corporate financial distress and bankruptcy. John Wiley & Sons, Incorporated, Chichester (1993)
2. Andreasik, J.: A case base reasoning system for predicting the economic situation on enterprises – tacit knowledge capture process (externalization). In: Kurzyñski, M., et al. (eds.) Computer Recognition Systems, vol. 2, pp. 718–730. Springer, Heidelberg (2007)
3. Andreasik, J.: Enterprise ontology – diagnostic approach. In: Proceedings of the Conference on Human-System Interactions, pp. 497–503. Cracow, Poland (2008)
4. Andreasik, J.: Inteligent system for predicting economic situation of SME. In: Józefczyk, J., et al. (eds.) Proceedings of the Congress of Cybernetics and Systems of WOSC. Wroclaw, Poland (2008)
5. Andreasik, J.: Decision support system for assessment of enterprise competence. In: Kurzyński, M., Woźniak, M. (eds.) Computer Recognition Systems 3. AISC, vol. 57, pp. 559–567. Springer, Heidelberg (2009)
6. Argenti, J.: Corporate collapse. The Causes and Symptoms. McGraw-Hill, New York (1976)
7. Bocheński, J.M.: Przyczynek do filozofii przedsiębiorstwa przemysłowego, pp. 162–186. Państwowe Wydawnictwa Naukowe, Warsaw (1993) (in Polish)
8. Carlucci, D., Schiuma, G.: Knowledge assets value creation map assessing knowledge assets value drivers using AHP. Expert Systems with Applications 32, 814–821 (2007)
9. Choo, C.W., Bontis, N.: The strategic management of intellectual capital and organizational knowledge. Oxford University Press, Oxford (2002)
10. Dietz, J.L.G.: Enterprise ontology. Springer, Heidelberg (2006)
11. Dunn, C., Cherrington, J.O., Hollander, A.S.: Enterprise Information Systems. A Pattern-Based Approach. McGraw-Hill, New York (2003)
12. E-barometr Project: e-barometr manual, http://www.wszia.edu.pl/eng/files/e-barometr-manual.pdf
13. Ertugrul, I., Karakasoglu, N.: Performance evaluation of Turkisch cement firms with fuzzy analytic hierarchy process and TOPSIS methods. Expert Systems with Applications 36, 702–715 (2009)
14. Gruninger, M., Atefi, K., Fox, M.S.: Ontologies to support process integration in enterprise engineering. Computational & Mathematical Organization Theory 6, 381–394 (2000)
15. Hamel, G., Prahalad, C.K.: Competing for the Future. Harvard Business School Press (1994)

16. Harzallah, M., Berio, G., Vernadat, F.: Analysis and modeling of individual competencies: Toward better management of human resources. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 36(1), 187–207 (2006)
17. Huang, H.C.: Designing a knowledge-based system for strategic planning: A balanced scorecard perspective. Expert Systems with Applications 36, 209–218 (2009)
18. Ingarden, R. (ed.): Spór o istnienie świata. Tom II Ontologia formalna, cz. 1. Forma i Istota. Państwowe Wydawnictwo Naukowe, Warsaw (1987) (in Polish)
19. Jussupova-Mariethoz, Y., Probst, A.R.: Business concepts ontology for an enterprise performance and competences monitoring. Computers in Industry 58, 118–129 (2007)
20. Knuth, D.E.: Backus normal form vs. Backus Naur form. Communications of the ACM Archive 7(12), 735–736 (1964)
21. Lee, A.H., Chen, W.C., Chang, C.J.: A fuzzy AHP and BSC approach for evaluating performance of IT department in manufacturing industry in Taiwan. Expert Systems with Applications 34, 96–107 (2008)
22. Li, H., Sun, J.: Ranking-order case-based reasoning for financial distress prediction. Knowledge-Based Systems 21, 868–878 (2008)
23. Lin, M.C., Wang, C.C., Chen, M.S., Chang, C.A.: Using AHP and TOPSIS approaches in customer-driven product design process. Computers in Industry 59, 17–31 (2008)
24. Pal, S.K., Shiu, S.C.K.: Foundations of Soft Case-Based Reasoning. Wiley-Interscience, Hoboken (2004)
25. Pepiot, G., Cheikhrouhou, N., Furbringer, J.M., Glardon, R.: UECML: Unified enterprise competence modelling language. Computers in Industry 58, 130–142 (2007)
26. Richardson, B., Nwankwo, S., Richardson, S.: Understanding the causes of business failure crises, generic failure types: boiled frogs, drowned frogs, bullfrogs and tadpoles. Management Decision 32(4), 9–22 (1994)
27. Rittgen, P.: Handbook of ontologies for business interaction. Information Science Reference, Hershey (2008)
28. Saaty, T.L.: Decision making for leaders. RWS Publications (2001)
29. Thomaidis, F., Mavrakis, D.: Optimum route of south transcontinental gas pipeline in SE Europe using AHP. Journal of Multi-Criteria Decision Analysis 14, 77–88 (2006)
30. Wen, W., Chen, Y.H., Chen, I.C.: A knowledge-based decision suport system for measuring enterprises performance. Knowledge-Based Systems 21, 148–163 (2008)
31. Williamson, D., Cooke, P., Jenkins, W., Moreton, K.M.: Strategic Management and Business Analysis. Elsevier, Amsterdam (2004)
32. Wojtysiak, J.: Filozofia i życie. ZNAK. Cracow, Poland (2007) (in Polish)
33. Zopounidis, C., Dimitras, A.I.: Multicriteria Decision Aid Methods for the Prediction of Business Failure. Kluwer Academic Publishers, Dordrecht (1998)

Hand Shape Recognition
for Human-Computer Interaction

Joanna Marnik

Abstract. The paper presents a novel method which allows to communicate with computers by means of hand postures. It is assumed that an input to the method is a binary image of a hand presenting a gesture. A curvature of a hand boundary is analysed in the proposed method. Boundary points which correspond to the boundary parts with specified curvature are used to create a feature vector describing a hand shape. Feature vectors corresponding to shapes which are to be recognised by a system are recorded in a model set. They serve as patterns in a recognition phase. In this phase an analysed shape is compared with all patterns included in the database. A similarity measure, proposed specifically for the method, is used here. One advantage of the method is that it allows to easily add a shape to the recognised shapes set. Moreover, the method can be applied to any shapes, not only hand shapes. The results of the tests carried out on the posture database, which includes 12 664 images of 8 hand shapes, are also presented in the paper.

Keywords: human-computer communication, shape recognition, similarity measure, binary images.

1 Introduction

In recent years, more natural ways of communicating with a computer than a keyboard and a mouse have become more and more important. Multitouch screens can serve this purpose, but they are not in a very wide use yet because of their cost [3].

Joanna Marnik
Rzeszów University of Technology,
Wincentego Pola 2, 35-235 Rzeszów, Poland
e-mail: jmarnik@prz-rzeszow.pl

Speech is the most natural way of communicating with others. A speech recognition technology is built in the Windows Vista operating system [9]. Such systems have some limitations concerning other voices and sounds appearing in the environment. Another means of communicating with machines are hand gestures. A technology allowing to control mobile phones or computers in this way has recently been patented by Sumsung [2]. Practical application of hand gestures recognition methods can be found on the Internet, to navigate a website, specifically prepared for this purpose [11].

There are a few approaches to resolving the problem of hand posture recognition. Wilkowski [14] uses a Linear Discriminant Analysis and a statistical maximum likelihood classifier to recognise 9 hand postures with 92.3% accuracy. Features used to build a feature vector in this method do not allow rotations and scale changes of the hand, and they cannot be applied to bigger sets of postures. Moreover, a different classifier must be learnt for each set of the postures under consideration.

Another approach uses morphological 'hit-or-miss' operation to analyse a hand shape and a neural network or a decision tree as a classifier [4, 8]. This approach was applied to recognise 8 hand postures which can be used in human-computer interaction [4] with a recognition ratio of 96.7% for the decision tree and 96.4% for the neural network. In [8], 20 hand shapes occurring in Polish finger alphabet (PFA) were recognised with about 87% accuracy by means of this method. The drawback of the method is that a neural network and a decision tree are also closely related to the set of the recognised postures.

Triesh and Malsburg use the elastic graph matching (EGM) technique to recognise 12 hand shapes [13] with a recognition ratio of 92.9% and 85.8% for the postures performed against simple and complex background, respectively. An advantage of the method is that the background of the hand can be complex. However, the method's computational time is too long for a real-time recognition.

This approach was applied to recognise hand shapes occurring in PFA [6]. Besides Gabor filters, edge information and depth information were also used for the graph nodes description. A recognition ratio of 80.9% was achieved in the experiment presented in the paper [6].

The paper proposes a novel method for recognising hand shape. The method allows to communicate with computers by means of hand postures. A set of recognised hand shapes can be defined by a user of the method. The method is also suitable for any shapes, not only hand shapes.

It is assumed that a binary image of a hand presenting a gesture is the input to the method in question. The binary image can be obtained, for example, by background subtraction [10] or using a skin colour model [7]. A curvature of a hand boundary is then analysed to build a feature vector. Feature vectors corresponding to shapes, which are to be recognised by the system, are recorded in a model set. They serve as the patterns in a recognition phase. In this phase, the analysed shape is compared with all patterns included in the model set. A similarity measure, proposed specifically for the method, is used here. A detailed description of the method is given in

2 Method Description

The proposed method of a hand shape recognition consists of three stages. In the first
stage, a set of a high curvature points is determined. On the basis of this set a feature
vector is created in the next stage. Feature vectors obtained for sample images of the
considered hand shapes are stored in a model set, which is used during a recognition
stage.

2.1 High Curvature Points Detection

A binary image of a hand posture is the input to the method. At the beginning, a
boundary points sequence is obtained with extended boundary tracing method [12].
The method is fast because it is based on a look-up table. The sequence of points
serves as an input to the high curvature points detection algorithm [1]. It is a two-pass
algorithm. In the first phase, candidate points are extracted. The points are depicted
by the apex of the isosceles triangle of specified size and opening angle inscribed in a
boundary. Such an approach allows to easily determine if the point lays on a concave
or convex part of the boundary. A second pass is a post-processing step aiming at
removing superfluous candidates. The remaining points are the desired high curvature
points. There are 3 parameters of the algorithm: L_{min}, L_{max} – minimal and maximal
length of the triangle's both sides, respectively, and α – maximal acceptable opening
angle between the sides of the triangle. An example binary image with candidate
points obtained after the first pass of the algorithm is shown in Fig. 1a.

High curvature points can be seen in Fig. 1b. Circles correspond to the convex
parts of the boundary, squares to the concave parts.

| (a) | (b) |
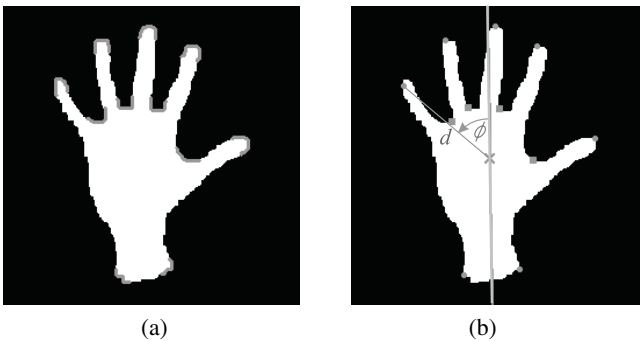
Fig. 1 (a) An example image with candidate points. **(b)** High curvature points and feature
vector construction

2.2   Structure of a Feature Vector

A binary image of a gesture and a set of high curvature points obtained for this image
are used to build a feature vector. First, a centroid and a slope of the major axis of
an object representing a given hand shape are calculated. Taking into account these
two values allows to make the method insensitive to rotation and location of the
object. The feature vector F consists of triples $f_i = (d, \phi, c)$, where $i = 1, 2, \ldots, n$ are
indices of successive high curvature points, d is a normalised distance of the given
point from the centroid, ϕ is the value of an angle created by the major axis and a
ray coming out from the centroid and going through the point, and c is a convexity
of the point. Elements d and ϕ can be treated as a vector $\mathbf{v}$ in a coordinate system
originating at the centroid of the object and X axis determined by the major axis
of the object. The normalisation of distance values consists in dividing the original
value by the greatest of the distances obtained for the image's high curvature points.
A feature vector construction is illustrated in the image Fig. 1b.

The feature vectors obtained for sample images of hand shapes, which are to be
recognised with their labels are stored in the model database $\mathcal{M}$. This database is
used in the recognition stage.

2.3   Recognition

At a recognition stage a similarity measure described below is used to calculate a
similarity value of two feature vectors. Because the number of high curvature points
can be different for different images, a special measure was proposed for the method.

2.3.1   Similarity Measure

The similarity $S(F_1, F_2)$ of two feature vectors F_1 and F_2 is an arithmetic average of
similarities $S(F_1 \rightarrow F_2)$ and $S(F_2 \rightarrow F_1)$ of the model image to the examined image
and the examined image to the model image, respectively:

$$S = \frac{S(F_1 \rightarrow F_2) + S(F_2 \rightarrow F_1)}{2} . \tag{1}$$

Similarity $S(F_1 \rightarrow F_2)$ is an average distance of all high curvature points included
in F_1 to the nearest high curvature point included in F_2 with the same convexity and
the angle value close to the angle corresponding to the given point from F_1:

$$S(F_1 \rightarrow F_2) = \frac{\sum_{i=1}^{n} DIST_{\min}(f_i, F_2)}{n} , \tag{2}$$

where

$$DIST_{\min}(f_i, F) = \begin{cases} |\mathbf{v}(f_i) - \mathbf{v}(f_{j\min}(F))| & \text{if } |\phi(f_{j\min}(F)) - \phi(f_i)| \leq \theta \\ d(f_i) & \text{otherwise} \end{cases} \tag{3}$$

and

$$\text{jmin} = \underset{j=1,\dots,n_F}{\arg\min} \left(|\mathbf{v}(f_i) - \mathbf{v}(f_j(F))| : c(f_i) = c(f_j(F)) \right) \tag{4}$$

θ determines a range in which the angles of the vectors $\mathbf{v}(f_i)$ and $\mathbf{v}(f_j(F))$ are recognised as compatible. $d(f_i)$, $\phi(f_i)$ and $c(f_i)$ are the distance, the angle and the convexity of the point related to the triple f_i, respectively. $\mathbf{v}(f_i)$ is a vector determined by the values d and ϕ of the triple f_j, $f_j(F)$ is jth triple of the feature vector F, and n_F is a number of triples in F.

The proposed similarity measure gives the smallest values for more similar shapes.

2.3.2 Recognition stage

During the recognition stage, we calculate similarity values of the feature vector, obtained for the examined image, to each of the feature vectors stored in the model set $\mathcal{M}$. If the smallest of such obtained values is below specified threshold T, it is assumed that the result of the recognition is the shape related to the feature vector for which this value was achieved. Otherwise, the image remains unclassified.

3 Experiments

Experiments were performed on the image database compound of 14 360 binary images of 8 hand shapes presented in Fig. 2.

The resolution of the images was 256×256 pixels. This image database was earlier used in experiments concerning a hand shape recognition, described in [4]. A method using morphological 'hit-or-miss' operation was used to generate a feature
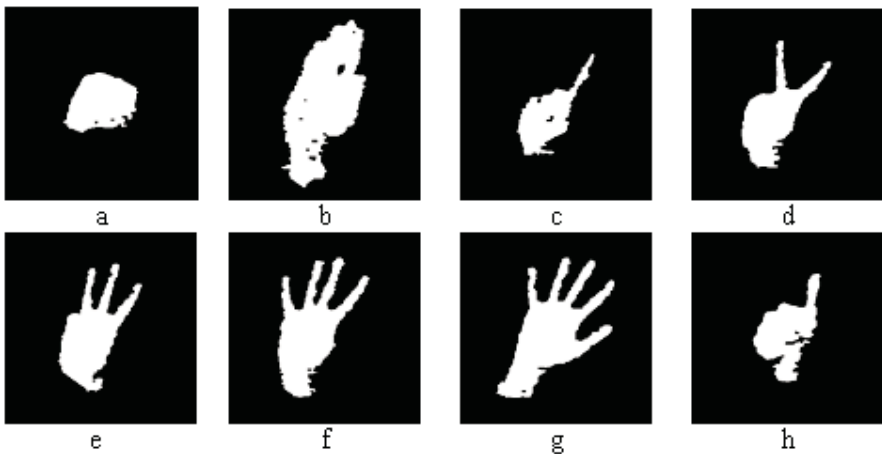
Fig. 2 Hand shapes used in the experiments: **a** wrist, **b** palm, **c–g** 1–5 fingers, **h** thumb

Table 1 The number of images of particular hand shapes in the test set

1 *finger*	2 *fingers*	3 *fingers*	4 *fingers*	5 *fingers*	*wrist*	*palm*	*thumb*
1751	1354	1561	1635	1410	1551	1290	2112

vector [8]. To classify a hand shape, an MLP neural network and a decision tree were used.

The image database contains two groups of images: a group of 1696 training images (212 for each of the 8 postures) and a group of 12 664 test images. The number of the test images of individual hand shapes is collected in Table 1.

The set of the model feature vectors was created on the basis of the training images. First, one image for each hand shape was chosen to create an 8-element model set. Next, the feature vectors obtained for each of the remaining training images were classified using the proposed method. The feature vectors for which the classification was faulty were added to the model set. As a result, the set of the model feature vectors contained 10, 7, 9, 11 and 5 elements for the 1–5 fingers respectively, 10 elements for *wrist*, 5 for *palm*, and 12 for *thumb*. All the images were earlier filtered by a morphological CO operation of size 7×7. The parameters of the method were set as follows: $L_{\min} = 15$, $L_{\max} = 40$, $\alpha = \frac{3}{4}\pi$, $\theta = \pi/9$ (see Sect. 2). $L_{\min}$ and $L_{\max}$ depended on the size of the objects representing the hand. α and θ had default values. The threshold T for the similarity value was set to 0.15.

The experiments were carried out on test images. Parameters values, the same as the ones used during creating a set of the model feature vectors, were used here. To make a comparison with the results presented in [4] possible, the threshold T was neglected. By considering this parameter, we obtain unrecognised gestures, which was not the case during the experiments to which the comparison is to be made. We expect that using this threshold will allow to depict moments of transition between the known postures during a continuous recognition in a video sequence.

Table 2 Results of the recognition obtained for the test images from the database

Correct [%]	1 *finger*	2 *fingers*	3 *fingers*	4 *fingers*	5 *fingers*	*wrist*	*palm*	*thumb*	*sum*
Our method	99.3	97.9	94.8	97.9	99.9	99.7	98.2	97.4	98.1
MLP	97.2	92.2	92.1	95.8	96.9	99.9	98.4	98.4	96.4
Decision tree	94.4	95.0	96.2	97.6	98.8	98.3	97.8	96.1	96.7

The results of our experiment are presented in Table 2. The rows present names of the methods and the percent of correctly classified images. MLP and Decision tree values are taken from [4]. The values obtained for all the test images are given in the last column. For almost all the postures the best results are obtained by means of the proposed method. Only for the postures *3* and *5 fingers* better results were achieved

Fig. 3 A test image of *3 fingers* posture classified as *4 fingers*
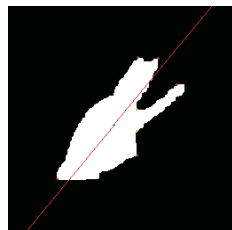
for 'hit-or-miss' features and the decision tree classifier. Also postures *wrist* and *palm* were only slightly better recognised by the MLP classifier.

The most often misclassified shape was *3 fingers*. It was confused mainly with *2* and *4 fingers*, which are the postures most resembling *3 fingers* posture. This posture was also confused with *palm*. It occurred when the outstretched fingers are close to each other. An example image of *3 fingers* posture classified as *palm* posture is depicted in the Fig. 3.

4 Conclusions

The presented method allows for the recognition of hand shapes and can be used in a human–computer interaction. In the experiments performed on 8 hand postures, high recognition rates were achieved. However, the method can be applied to other sets of postures. Another advantage of the method is that it allows to easily add a posture to the model feature vectors set. Learning a classifier is not necessary here. Only a feature vector corresponding to the new gesture has to be added to that set. It can be done online while using a hand gesture recognition system. It was possible neither in case of the 'hit-or-miss' method [4, 8] nor with the method proposed by Wilkowski [14].

The method is suitable for a real–time recognition. In the experiments (Sect. 3 an average time of the classification of a single frame equalled 27 ms.

We are going to use this method to recognise hand shapes occurring in the Polish finger alphabet. Previous works [8] showed difficulty in recognising some postures from this set. We hope that the proposed method will provide better results, but more research is needed.

The drawback of the method is that it is based on binary images. They must be obtained on the basis of color camera images. Different methods can be used to do that, i.e. background subtraction [10] or a method using a skin colour model [7]. A quality of these images are crucial for the method. All irregularities of the boundary can lead to misclassification. A CO filter was applied to smooth the boundary in the above experiments, but more sophisticated methods can give better results, for example Latecki's Polygon Evolution by Vertex Deletion method [5].

References

1. Chetverikov, D., Szabo, Z.: A simple and efficient algorithm for detection of high curvature points in planar curves. In: Proceedings of the 23rd Workshop of the Austrian Pattern Recognition Group, pp. 175–184 (1999)
2. Heon, K.S., Gumi-si, K.R., Sul, K.T., Imsil-gun, K.R.: Hand gesture recognition input system and method for a mobile phone, patent 20080089587 (2008), http://www.freepatentsonline.com/20080089587.html
3. Hodges, S., Izadi, S., Butler, A., Rrustemi, A., Buxton, B.: ThinSight: Versatile multi–touch sensing for thin form–factor displays. In: Proceedings of the ACM Symposium on User Interface Software and Technology, pp. 259–268 (2007)
4. Kapuściński, T., Marnik, J., Wysoki, M.: Rozpoznawanie gestów rąk w układzie wizyjnym. Pomiary Automatyka Kontrola 1, 56–59 (2005)
5. Latecki, L.J., Lakämper, R.: Polygon evolution by vertex deletion. In: Nielsen, M., Johansen, P., Olsen, O.F., Weickert, J. (eds.) Proceedings of the International Conference on Scale-Space in Computer Vision. Springer, London (1999)
6. Marnik, J.: The Polish finger alphabet hand postures recognition using elastic graph matching. In: Kurzyński, M., et al. (eds.) Computer Recognition Systems. Advances in Soft Computing, vol. 2, pp. 454–461. Springer, Heidelberg (2007)
7. Marnik, J., Kapuściński, T., Wysoki, M.: Rozpoznawanie skóry ludzkiej na obrazach cyfrowych. In: Materiały 5tej Krajowej Konferencji Naukowo-Technicznej Diagnostyka Procesów Przemysłowych, pp. 279–282. Łagów, Lubuski (2001)
8. Marnik, J., Wysoki, M.: Hand posture recognition using mathematical morphology. Archiwum Informatyki Teoretycznej i Stosowanej 16(4), 279–293 (2004)
9. Microsoft Corporation: Windows speech recognition, http://www.microsoft.com/enable/products/windowsvista/speech.aspx
10. Piccardi, M.: Background subtraction techniques: a review. In: Proceedings of IEEE Inernational Conference on Systems, Man and Cybernetics, The Hague, The Netherlands (2004)
11. Publics & Halriney, http://www.hrp.com
12. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. Thomson Engineering, Toronto (2007)
13. Triesch, J., von der Malsburg, C.: A system for person-independent hand posture recognition against complex backgrounds. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(12), 1449–1453 (2002)
14. Wilkowski, A.: An efficient system for continuous hand-posture recognition in video sequences. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J. (eds.) Computational Intelligence: Methods and Applications, pp. 411–422. AOW EXIT, Warsaw (2008)

System for Knowledge Mining in Data
from Interactions between User and Application

Ilona Bluemke and Agnieszka Orlewicz

Abstract. The problem of knowledge extraction from the data left by web users during their interactions is a very attractive research task. The extracted knowledge can be used for different goals such as service personalization, site structure simplification, web server performance improvement or even for studying the human behavior. The objective of this paper is to present a system, called ELM (Event Logger Manager), able to register and analyze data from different applications. The registered data can be specified in an experiment. Currently ELM system provides several knowledge mining algorithms, i.e., apriori, ID3, C4.5 but easily other mining algorithms can be added.

Keywords: data mining, service personalization, knowledge discovery algorithms.

1 Introduction

The time spent by people in front of computers still increases so the problem of knowledge extraction from the enormous amount of data left by web users during their interactions is a research task that has increasingly gained attention in the last years. The analysis of such data can be used to understand users preferences and behaviors in a process commonly referred to as Web Usage Mining (WUM). The extracted knowledge can be used for different goals such as service personalization, site structure simplification, web server performance improvement or even for studying the human behavior. In the past, several WUM systems have been made, some of them are presented in Sect. 2. The objective of this paper is to present a WUM system, called *ELM* (Event Logger Manager), designed and implemented at the Department of Electronics and Information Systems Warsaw University of Technology. Our

Ilona Bluemke
Institute of Computer Science, Warsaw University of Technology,
Warsaw, Poland
e-mail: `I.Bluemke@ii.pw.edu.pl`

system significantly differs from existing WUM systems. ELM is flexible and easy to integrate with any Java application. We use aspect modification, as proposed in [2], to add code responsible for registering required data. ELM is able to collect and store data from users interactions from many applications in one data base, and analyze them using knowledge discovery algorithms. ELM system contains several knowledge mining algorithms, i.e., apriori, ID3 and C4.5 taken from weka library [13], but without any difficulty other algorithms can be added. In our system a user can also decide on the kind of data aquisited and filtered. In Sect. 3 the architecture of ELM and its main modules are presented. Conclusions are given in Sect. 4.

2 Web Usage Mining Systems

The information placed in Internet is still increasing. During navigation web users also leave many records of their activity. This enormous amount of data can be a useful source of knowledge but sophisticated processes are need for the analysis of these data. Data mining algorithms can be applied to extract, understood and use knowledge from these data and all these activities are called web mining. Depending on the source of input data web mining can be divided into three types:

1. contents of internet documents are analyzed in Web Content Mining,
2. structure of internet portals are analyzed in Web Structure Mining,
3. the analysis of data left by users can be used to understand users preferences and behavior in a process commonly referred to as Web Usage Mining (WUM).

The web usage mining process as described in [11] consists of following phases:

- data acquisition,
- data preprocessing,
- pattern discovery,
- pattern analysis.

Often the results of pattern analysis are feedback to pattern discovery activity. Effective data acquisition phase is crucial for web usage mining. Data from users interactions with internet application can be stored on server side, proxy servers, client-side. Data can be stored in browser caches or in cookies at client level, and in access log files at server or proxy level. The analysis of such data can be used to understand users preferences and behavior in a Web Usage Mining (WUM) [7, 15]. The knowledge extracted can be used for different goals such as service personalization, site structure simplification, and web server performance improvement, e.g., [8, 10].

There are different WUM systems: small systems performing fixed number of analysis and there are also systems dedicated to large internet services. In the past, several WUM projects have been proposed, e.g., [1, 4, 5, 6, 9, 14]. In Analog system [14] users activity is recorded in server log files and processed to form clusters of user sessions. The online component builds active user sessions which are then classified into one of the clusters found by the offline component. The classification allows to identify pages related to the ones in the active session and to return the requested page with a list of related documents. Analog was one of the

first project of WUM. The geometrical approach used for clustering is affected by several limitations, related to scalability and to the effectiveness of the results found. Nevertheless, the architectural solution introduced was maintained in several other more recent projects. Web Usage Mining (WUM) system, called SUGGEST [1], was designed to efficiently integrate the WUM process with the ordinary web server functionalities. It can provide useful information to make easier the web user navigation and to optimize the web server performance.

Many Web usage mining systems are collaborating with an internet portal or service. The goal is to refine the service, modify its structure or make it more efficient. In this group are systems SUGGEST [1], WUM [7], WebMe [9], OLAM [6]. Data are extracted from www server's logs. So far the only system we were able to find using data from client side is Archcollect [5]. This system is registering data by modified explorer. Archcollect is not able to perform complex analysis. We have not found a system registering data on both client and server sides.

3 ELM Web Usage Mining System

In *Event Logger Manager* (ELM) system all web usage mining processes (mentioned in Sect. 2) are performed. ELM is responsible for data storage and also data analysis by some knowledge discovery algorithms. System is able to store data in local or remote data base. ELM user is able to define what events should be stored and when. Some basic types of events are predefined but user can also specify its own logical events. In ELM system data are preprocessed, data mining algorithms can be executed and its results may be observed. Without any difficulties new algorithms can be added.

3.1 ELM Architecture

In Fig. 1 the main parts of ELM system are presented. Data acquisition is performed by EventLogger and analysis by EventManager. Both modules are using eventDB, relational data base, containing stored events. EventLogger API is written to facilitate

Fig. 1 ELM architecture
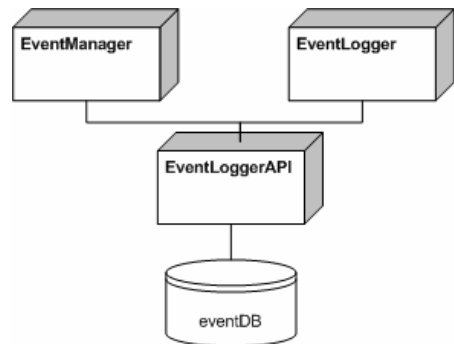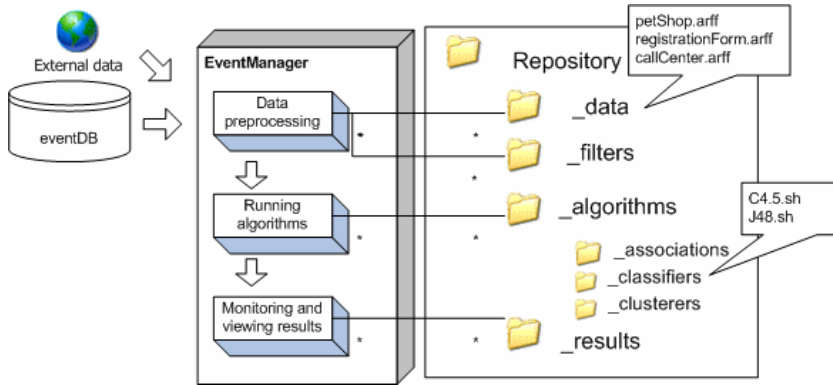
Fig. 2 ELM repository

the access to data base for data registering and data analyzing modules. In this module
logical types of events may be defined. For defined events also the parameters may
be created. EventLogger API writes or reads events with parameters, in accessing the
data base only JDBC (Java DataBase Connectivity) is used. JDBC provides methods
for quering and updating data in a relational database.

EventLogger is able to register events handled at server side. We use aspect modi-
fication proposed in [2]. An aspect, written in AspectJ [12] is responsible for storing
appropriate information describing the event. This aspect can be woven to the code
of Java application, even if only compiled code is available.

3.2 ELM Repository

ELM system is a tool which can be used by internet application owners and expert an-
alyzing users behavior. Data registered in different applications are kept in eventDB
(Fig. 2). Application EventManager reads the registered data from eventDB, presents
and transforms them, runs mining algorithms and presents the results. In Fig. 2 the
ELM repository is shown. Directory _data contains input data for mining algo-
rithms, extracted from eventDB. These data are in ARFF (Attribute Relation File
Format) [15] format. In directory _filters there are scripts processing data, e.g.,
merging data from several files. Directory _algorithms contains scripts running
mining algorithms. Currently following algorithms: apriori, ID3, and C4.5 from weka
library [13] are available. In this directory scripts running others algorithms can also
be inserted. The implementation of newly added algorithm should work on input data
in ARFF format. More implementation details are given in [3].

3.3 EventManager

EventManager presents events registered in data base, is able to browse and process
algorithm's input data, runs mining algorithms and also presents its results. The

Fig. 3 EventManager initial screen

Fig. 4 EventBrowser screen

initial screen of EventManager is shown in Fig. 3. From this screen following modules can be called:

- EventBrowser (screen shown in Fig. 4),
- ArffBrowser (screen shown in Fig. 5),
- AlgorithmRunner (screen shown in Fig. 6),
- AlgorithmMonitor (screen shown in Fig. 7).

The web mining process requires successive calls of all four modules.

EventBrowser presents events defined in EventLogger module and stored in data base eventDB (Fig. 1). User may choose interesting type of event from a pull down list and events only of this type are shown on a screen (Fig. 4). In columns the parameters describing events are shown. Common for all events parameters are e.g.: identifier, time, user name and session identifier. In EventLogger module user may define specific for an event parameters, these parameters also are displayed. In EventBrowser user is able to choose events from determined time period and observe only some events parameters. The selected from data base events can be saved in _data directory (Fig. 2) as ARFF file for future analysis.

ArffBrowser presents ARFF files saved in _data directory (Fig. 2). From a pull down list an ARFF file can be chosen. This file is displayed in a tabular form. In Fig. 5 an example of a file saved by EventBrowser is seen. Usually, files saved by EventBrowser are inadequate for a mining algorithm. ArffBrowser is also responsible for the ARFF file adaptation. The typical adaptations are e.g. removing

Fig. 5 ArffBrowser screen

Fig. 6 AlgorithmRunner screen

Fig. 7 AlgorithmMonitor screen

parameters, grouping, merging data from many rows into one row. The modified file can be stored in _data directory as a new file or replace the old one. From this module filters, performing operations on files can also be started.

The mining algorithms are started by AlgorithmRunner module. From a pull down list (Fig. 6) one of following algorithm's types may be selected: classification, clustering, discovering sequential patterns and associations. After the selection

of algorithm type a list of algorithms available in this category is presented e.g: for classification algorithms currently ID3 and C4.5 [15] can be chosen. The implementation of algorithms are taken from weka library [13]. The algorithms are kept as scripts in _algorithms directory (Fig. 2). The script running algorithm is also displayed in a text window. This window can be edited by a user, so some arguments can be changed. User should also select the ARFF file containing input data for the algorithm. After pressing the View button (Fig. 6), AlgorithmRunner checks, if the selected file contains data appropriate for the chosen type of algorithm, e.g. for classification algorithm class attribute is necessary. The Run button starts the execution of mining algorithm.

AlgorithmMonitor module monitors started algorithms and manages _results directory in ELM repository (Fig. 2). For each started algorithm appropriate subdirectory is created. The name of this directory is composed of the name of algorithm and date of execution. In this subdirectory the results of algorithm are stored in results.txt file. On the screen of AlgorithmMonitor module (Fig. 7) a list of started algorithms is presented, still running algorithms have different color than the finished ones. User may read the results or delete them.

4 Conclusions

Knowledge about users and understanding user needs is essential for may web applications. Registration of user interactions with internet application can be the basis for different analysis. Using algorithms for knowledge discovery it is possible to find many interesting trends, obtain pattern of user behavior, better understand user needs. We present a general system for web usage mining and business intelligence reporting. Our system is flexible and easy to integrate with web applications. To integrate ELM with any web application only an aspect should be written and woven. ELM is able to collect and store data from users interactions from many applications in one data base, and analyze them using knowledge discovery algorithms. A user of our system may add own algorithms or tune implemented ones, as well as decide what information will be stored, filtered and analyzed.

References

1. Baraglia, R., Palmerini, P.: Suggest: A web usage mining system. In: Proceedings of the International Conference on Information Technology: Coding and Computing, pp. 282–287 (2002)
2. Bluemke, I., Billewicz, K.: Aspects in the maintenance of compiled programs. In: Proceedings of the 3rd International Conference on Dependability of Computer Systems, pp. 253–260 (2008)
3. Bluemke, I., Chabrowska, A.: The design of a system for knowledge discovery from user interactions. Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej 6, 287–292 (2008) (in Polish)

4. Botia, J.A., Hernansaez, J.M., Gomez-Skarmeta, A.: METALA: a distributed system for web usage mining. In: Mira, J., Álvarez, J.R. (eds.) IWANN 2003. LNCS, vol. 2687, pp. 703–710. Springer, Heidelberg (2003)
5. de Castro Lima, J., et al.: Archcollect front-end: A web usage data mining knowledge acquisition mechanism focused on static or dynamic contenting applications. In: Proceedings of the International Conference on Enterprise Information Systems, vol. 4, pp. 258–262 (2004)
6. Cercone, X.H.: An OLAM framework for web usage mining and business intelligence reporting. In: Proceedings of the IEEE International Conference on Fuzzy Systems, vol. 2, pp. 950–955 (2002)
7. Chen, J., Liu, W.: Research for web usage mining model. In: Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation, p. 8 (2006)
8. Clark, L., et al.: Combining ethnographic and clickstream data to identify user web browsing strategies. Information Research 11(2), 249 (2006)
9. Lu, M., Pang, S., Wang, Y., Zhou, L.: WebME—web mining environment. In: Proceedings of the International Conference on Systems, Man and Cybernetics, vol. 7 (2002)
10. Nasraoui, O.: World wide web personalization. In: Wang, J. (ed.) Encyclopedia of Data Mining and Data Warehousing. Idea Group (2005)
11. Srivastava, J., et al.: Web usage mining: Discovery and applications of usage patterns from web data. SIGKDD Explorations 1(2), 12–23 (2000)
12. The Eclipse Foundation: AspectJ page, http://www.eclipse.org/aspectj/
13. The University of Waikato: Weka home page, http://www.cs.waikato.ac.nz/ml/weka
14. Turner, S.: Analog page, http://www.analog.cx
15. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Part III
Computational Techniques in Biosciences

Analyze of Maldi-TOF Proteomic Spectra with Usage of Mixture of Gaussian Distributions

Małgorzata Plechawska, Joanna Polańska, Andrzej Polański, Monika Pietrowska, Rafał Tarnawski, Piotr Widlak, Maciej Stobiecki, and Łukasz Marczak

Abstract. The article presents the method of processing mass spectrometry data, with detection of peaks by using the mixture of distributions. Spectra are firstly subjected to preprocessing, involving calibration, normalization, denoising and baseline correction. After that they are modeled with Gaussian distributions. Each distribution represents a single peak. Means of Gaussians describe m/z values of peaks while standard deviations represent their width. Third parameters are weights representing highs of individual peaks. The simulation presents usage of Expectation-Maximization algorithm to processing spectra with known m/z value of albumin and unknown m/z value of other compounds existing in the analyzed data sets. The developed algorithm was employed in identification of m/z values of proteins attached to the albumin with usage of the decomposition of Gaussian components. Searched m/z values were discovered with predetermined accuracy.

Keywords: maldi-tof, mass spectrometry, EM algorithm.

Małgorzata Plechawska
Computer Science Institute, Lublin University of Technology,
Nadbystrzycka 36b, 20-618 Lublin, Poland

Joanna Polańska
Institute of Automatic Control, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland

Andrzej Polański
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland

Monika Pietrowska · Rafał Tarnawski · Piotr Widlak
Department of Experimental and Clinical Radiobiology,
Comprehensive Cancer Centre Maria Sklodowska-Curie Memorial Institute,
Wybrzeże Armii Krajowej 15, 44-101 Gliwice, Poland

Maciej Stobiecki · Łukasz Marczak
Institute of Bioorganic Chemistry, Polish Academy of Sciences,
Z. Noskowskiego 12/14, 61-704 Poznań, Poland

1 Introduction

Mass spectrometry is a technique of processing tissue samples performing samples ionization. A basis of this method is the measurement of mass to charge (m/z) values, which are presented on mass spectra. Mass spectra obtained from a spectrometer need to be processed to find compounds of analyzed data and other significant information. To obtain reliable results one needs to perform preprocessing methods like baseline removal, normalization, denoising, peaks detection or alignment [16] before peaks determination and qualification.

The goal of the presented work is the decomposition of mass spectrum and the identification of m/z value of significant proteins. Proposed method consists in the decomposition peaks with usage of Gaussian components. Analyzed spectra have multiple overlapped peaks with large variation what makes them hard to solve with typical available methods. Algorithms of finding local maximum and minimum values which fulfils given criteria (like thresholds) do not always work efficiency. Additionally, presented method enable easier way of the adjusting of peaks among several spectra.

2 Related Work

The process of peaks identification may be performed with several techniques. Peaks may be identified by the apexes compared to the surrounding noise level [5]. Another method consists in the determination of the start and end of the peak on the basis of valleys on both sides of the apex. The area of the peak is calculated and compared to the minimal value of true peak area threshold [11, 18, 19]. The detection of peaks in the single spectra may be also performed by finding all the local maxima and choosing peaks higher than a noise-proportional, non-uniform threshold [9, 10] or by finding local maxima and minima in denoised spectrum or mean spectrum [2]. Also the signal-to-noise ratio may be used. It enables the detection of peaks that have the highest intensity among their peak clusters [20]. Other methods treats peaks as a continuous range of points with intensities above the ratio of a noise. There are also methods considering a shape of the peaks what helps in distinguishing peptides from other contaminants or noise [4]. Gentzel et al. [8] presented a method, which considers the resolution of the mass spectrometer. Other methods consider spectrum decomposition into a sum of constituent functions [17]. Results of peak detection process differs due to variety of preprocessing methods applied to the set of spectra. The complexity of intensities interpretation consists in high frequency noise. Additionally spectra from a MALDI-TOF spectrometer contain a variety of non-biological content [1].

3 Background

The data-set which was used in the experiment is a mass spectrometry data obtained in the study on healthy patients. Samples were taken from several patients in the

period of several days, 3–4 times a day. Obtained data presents protein called albumin which absorbs other organic compounds. As a result spectra with wide peak with right-side skewness are received. The goal of the work is the identification of m/z value of proteins attached to the albumin with usage of the decomposition of Gaussian components.

Spectra, before analyzing with EM algorithm, were preprocessed in a few steps. The operation of binning with mask of 2 allows reducing the number of data points by a half. This fastens calculations and additionally smoothes spectrum. Besides binning, there are multiple dedicated methods of smoothing and noise reduction [16]. In our simulation method implementing least squares digital polynomial filter of Savitzky and Golay [14] was used. This method occurred fast and effective and was chosen from the number of other algorithms using wavelet transformations or nonparametric smoothing with specified window size and the type of kernel. The next step is baseline correction. This process flattens the baseline and averages its to zero what improves the quality of an analyze. The estimation of the baseline is done within multiple shifted windows of defined width. Spline approximation is used to regress varying baseline to the window points. Useful techniques in case of analyzing and comparing of few spectra are normalization and interpolation. The normalization method which was used is scaling all spectra to total ion current (TIC) value or to constant noise.

4 Methods

Proposed methods are based on modeling analyzed spectrum with Gaussian Mixture Model [6]. Gaussian distributions of the mixture model represent searched peaks. Peaks describe organic compounds which are to be determined.

A mixture model is the combination of a finite number of probability distributions. Distribution, which are components of a model are usually the same type but it is not obvious. The mixture model is given with:

$$f^{\text{mix}}(x, \alpha_1, \ldots, \alpha_K, p_1, \ldots, p_K) = \sum_{k=1}^{K} \alpha_k f_k(x, p_k) , \tag{1}$$

where:

- $\alpha_1, \ldots, \alpha_K, p_1, \ldots, p_K$ – mixture parameters,
- $\alpha_1, \ldots, \alpha_K$ – weights, $\sum_{k=1}^{K} \alpha_k = 1$,
- $f_k(x, p_k)$ – density distribution functions.

Each component is given with the set of parameters which are determined by he type of distribution. Additionally each component has a weigh, which defines its contribution to the mixture. Sum of weights of all components must be one. Parameters of Gaussian distribution is mean μ_k and standard deviation σ_k.

A well known method of the estimation of unknown parameters is Expectation-Maximization (EM) algorithm [3]. It is nonlinear, iterative method composed of three steps. The first one is a calculation of initial parameters (or obtaining them

from randomization). Second and third steps (steps of Expectation and of Maximization) are performed in the loop until obtained solution achieves defined criterion. The expectation step (E) consists in calculation conditional probability of belonging of sample x_n to kth component:

$$p(k|x_n, p^{\text{old}}) = \frac{\alpha_k^{\text{old}} f_k(x_n, p^{\text{old}})}{\sum_{\kappa=1}^{K} \alpha_\kappa^{\text{old}} f_\kappa(x_n, p^{\text{old}})} \ . \tag{2}$$

The maximization step (M) is responsible for calculation of new parameters values. It is given with:

$$\mu_k^{\text{new}} = \frac{\sum_{n=1}^{N} x_n p(k|x_n, p^{\text{old}})}{\sum_{n=1}^{N} p(k|x_n, p^{\text{old}})}, \ k = 1, 2, \ldots, K,$$

$$(\sigma_k^{\text{new}})^2 = \frac{\sum_{n=1}^{N} (x_n - \mu_k^{\text{new}})^2 p(k|x_n, p^{\text{old}})}{\sum_{n=1}^{N} p(k|x_n, p^{\text{old}})} \ , \tag{3}$$

$$\alpha_k^{\text{new}} = \frac{\sum_{n=1}^{N} p(k|x_n, p^{\text{old}})}{N} \ .$$

The whole process is performed until the convergence with defined accuracy is reached. There are few different stop criteria which may be applied to EM. Some of them are based on different distance measures of obtained parameters [13] (like Chi2, Euclidean, Chebyshev). Others consists in calculation of value of likelihood function (4) and maximum likelihood rule. Maximum likelihood rule states, that the higher value of likelihood function is, the better parameters estimation can be gained. Usage of maximum likelihood rule gives a certainty of stability, because of monotonicity of likelihood function – it is ascending or, alternatively – stable:

$$L(p, x) = L(p) = f(x_1, x_2, \ldots, x_N, p) = \prod_{n=1}^{N} f(x_n, p) \ . \tag{4}$$

In our calculations standard criterion of distance between consecutive values of likelihood function was used.

Application of EM algorithm to standard Gaussian mixture model uses only data placed on one dimension (X axis) [14]. Spectrometry data are given with two vectors (of m/z values and intensities). Adjustment of EM algorithm to those data resulted in developing weighted version of this method.

Values of intensities describe the number of repeats of corresponding m/z values. Each single m/z value from X axis (x_k) should be repeated (y_k) times to obtain the single vector of parameters which can be used in EM algorithm. Therefore, to achieve vector meeting requirements of typical EM algorithm, proper m/z values need to be copied. Problem concerning such solution is that EM algorithm, which is quite computational demanding, needs to use such long, full of repeated values vector.

Instead of creation of one vector, appropriate multiplication is applied. Value of probability that sample xn belongs to kth component calculated in E step remain

stable (2). Differences may be seen in calculations of M step. New formula is given with:

$$\mu_k^{\text{new}} = \frac{\sum_{n=1}^{N} x_n y_n p(k|x_n, p^{\text{old}})}{\sum_{n=1}^{N} p(k|x_n, p^{old}) y_n}, \quad k = 1, 2, \ldots, K,$$

$$(\sigma_k^{\text{new}})^2 = \frac{\sum_{n=1}^{N} (x_n - \mu_k^{\text{new}})^2 p(k|x_n, p^{\text{old}})}{\sum_{n=1}^{N} p(k|x_n, p^{old}) y_n}, \quad (5)$$

$$\alpha_k^{\text{new}} = \frac{\sum_{n=1}^{N} p(k|x_n, p^{\text{old}}) y_n}{N}.$$

The convergence and complexity of EM algorithm depends on data size and structure. The algorithm converges relatively close to the maximum in several dozens iterations [13] (according to Fallin and Schork in less than 50 [7]). Fast convergence at the beginning of computations changes in slow 'toddling' what might significantly increase complexity especially in case of increasing the accuracy of the stopping criterion [13, 15]. EM-type algorithms are guaranteed to always increase the likelihood value, however there is also a problem of the existence of local maxima, what may cause erroneous or imprecise results (more in [3, 13]).

5 Determination of the Number of Components

Usage of EM algorithm needs to define a number of components in the mixture at the beginning of the calculations. There are several criteria which may be used to solve this problem. Criterion, which was chosen was Bayesian Informatics Criterion (BIC). It is based on likelihood function value. According to BIC, optimal number of parameters should maximize the formula:

$$-2\ln p(x|k) \approx BIC = -2\ln L + k\ln(n), \quad (6)$$

where:

- x – the observed sample,
- n – the sample size,
- k – the number of parameters to estimation,
- $p(x|k)$ – the likelihood of the observed data given the number of parameters,
- L – the maximized value of the likelihood function for the estimated model.

The determination of number of components with usage of BIC is time consuming due to the necessity of multiple repetition of calculations of EM algorithm with different number of components. Obtained results are presented in Fig. 2. Calculations were performed multiple times for each considered number of components.

As Fig. 1 shows, values of BIC criterion are growing with raising numbers of components. However BIC values begin to stabilize on the rate of 30–35 components. That is why the number of components was chosen from this range. Too large number of components in the adaptation of EM algorithm results in
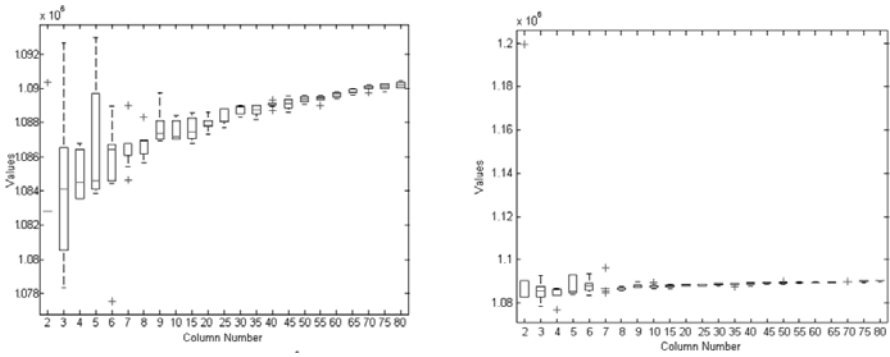
Fig. 1 BIC estimation of the number of components

formation of flat Gaussians with large standard deviation. This kind of components are of small importance in the problem of mass spectra data solving and are ignored in further process of analysis.

6 Results

Fig. 2 presents results of the analysis of five spectra of one healthy woman. Samples for spectrometric analysis were taken from the patient in the period of several days in different period of a day.
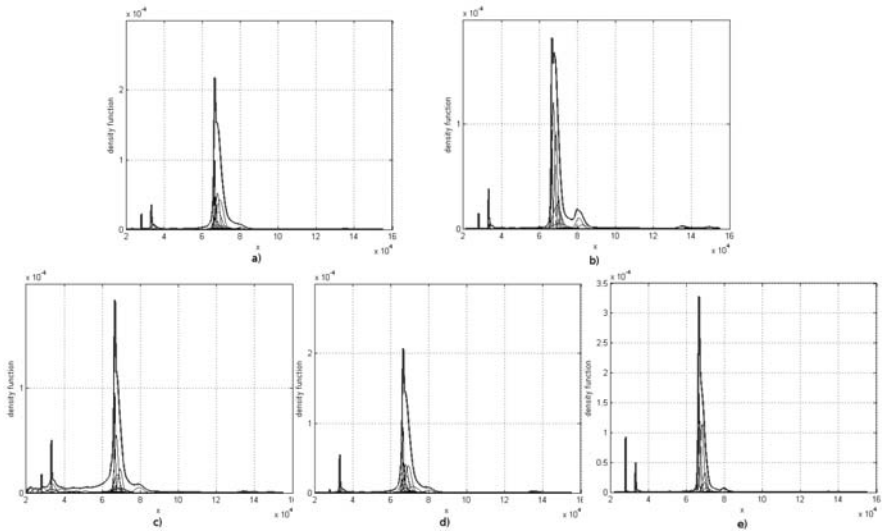
Fig. 2 Results of spectra analysis with EM alg. obtained in the time respectively: **(a)** 04.10, 8 am; **(b)** 10.10, 8 am; **(c)** 17.10, 2 pm; **(d)** 18.10, 10 am; **(e)** 25.10, 10 am

Table 1 Significant m/z values of analyzed spectra

Spectrum	Significant m/z values
a)	66611 66863.5 68001.4 68082.8 68343.4 68503.3 70827.7 80138.7
b)	66540.7 67363.2 68725.8 69297.7 69993.2 70615.6 70966.1 79607.6 82820.4
c)	66540.8 68725.8 69993.2 74573.7 74994.1 75693.3 77260.8 78127.8 79098
d)	66528.7 66754.7 67787.3 67795.3 67862.1 67935.2 70696.9 71699.8 80025.6
e)	66306.9 66590.2 67019.9 68272.2 69651.5 69731 70682.2 71259.2 72355.2 9786.2

Presented results come from EM algorithm calculations performed for five original mass spectra data. On each figure thinner lines present single calculated components whereas the bold line represents an their envelope. The hole process of analysis involve preprocessing in the following order: calibration, baseline correction, binning, noise removal, normalization. After that EM procedure was applied. Due to the fact that m/z value of the albumin is known (69366 Da), one of mixture component was set up constantly. Operation of substitution was performed on every iteration to keep correct value of likelihood function.

The analysis was done to determine m/z values of compounds, which are adjacent to the albumin. Significant values of m/z were chosen among all mixture components found by EM algorithm. Selection was done to find rather high components with small standard deviations and high weights values. Remaining components, which are flat and long-winded, constitute only background of analyzed mixture and do not have influence on the shape of the spectrum. Significant m/z values for spectra given in Fig. 2 are presented in Table 1.

7   Conclusions

This paper presents the method of proteomic spectra analysis based on Gaussian Mixture Models with constantly set the m/z value of the albumin. This method occurred to be reliable and efficient. Time of calculations of one spectrum of 80000 points is several minutes period. The disadvantage of the method is that it is slower than methods using local maxima and minima. Nevertheless computational time is acceptable and method may be applied to complex spectra with many overlapped peaks of considerable value of variance. Such spectra are hard to solved with traditional methods based on local maxima.

References

1. Baggerly, K., et al.: A comprehensive approach to the analysis of matrix-assisted laser desorption/ionization-time of flight proteomics spectra from serum samples. Proteomics 3, 1667–1672 (2005)
2. Coombes, K.: Pre-processing mass spectrometry data. In: Dubitzky, W., Granzow, M., Berrar, D. (eds.) Fundamentals of Data Mining in Genomics and Proteomics, pp. 79–99. Kluwer, Boston (2007)

3. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society 39(1), 1–38 (1977)

4. Du, P., Kibbe, W., Lin, S.: Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. Bioinformatics 22(17), 2059–2065 (2006)

5. Eidhammer, I., et al.: Computational methods for mass spectrometry proteomics. John Wiley & Sons, Incorporated, Chichester (2007)

6. Everitt, B.S., Hand, D.J.: Finite Mixture Distributions. Chapman and Hall, New York (1981)

7. Fallin, D., Schork, N.J.: Accuracy of haplotype frequency estimation for biallelic loci, via the Expectation-Maximization algorithm for unphased diploid genotype data. American Journal of Human Genetics 67(4), 947–959 (2000)

8. Gentzel, M., Kocher, T., Ponnusamy, S., Wilm, M.: Preprocessing of tandem mass spectrometric data to support automatic protein identyfication. Proteomics 3, 1597–1610 (2003)

9. Mantini, D., et al.: LIMPIC: a computational method for the separation of protein signals from noise. BMC Bionformatics 8(101) (2007)

10. Mantini, D., et al.: Independent component analysis for the extraction of reliable protein signal profiles from MALDI-TOF mass spectra. Bioinformatics 24, 63–70 (2008)

11. Morris, J., et al.: Feature extraction and quantification for mass spectrometry data in biomedical applications using the mean spectrum. Bioinformatics 21(9), 1764–1775 (2005)

12. Norris, J., et al.: Processing MALDI mass spectra to improve mass spectral direct tissue analysis. National Institutue of Health, US (2007)

13. Plechawska, M.: Comparing and similarity determining of Gaussian distributions mixtures. In: Materials of SMI Conference, Świnoujście, Poland (2008)

14. Plechawska, M.: Using mixtures of Gaussian distributions for proteomic spectra analysis. In: Proceedings of the Xth International PhD Workshop OWD. Gliwice, Poland (2008)

15. Polanska, J.: The EM algorithm and its implementation for the estimation of frequencies of SNP-haplotypes. International Journal Of Applied Mathematics And Computer Science 13(3), 419–429 (2003)

16. Polański, A., et al.: Application of the Gaussian mixture model to proteomic MALDI-ToF mass spectra. Journal of Computational Biology (2007)

17. Randolph, T., et al.: Quantifying peptide signal in MALDI-TOF mass spectrometry data. Molecular & Cellular Proteomics 4(12), 1990–1999 (2005)

18. Tibshirani, R., et al.: Sample classification from protein mass spectrometry, by 'peak probability contrasts'. Bioinformatics 20, 3034–3044 (2004)

19. Yasui, Y., et al.: A data-analytic strategy for protein biomarker discovery: profiling of high-dimensional proteomic data for cancer detection. Biostatistics 4, 449–463 (2003)

20. Zhang, S.Q., et al.: Peak detection with chemical noise removal using short-time FFT for a kind of MALDI data. Lecture Notes in Operations Research 7, 222–231 (2007)

Energy Properties of Protein Structures in the Analysis of the Human RAB5A Cellular Activity

Dariusz Mrozek, Bożena Małysiak-Mrozek, Stanisław Kozielski, and Sylwia Górczyńska-Kosiorz

Abstract. Energy features of protein structures can be used to study protein abilities to interact with other biological molecules and possibilities to take part in cellular reactions. In the paper, we present the analysis of the human RAB5A activity with the use of different energy characteristics. Energy characteristics are distributions of potential energy over amino acid chains of proteins. Conformational changes, mutations, structural deformations and other disorders are reflected in energy distributions. Therefore, energy characteristics can be used in detection and verification of such states and further analysis of their influence on protein activity.

Keywords: protein structures, cellular activity, energy characteristics.

1 Introduction

Comparative analysis of proteins allows to diagnose and to recognize many present diseases. The meaning of the process is huge, if we understand how important role proteins play in all biological reactions in living cells. They take a part in cellular reactions as substrates or enzymes (molecules that catalyze reactions) and they are also resultant products of these reactions destined to many different functions, like: energy storage, signal transmission, maintaining of a cell structure, immune

Dariusz Mrozek · Bożena Małysiak-Mrozek · Stanisław Kozielski
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail:`{Dariusz.Mrozek,Bozena.Malysiak,`
`Stanislaw.Kozielski}@polsl.pl`

Sylwia Górczyńska-Kosiorz
Clinical Department of Nephrology, Diabetology and Internal Diseases,
Medical University of Silesia,
Poniatowskiego 15, Katowice, Poland
e-mail: `sykos@wp.pl`

response, transport of small bioparticles, regulation of a cell growth and division, and many others [8, 17]. Comparative analysis of proteins is usually carried on one of two levels of protein construction – amino acid sequence (also known as primary structure) and/or spatial structure (also known as conformation or tertiary structure) determined by the location and arrangement of atoms [1, 4]. Since protein spatial structure provides much more biological information than protein amino acid sequence, the structural analysis is a very important process used by the modern biology, medicine and pharmacology. It has a fundamental meaning for detection of many physical-chemical properties of proteins and prediction of protein-drug interactions. Moreover, it plays a key role in the molecular pathology, which studies dysfunctions of biological molecules in cellular processes as a result of mutations in protein construction and conformational deformations [2, 7, 9]. In the paper, we analyze energy features of protein conformations in order to analyze the cellular activity of the human RAB5A molecule and its mutants. We follow the analysis presented in [14, 15, 25] regarding the RAB molecules. Proteins from the RAB family are members of the bigger group of GTPases – particles that have the ability to bind and hydrolyze GTP molecules. Therefore, proteins from the RAB family play an important role in intracellular reactions of living organisms. They are elements of signal pathways where they serve as molecular controllers in the switch cycle between the active form of the GTP molecule and the inactive form of the GDP [25]. In the work, we study conformational deformations of the human RAB5A through the observation of changes in potential energy distributions. On the basis of structures of the RAB5A and its mutants, we show that any structural deformation, which appears in the protein structure, is reflected in the shape of the potential energy distribution. Therefore, energy distributions help to navigate through conformational changes that can be crucial for the analysis of protein reactivity. In Sect. 3 we give a general idea of energy characteristics that we use in the analysis. A detailed study on the RAB5A activity is presented in Sect. 4.

2   Related Works

Structure-activity analysis is quite a challenging area. There is still a need to develop new tools and methods that can be used in studies of protein activity based on the macromolecular structures. Several solutions in the domain, like [12, 21, 23], make use of Molecular Interaction Potentials (MIPs) or Molecular Interaction Fields (MIFs). MIP/MIFs are results of interaction energies between the considered compounds and relevant probes and are used for the comparison of series of compounds displaying related biological behavior [21]. MIPs can be calculated with the use of popular GRID program [10]. The GRID is also used by the MIPSim software (Molecular Interaction Potentials Similarity analysis), which supports studies and comparisons of MIP distributions for series of biomolecules [5]. MIPs-based methods are frequently used to study ligand-receptor interactions, which is crucial for the pharmacology and development of new drugs. Other methods, like [11, 16, 24], use

the three-dimensional quantitative structure-activity relationships (3D-QSAR). The 3D-QSAR involves the analysis of the quantitative relationship between the biological activity of a set of compounds and their three-dimensional properties [19].

The method that we present in the paper is much simpler than methods mentioned above. Therefore, it is also less computationally expensive. It bases on similar assumptions and calculates interaction energy distributions. However, it uses force field methods and simplified potential energy representation (presented in the next section), which is sufficient to investigate and detect protein conformational switching. For a more detailed similarity analysis, we have developed the FS-EAST method presented in [18].

3 Theoretical Background

Let's consider a simple protein P built up with m amino acids (residues). The primary structure (sequence) of the protein P will have the following form: $P = (p_1, p_2, \ldots, p_m)$. The tertiary structure (spatial structure) will be symbolized by a set of N atoms A^N. The structure A^N can be also represented as a sequence: $A^N = (A_1^{n_1}, A_2^{n_2}, \ldots, A_m^{n_m})$, where each $A_i^{n_i}$ is a subgroup of atoms corresponding to the ith residue p_i of the protein P, n_i is a number of atoms in the the ith residue p_i depending on the type of the residue, and:

$$A^N = \bigcup_{i=1}^{m} A_i^{n_i} \text{ and } N = \sum_{i=1}^{m} n_i .\tag{1}$$

Locations of atoms in the structure A^N are described in the 3D space by the (x, y, z) Cartesian coordinates. In our research, we retrieve protein structures from the macromolecular structure database Protein Data Bank (PDB) [3].

For the structure A^N we calculate *energy characteristics* E^t that describe energy properties for each substructure $A_i^{n_i}$ in the amino acid chain of the protein P. Energy characteristics are calculated according the rules of molecular mechanics [13] and on the basis of Cartesian coordinates of small groups of atoms that constitute each peptide p_i.

The energy characteristic E^t (also called the *energy pattern* or *energy distribution*) is a sequence of energy points e^t:

$$E^t = (e_1^t, e_2^t, \ldots, e_m^t) \text{ and } t \in T\tag{2}$$

where single energy point e_i corresponds to a single peptide p_i and respective subgroup of atoms $A_i^{n_i}$. The T is a set of energy types related to the force field [13] used in the computation process. In our computations we usually use the Amber94 force field, which generates five main types of potential energy: bond stretching, angle bending, torsional angle, van der Waals, and electrostatic (charge-charge) [6].

The following functional form for the force field can be used to model entire molecule A^N or small molecular subsystems $A_i^{n_i}$:

$$E^T(A_i^{n_i}) = E_{BS} + E_{AB} + E_{TA} + E_{VDW} + E_{CC}, \tag{3}$$

where $E^T(A_i^{n_i})$ denotes the total potential energy [13]. There are different types of contributing energies that are calculated for substructures $A_i^{n_i}$:

- bond stretching (E_{BS}):

$$E_{BS}(A_i^{n_i}) = \sum_{j=1}^{bonds} \frac{k_j}{2}(d_j - d_j^0)^2, \tag{4}$$

  where: k_j is a bond stretching force constant, d_j is a distance between two atoms, d_j^0 is an optimal bond length;
- angle bending (E_{AB}):

$$E_{AB}(A_i^{n_i}) = \sum_{j=1}^{angles} \frac{k_j}{2}(\theta_j - \theta_j^0)^2, \tag{5}$$

  where: k_j is a bending force constant, θ_j is an actual value of the valence angle, θ_j^0 is an optimal valence angle;
- torsional angle (E_{TA}):

$$E_{TA}(A_i^{n_i}) = \sum_{j=1}^{torsions} \frac{V_j}{2}(1 + \cos(n\omega - \gamma)), \tag{6}$$

  where: V_n denotes the height of the torsional barrier, n is a periodicity, ω is the torsion angle, γ is a phase factor;
- van der Waals (E_{VDW}):

$$E_{VDW}(A_i^{n_i}) = \sum_{k=1}^{N} \sum_{j=k+1}^{N} (4\varepsilon_{kj}[(\frac{\sigma_{kj}}{r_{kj}})^{12} - (\frac{\sigma_{kj}}{r_{kj}})^6]), \tag{7}$$

  where: r_{kj} denotes the distance between atoms k and j, σ_{kj} is a collision diameter, ε_{kj} is a well depth;
- electrostatic (E_{CC}), also known as Coulomb or charge-charge:

$$E_{CC}(A_i^{n_i}) = \sum_{k=1}^{N} \sum_{j=k+1}^{N} \frac{q_k q_j}{4\pi\varepsilon_0 r_{kj}}, \tag{8}$$

  where: q_k, q_j are atomic charges, r_{kj} denotes the distance between atoms k and j, ε_0 is a dielectric constant [13].

Therefore, in our research, we have generated five different energy patterns for a single protein structure A^N. They describe protein structures in terms of bonded and

a)   b)
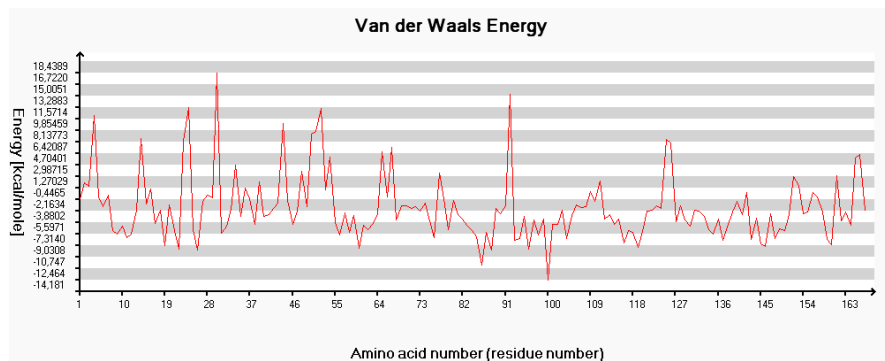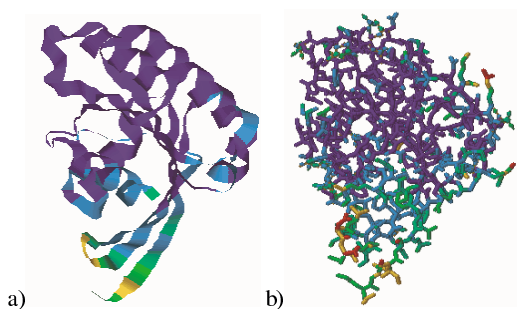
Fig. 2 Van der Waals energy characteristic for the Human RAB5A, molecule 1N6H

non-bonded interactions between atoms. In Fig. 1 we can see the spatial structure of the Human RAB5A, molecule 1N6H in the Protein Data Bank [3]. The molecule is visualized in the RasMol [22]. In Fig. 2 we can observe graphical representation of the van der Waals energy characteristic for the molecule 1N6H.

The energy characteristic presented in Fig. 2 is retrieved from the Energy Distribution Data Bank (EDB). The EDB [20] is a special repository that we have designed and developed to store so called energy profiles. Energy profiles are distributions of different potential energies over all atoms in protein structures. The EDB website (http://edb.aei.polsl.pl) also allows to generate all five energy characteristics for chosen proteins. As of Wednesday 2008-12-31 there are 32 229 energy profiles in the EDB.

4 The RAB5A Cellular Activity Observed at the Level of Energy Characteristics

Entire energy profiles and different component energy characteristics support a variety of studies on protein structures. In our research, we use them to search strong protein similarities [18]. We also examine protein structure modifications with the

use of data stored in the Energy Distribution Data Bank. In the section, we show the use of energy characteristics in the analysis of the human RAB5A conformational switching and the influence of the switching on the RAB5A cellular activity. The following molecules retrieved from the Protein Data Bank were considered during our analysis:

- Crystal Structure of Human RAB5A (molecule 1N6H),
- Crystal Structure of Human RAB5A A30P Mutant Complex with GDP (molecule 1N6I),
- Crystal Structure of Human RAB5A A30P Mutant Complex with GDP and Aluminum Fluoride (molecule 1N6K),
- Crystal Structure of Human RAB5A A30P Mutant Complex with GTP (molecule 1N6L),
- Crystal Structure of Human RAB5A A30R Mutant Complex with GPPNHP (molecule 1N6N),
- Crystal Structure of Human RAB5A A30K Mutant Complex with GPPNHP (molecule 1N6O),
- Crystal Structure of Human RAB5A A30E Mutant Complex with GPPNHP (molecule 1N6P),
- Crystal Structure of Human RAB5A A30L Mutant Complex with GPPNHP (molecule 1N6R).

The 1N6H molecule represents the crystal structure of human RAB5A GTPase domain and other molecules represent human RAB5A GTPase domain mutants in complex with different nucleotides. All molecules have similar overall folding. The mutation exists at the Ala^{30} residue of the amino acid chain in the specific region called *P-loop* (Fig. 3). The *P-loop* is a part of the RAB5A structure built up with residues ^{27}GESAVGKS^{34}. This part of the RAB5A plays a crucial role in GTP hydrolysis. Mutations in the *P-loop* have an associated influence on two other characteristic regions of the protein called *switch region I* (residues 47–65) and *switch region II* (residues 77–93). This kind of point mutation caused by the replacement of one amino acid with other amino acids can have significant biological consequences. In the active stage, the RAB5A exposes the *switch region I* and the *switch region II*. This enables the hydrolysis to occur. In the inactive stage, these two regions are not exposed enough to interact with other molecules. Therefore, conformational

Fig. 3 Structure of the human RAB5A (1N6H) molecule with characteristic regions: *P-loop*, *switch region I* (SW1) and *switch region II* (SW2)
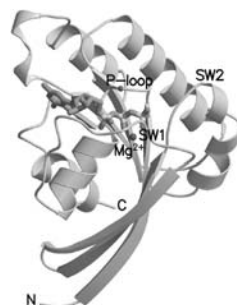
```
>1N6H:A|PDBID|CHAIN|SEQUENCE
GNKICQFKLVLLGESAVGKSSLVLRFVKGQFHEFQESTIGAAFLTQTVCLDDTTVKFEI
WDTAGQERYHSLAPMYYRGAQAAIVVYDITNEESFARAKNWVKELQRQASPNIVIALSG
NKADLANKRAVDFQEAQSYADDNSLLFMETSAKTSMNVNEIFMAIAKKLPKN

>1N6O:A|PDBID|CHAIN|SEQUENCE
GNKICQFKLVLLGESKVGKSSLVLRFVKGQFHEFQESTIGAAFLTQTVCLDDTTVKFEI
WDTAGQERYHSLAPMYYRGAQAAIVVYDITNEESFARAKNWVKELQRQASPNIVIALSG
NKADLANKRAVDFQEAQSYADDNSLLFMETSAKTSMNVNEIFMAIAKKLPKN
```

Fig. 4 Primary structure of the human RAB5A GTPase domain 1N6H (residues 15-181) and mutant 1N6O with the characteristic *P-loop*

disorders in these two regions produced by the point mutation at the Ala[30] residue have disadvantageous influence on the RAB5A activity. In the studied case, the mutation reduces catalytic abilities of the RAB5A molecule.

Amino acid sequences of the human RAB5A (1N6H) and mutant 1N6O are presented in Fig. 4. Comparison of energy characteristics allows to observe the influences of the mutation and produced conformational changes in structures of studied proteins. In Fig. 5 we can see van der Waals and electrostatic energy characteristics for referential molecule 1N6H and mutant 1N6O. Replacements in the *P-loop*

a)

b)
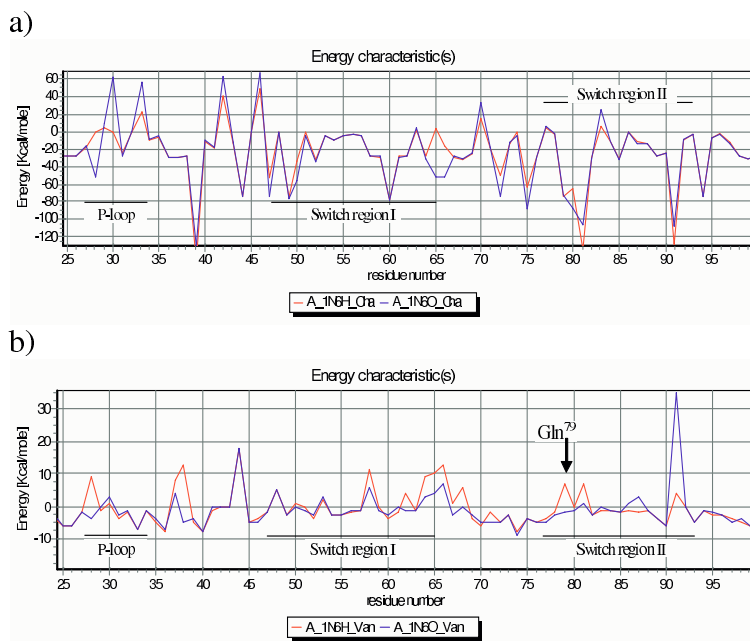
Fig. 5 Comparison of electrostatic **a** and van der Waals **b** energy characteristics for molecules 1N6H and 1N6O. Only distinctive regions are presented

Fig. 6 Separated struc-
ture of the *P-loop* (27–34
residues) for molecules: **a**
1N6H (Human RAB5A),
b 1N6O (Human RAB5A
A30K Mutant Complex with
GPPNHP)

a)   b)
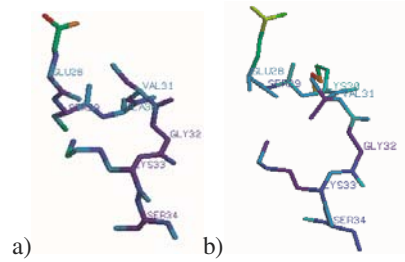
are visible in the form of changes in the distribution of energy in the *P-loop* region
of energy characteristics. Associated conformational changes in the *switch region I*
and *switch region II* are also visible as disorders in energy characteristics presented
in Fig. 5.

Further analysis of energy characteristics presented in Fig. 5 confirms studies,
observations and conclusions regarding structural mutants of the RAB5A presented
in [8]. Point mutations at the Ala[30] residue in the *P-loop* result in conformational
changes in the characteristic *switch region I* and *switch region II* and also in the
neighborhood of the Gln[79] residue, and conformational change at the Lys[33] residue.
These changes are visible in the energy characteristics as energy discrepancies.

Conformational disorders can be viewed by the observation of energy charac-
teristics and further verified with the use of molecular viewers, like RasMol [10].
Spatial structures of specific regions of the RAB5A (1N6H) and its mutant (1N6O)
are compared in Figs. 6–8. Referential structure and mutated structure of the *P-loop*
are presented in Fig. 6. Replacement occurred at the 30th residue. In the case, the
Ala amino acid was replaced by the Lys. Conformational deformations are visible
near residues Glu[28], Ser[29], and Lys[33]. In Fig. 7 we can observe conjugate deforma-
tions of the *switch region I* – visible changes near residues Ser[51], Asp[65], and Asp[66].

a)   b)
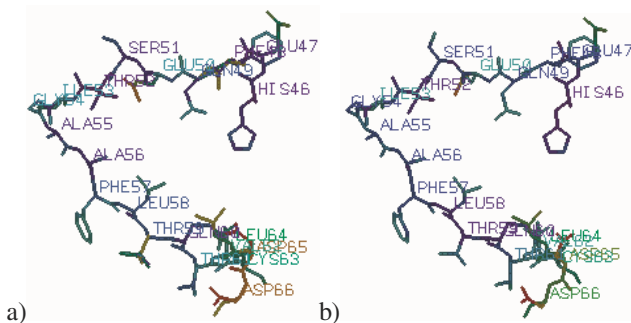
Fig. 7 Separated structure of the *switch region I* (46–65 residues) for molecules: **a** 1N6H
(Human RAB5A), **b** 1N6O (Human RAB5A A30K Mutant Complex with GPPNHP)

a)   b)
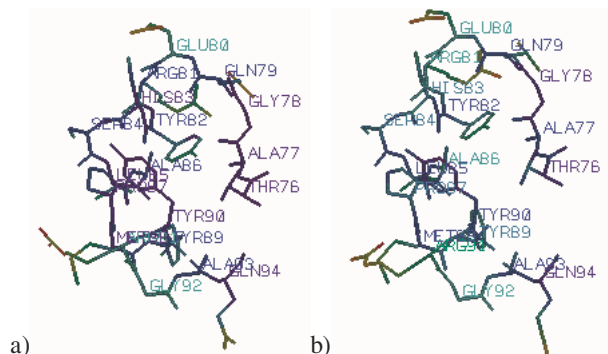
Fig. 8 Separated structure of the *switch region II* (76–94 residues) for molecules: **a** 1N6H (Human RAB5A), **b** 1N6O (Human RAB5A A30K Mutant Complex with GPPNHP)

In Fig. 8 we can see conformation deformations near residues Gln[79] and Arg[91] in the *switch region II*.

5 Concluding Remarks

Structural disorders caused by replacements of just a single residue in a protein amino acid chain can have very serious biological consequences. As a result of such a change mutant molecule may not be able to interact properly with other molecules in cellular reactions. Therefore, some biological processes can be impeded or completely disabled. However, not every conformational change causes so serious results as it was presented in previous section for the RAB5A molecule. Many deformations are effects of some natural biological processes, e.g., cellular reactions, different interactions, environmental influence, binding the molecule to another one. Moreover, some deformations caused by possible mutations can appear or influence regions that does not play a key role for any biological process, e.g., outside the active site of an enzyme. As a result, they do not influence the functioning of the whole particle. In Fig. 5 we can observe regions that indicated conformational deformations but were not directly involved in the hydrolysis process. Protein energy characteristics allow to observe such situations. They are generally supportive in the analysis of protein conformation switching and investigation of other structural deformations. Therefore, they support studies on the activity of proteins, such as human RAB5A.

Acknowledgment

References

1. Allen, J.P.: Biophysical Chemistry. Wiley-Blackwell, Chichester (2008)
2. Attwood, T.K., Parry-Smith, D.J.: Introduction to Bioinformatics. Prentice Hall, Englewood Cliffs (1999)
3. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., et al.: The protein data bank. Nucleic Acids Research 28, 235–242 (2000)
4. Branden, C., Tooze, J.: Introduction to Protein Structure. Garland (1991)
5. Caceres, M., Villa, J., Lozano, J.J., Sanz, F.: MIPSIM: Similarity analysis of molecular interaction potentials. Bioinformatics 16(6), 568–569 (2000)
6. Cornell, W.D., Cieplak, P., et al.: A second generation force field for the simulation of proteins. Nucleic acids, and organic molecules. Journal of American Chemical Society 117, 5179–5197 (1995)
7. Creighton, T.E.: Proteins: Structures and Molecular Properties, 2nd edn. Freeman, San Francisco (1993)
8. Dickerson, R.E., Geis, I.: The Structure and Action of Proteins, 2nd edn. Benjamin/Cummings, Redwood City (1981)
9. Gibas, C., Jambeck, P.: Developing Bioinformatics Computer Skills, 1st edn. O'Reilly, Sebastopol (2001)
10. Goodford, P.J.: A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. Journal of Medicinal Chemistry 28(7), 849–857 (1985)
11. Hansch, C., Verma, R.P.: 20-(S)-Camptothecin analogues as DNA Topoisomerase I Inhibitors: a QSAR study. Chem. Med. Chem. 2(12), 1807–1813 (2007)
12. Ji, H., Li, H., et al.: Computer modeling of selective regions in the active site of nitric oxide synthases: Implication for the design of isoform-selective inhibitors. Journal of Medicinal Chemistry 46(26), 5700–5711 (2003)
13. Leach, A.: Molecular Modelling: Principles and Applications, 2nd edn. Pearson Education, UK (2001)
14. Li, G., Liang, Z.: Phosphate-binding loop and Rab GTPase Function: Mutations at Ser29 and Ala30 of Rab5 lead to loss-of-function as well as gain-of-function phenotype. Biochemical Journal 355, 681–689 (2001)
15. Liang, Z., Mather, T., Li, G.: GTPase mechanism and function: New insights from systematic mutational analysis of the phosphate-binding loop residue Ala30 of Rab5. Biochemical Journal 346, 501–508 (2000)
16. Liu, H.C., Lyu, P.C., Leong, M.K., Tsai, K.C., Hsiue, G.H.: 3D-QSAR studies on PU3 analogues by comparative molecular field analysis. Bioorganic Medical Chemical Letters 14(3), 731–734 (2004)
17. Lodish, H., Berk, A., Zipursky, S.L., et al.: Molecular Cell Biology, 4th edn. W.H. Freeman and Company, New York (2001)
18. Małysiak, B., Momot, A., Kozielski, S., Mrozek, D.: On using energy signatures in protein structure similarity searching. In: Rutkowski, L., et al. (eds.) ICAISC 2008. LNCS, vol. 5097, pp. 939–950. Springer, Heidelberg (2008)
19. McNaught, A.D., Wilkinson, A.: IUPAC. Compendium of Chemical Terminology, 2nd edn. Blackwell Scientific Publications, Oxford (1997)
20. Mrozek, D., Małysiak-Mrozek, B., Kozielski, S., Świerniak, A.: The energy distribution data bank: Collecting energy features of protein molecular structures. In: Proceedings of the 9th IEEE International Conference on Bioinformatics and Bioengineering, pp. 1–6 (2009)

21. Rodrigo, J., Barbany, M., et al.: Comparison of biomolecules on the basis of molecular interaction potentials. Journal of Brazilian Chemical Society 13(6), 795–799 (2002)
22. Sayle, R., Milner-White, E.J.: RasMol: Biomolecular graphics for all. Trends in Biochemical Sciences 20(9) (1995)
23. Thorner, D.A., Wild, D.J., et al.: Similarity searching in files of three-dimensional chemical structures: Flexible field-based searching of molecular electrostatic potentials. Journal of Chemical Information and Computer Science 36(4), 900–908 (1996)
24. Trossini, G.H., Guido, R.V., et al.: Quantitative structure-activity relationships for a series of inhibitors of Cruzain from Trypanosoma Cruzi: Molecular modeling, CoMFA and CoMSIA studies. Journal of Molecular Graphical Modelling (2009) (to be published)
25. Zhu, G., Liu, J., Terzyan, S., Zhai, P., Li, G., Zhang, X.C.: High resolution crystal structures of human Rab5a and five mutants with substitutions in the catalytically important phosphate-binding loop. Journal of Biologial Chemistry 278, 2452–2460 (2003)

Fuzzy Weighted Averaging
of Biomedical Signal Using Bayesian Inference

Alina Momot

Abstract. In many types of biomedical signals there is a need of noise attenuation which, in case of systems producing repetitive patterns such as ECG acquisition systems, may be made by means of averaging signals. Traditional arithmetic averaging technique assumes the constancy of the noise power cycle-wise, however the most types of noise are not stationary. In reality the variability of noise power from cycle to cycle is observed, which constitutes a motivation for using methods of weighted averaging. This paper proposes a new weighted method incorporating Bayesian and empirical Bayesian inference and its extension using fuzzy systems with fuzzy partitioning of input data in the time domain. Performance of the presented methods is experimentally evaluated and compared with the traditional averaging by using arithmetic mean and other well known weighted averaging methods.

Keywords: biomedical signals, averaging, Bayesian inference, fuzzy partitioning, time domain.

1 Introduction

Biological systems often produce repetitive patterns. In such cases averaging in the time domain may be used to attenuate the noise. One of biomedical signals with a quasi-cyclical character is the electrocardiographic signal (ECG), which is the recording of the hearts electrical potential versus time. Three types of predominant noise that commonly contaminate the signal are baseline wander noise, electromyographic interference (from the varying electrode-skin contact impedance caused by

Alina Momot
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: `alina.momot@polsl.pl`

the electrode movement), and 50 Hz (or 60 Hz) power line interference. Noise attenuation can be achieved by various methods such as wavelet discrimination [2] or robust principal component analysis [8]. Regarding repetitive patterns of the ECG signal, averaging in the time domain may also be used to attenuate the noise, and even in some medical applications the averaging is the only method taken into account. The computing of averaged ECG beats is necessary for the aim of evaluating some clinical indexes based on the ST depression such as ST versus HR (ST/HR) diagram as well as ST/HR hysteresis, which integrates ST/HR diagram during both exercise and recovery phases of the stress test [3].

Another example of the need of the use signal averaging is the high resolution electrocardiogram, which detects very low amplitude signals in the ventricles called Late Potentials in patients with abnormal heart conditions. Due to their very low magnitudes, Late Potentials are not visible in a standard ECG. Detecting ventricular ECG signals of very low magnitudes called Ventricular Late Potentials (VLP) requires averaging a number of signals. The high resolution electrocardiogram currently requires ensemble averaging of 200–600 beats to produce a high fidelity signal estimate [10], which corresponds to the duration of the test from 3 to 8 minutes. However, instead of averaging N successive cardiac complexes (recorded with one electrode), there were conducted studies on averaging the same cardiac complex recorded at N different locations on body surface (by N different electrodes) [7].

Noise attenuation by signal averaging is also used in the case of estimating brain activity evoked by a known stimulus. The number of averages needed to obtain a reliable response ranges from a few dozen for Visual Evoked Potentials (VEP) to a few thousand for Brainstem Auditory Evoked Potentials (BAEP), which corresponds to long duration of the test and therefore the assumption of constancy of noise is not hold. In order to shorten a typical EP session several algorithms have been developed which improve the signal to noise ratio of the averaged Evoked Potential, such as applying filters to the average response [6] or applying the weighted averaging [5].

This paper presents new method for resolving of signal averaging problem which incorporates Bayesian and empirical Bayesian inference and its extension using fuzzy systems with fuzzy partitioning of input data in the time domain. There is also presented performance comparison of all described methods using a synthetic ECG signal from CTS database in presence of synthetic (Gaussian and Cauchy) or real muscle noise. It is also worth noting that the presented methods can be use not only for weighted averaging ECG signal but for any signal which is quasi-repetitive and synchronized.

The paper is divided into three sections. Section 2 describes various methods of signal averaging, from the traditional arithmetic averaging, through the empirical Bayesian weighted averaging method EBWA [11], which will be treated as the reference method in numerical experiments, to proposed new weighted averaging method. Section 3 presents results of the numerical experiments together with conclusions.

2   Signal Averaging Methods

Let us assume that in each signal cycle $y_i(j)$ is the sum of a deterministic (useful) signal $x(j)$, which is the same in all cycles, and a random noise $n_i(j)$ with zero mean and variance in the ith cycle equal to σ_i^2 . Thus,

$$y_i(j) = x(j) + n_i(j) \,, \tag{1}$$

where i is the cycle index $i \in \{1, 2, \ldots, M\}$, and the j is the sample index in the single cycle $j \in \{1, 2, \ldots, N\}$ (all cycles have the same length N). The weighted average is given by

$$v(j) = \sum_{i=1}^{M} w_i y_i(j) \,, \tag{2}$$

where w_i is a weight for ith signal cycle ($i \in \{1, 2, \ldots, M\}$) and $\mathbf{v} = [v(1), v(2), \ldots, v(N)]$ is the averaged signal.

If the noise variance is constant for all cycles and noise has Gaussian distribution, then the weights obtained using the traditional arithmetic averaging (all the weights w_i equal to M^{-1}) are optimal in the sense of minimizing the mean square error between $\mathbf{v}$ and $\mathbf{x}$. However, in reality the variability of noise power from cycle to cycle is observed, which constitutes a motivation for using methods of weighted averaging. The idea behind these methods is to reduce influence of hardly distorted cycles on resulting averaged signal (or even eliminates them).

2.1   EBWA Method

The idea of the algorithm EBWA (Empirical Bayesian Weighted Averaging) [11] is based on the assumption that the random noise $n_i(j)$ in (1) is zero-mean Gaussian with variance in the ith cycle σ_i^2 and the signal $\mathbf{x} = [x(1), x(2), \ldots, x(N)]$ has also Gaussian distribution with zero mean and covariance matrix $B = \mathrm{diag}(\eta_1^2, \eta_2^2, \ldots, \eta_N^2)$. The zero-mean assumption for the signal expresses no prior knowledge about the real distance from the signal to the baseline.

The posterior distribution over signal and the noise variance (calculated from the Bayes rule) is proportional to

$$\begin{aligned}
p(\mathbf{x}, \alpha | \mathbf{y}, \beta) \propto \prod_{i=1}^{M} \alpha_i^{\frac{N}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{N} (y_i(j) - x(j))^2 \alpha_i \right) \\
\prod_{j=1}^{N} \beta_j^{\frac{1}{2}} \exp\left(-\frac{1}{2} \sum_{j=1}^{N} (x(j))^2 \beta_j \right) ,
\end{aligned} \tag{3}$$

where $\alpha_i = \sigma_i^{-2}$ and $\beta_j = \eta_j^{-2}$. The values $\mathbf{x}$ and α which maximize the posterior distribution (3) are given by

$$\forall j \in \{1, 2, \ldots, N\} \quad x(j) = \frac{\sum_{i=1}^{M} \alpha_i y_i(j)}{\beta_j + \sum_{i=1}^{M} \alpha_i} \tag{4}$$

and

$$\forall i \in \{1, 2, \ldots, M\} \quad \alpha_i = \frac{N}{\sum_{j=1}^{N} (y_i(j) - x(j))^2}. \tag{5}$$

Since it is impossible to measure β_j directly, the iterative expectation-maximization technique can be exploited. Assuming the gamma prior for β_j with scale parameter λ and shape parameter p for all j, conditional expected value of β_j is given by

$$E(\beta_j | x(j)) = \frac{2p + 1}{(x(j))^2 + 2\lambda}. \tag{6}$$

Assuming that p is a positive integer, the estimate $\widehat{\lambda}$ of hyperparameter λ can be calculated by applying empirical method:

$$\widehat{\lambda} = \left(\frac{\Gamma(p)(2p-1)}{(2p-1)!!} 2^{p-\frac{3}{2}} \frac{\sum_{j=1}^{N} |x(j)|}{N} \right)^2, \tag{7}$$

where $(2p-1)!! = 1 \times 3 \times \cdots \times (2p-1)$. When the hyperparameter λ is calculated based on first absolute sample moment as in formula (7), the method is denoted as EBWA.1 [11].

2.2 Fuzzy Weighted Averaging

Giving assumption as described in previous section, but $\beta = \eta_1^{-2} = \eta_2^{-2} = \cdots = \eta_N^{-2}$, the posterior distribution over signal and the noise variance can be calculated from the Bayes rule explicitly and it is Gaussian distribution with mean vector $\mathbf{m}$:

$$\forall j \in \{1, 2, \ldots, N\} \quad m(j) = \frac{\sum_{i=1}^{M} \alpha_i y_i(j)}{\beta + \sum_{i=1}^{M} \alpha_i}. \tag{8}$$

The unknown parameters α_i for $i \in \{1, 2, \ldots, M\}$ and β can be estimated using method of moments [4], by equating sample moments with unobservable population

moments and then solving those equations for the quantities to be estimated, which gives

$$\forall i \in \{1,2,\ldots,M\} \quad \alpha_i = \frac{N}{\sum\limits_{j=1}^{N}(y_i(j)-x(j))^2} \tag{9}$$

and

$$\beta = \frac{N}{\sum\limits_{j=1}^{N}(x(j))^2}. \tag{10}$$

Therefore the proposed new weighted averaging algorithm can be described as follows, where ε is a preset parameter:

1. Initialize $\mathbf{v}^{(0)} \in R^N$ and set iteration index $k = 1$.
2. Calculate $\beta^{(k)}$ using (10) and $\alpha_i^{(k)}$ using (9) for $i \in \{1,2,\ldots,M\}$, assuming $\mathbf{x} = \mathbf{v}^{(k-1)}$.
3. Update the averaged signal for kth iteration $\mathbf{v}^{(k)}$ using (8), $\beta^{(k)}$ and $\alpha_i^{(k)}$, assuming $\mathbf{v}^{(k)} = \mathbf{m}$.
4. If $\|\mathbf{v}^{(k)} - \mathbf{v}^{(k-1)}\| > \varepsilon$ then $k \leftarrow k+1$ and go to 2, else stop.

It is worth noting that this method is very similar to the described earlier EBWA and the only difference is in estimated parameter β, it shows that using different estimation methods can give the same results. The new method is simplification of the EBWA and does not require determining any additional parameters like p.

This new algorithm can be extended by fuzzy partitioning of input data in the time domain. The input signal can be divided into K fuzzy sets with Gaussian membership function with location parameter equal $a^k = (k-0.5)N/K$ (for $k \in \{1,2,\ldots,K\}$) and scale parameter $b = 0.25N/K$, where N is the length of cycles. The idea of this extension is to perform K times the averaging for $k \in \{1,2,\ldots,K\}$ input data in form

$$y_i^k(j) = y_i(j) \times \exp\left(-\left(\frac{j-a^k}{b}\right)^2\right) \tag{11}$$

for $i \in \{1,2,\ldots,M\}, j \in \{1,2,\ldots,N\}$ and then sum the weighted average. Of course this fuzzy partitioning can be also used with another weighted averaging method such as EBWA method. Using such fuzzy partitioning of the data set usually leads to improve the efficiency of the methods [9], which can be also observed in the numerical examples presented below.

3 Numerical Experiments and Conclusions

In this section there will be presented performance of the described methods. In all experiments, using weighted averaging, calculations were initialized as the arithmetic means of disturbed signal cycles and the parameter ε was equal to 10^{-6}. For a
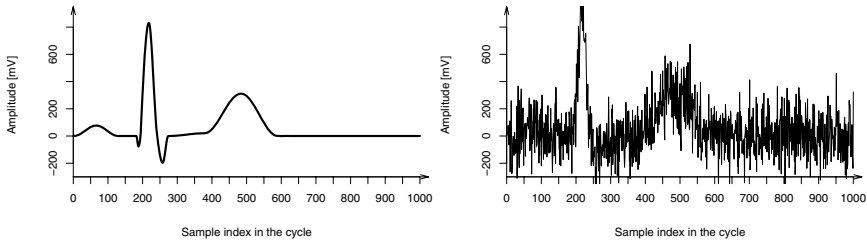
Fig. 1 The example of ECG signal and this signal with Gaussian noise

computed averaged signal the performance of tested methods was evaluated by the maximal absolute difference between the deterministic component and the averaged signal (MAX). The root mean-square error (RMSE) between the deterministic component and the averaged signal was also computed. All experiments were run in the R environment for R version 2.4.0 (www.r-project.org).

The simulated ECG signal cycles were obtained as the same deterministic component with added independent realizations of random noise. The deterministic component was ANE20000 from database CTS [1]. Figure 1 presents the ANE20000 ECG signal and this signal with Gaussian noise with standard deviation equal to sample standard deviation of the deterministic component.

A series of 100 ECG cycles was generated with the same deterministic component and zero-mean white Gaussian noise with different standard deviations with constant amplitude of noise during each cycle or real muscle noise with different amplitude as well as Cauchy noise which was treated as simulated impulse noise.

In the first case it was taken Gaussian noise. For the first, second, third and fourth 25 cycles, the noise standard deviations were respectively $0.1s$, $0.5s$, $1s$, $2s$, where s is sample standard deviation of the deterministic component. Table 1 presents the RMSE and the absolute maximal value (MAX) of residual noise for all tested methods. The best results are bolded. In the table AA denotes the traditional Arithmetic

Table 1 Results for Gaussian noise

Method	RMSE	MAX
AA	15.71243	70.41463
NBWA	2.778452	9.397265
FBWA (K=2)	2.778417	9.419718
FBWA (K=3)	2.254924	**8.496753**
FBWA (K=4)	2.390623	8.505228
FBWA (K=5)	**2.190022**	9.675443
EBWA.1	2.774010	9.356350
FEBWA.1 (K=2)	2.736333	9.104503
FEBWA.1 (K=3)	2.341914	8.655120
FEBWA.1 (K=4)	2.489946	8.637759
FEBWA.1 (K=5)	2.425370	9.721426

Table 2 Results for muscle noise

Method	RMSE	MAX
AA	13.48918	40.05002
NBWA	3.189185	9.002721
FBWA (K=2)	3.056670	7.810247
FBWA (K=3)	2.649014	8.807974
FBWA (K=4)	2.77059	9.09185
FBWA (K=5)	**2.523229**	8.915233
EBWA.1	3.175862	8.889550
FEBWA.1 (K=2)	3.004564	**7.699104**
FEBWA.1 (K=3)	2.688369	8.726101
FEBWA.1 (K=4)	2.835586	8.928537
FEBWA.1 (K=5)	2.740166	8.871437

Averaging, NBWA – New Bayesian Weighted Averaging method and EBWA.1 – Empirical Bayesian Weighted Averaging method with parameter $p = 1$. The letter F before the abbreviation denotes using fuzzy partitioning with K number of fuzzy sets. As it can be seen the fuzzy partitioning was used not only to the new Bayesian method but also to EBWA.1 method.

In the next case it was taken real muscle noise. The noise characteristic (amplitude for the first, second, third and fourth 25 cycles respectively $0.1s$, $0.5s$, $1s$, $2s$), reflects the variability of the amplitude of muscle noise characteristic where the parameter s describes signal to noise ratio. Table 2 presents results of the tests in this case.

In the last case it was taken Cauchy noise, where the Cauchy distribution function has the scale parameter equal to 0.05 times sample standard deviation of signal ANE20000. Table 3 presents results of the tests in this case.

Table 3 Results for Cauchy noise

Method	RMSE	MAX
AA	462.988	11712.980
NBWA	18.19040	90.71194
FBWA (K=2)	14.02582	64.74577
FBWA (K=3)	8.525746	38.441279
FBWA (K=4)	8.594912	43.493339
FBWA (K=5)	**6.622221**	**30.840294**
EBWA.1	17.08415	87.44787
FEBWA.1 (K=2)	12.81924	65.09986
FEBWA.1 (K=3)	9.163278	42.343444
FEBWA.1 (K=4)	8.904307	42.126111
FEBWA.1 (K=5)	7.342679	31.332357

In all performed experiments it can be seen that fuzzy partitioning of the input data gives better results (smaller errors) than the original Bayesian methods. It appears especially in case of Cauchy noise where the root mean square errors for fuzzy extensions are over two times smaller. Besides it is worth noting that although the proposed new Bayesian method gives worse results with compare to the EBWA method, using this method with fuzzy partitioning gives much better results with compare to the EBWA method with fuzzy partitioning. Moreover it can be seen that usually increasing number of fuzzy sets decreases at least root mean square errors.

References

1. International electrotechnical commission standard 60601-3-2 (1999)
2. Augustyniak, P.: Adaptive wavelet discrimination of muscular noise in the ECG. Computers in Cardiology 33, 481–484 (2006)
3. Bailon, R., Olmos, S., Serrano, P., Garcia, J., Laguna, P.: Robust measure of ST/HR hysteresis in stress test ECG recordings. Computers in Cardiology 29, 329–332 (2002)
4. Brandt, S.: Statistical and Computational Methods in Data Analysis. Springer, Heidelberg (1997)
5. Fan, Z., Wang, T.: A weighted averaging method for evoked potential based on the minimum energy principle. In: Proceedings of the IEEE EMBS Conference, vol. 13, pp. 411–412 (1991)
6. Furst, M., Blau, A.: Optimal a posteriori time domain filter for average evoked potentials. IEEE Transactions on Biomedical Engineering 38(9), 827–833 (1991)
7. Jesus, S., Rix, H.: High resolution ECG analysis by an improved signal averaging method and comparison with a beat-to-beat approach. Journal of Biomedical Engineering 10, 25–32 (1988)
8. Kotas, M.: Application of projection pursuit based robust principal component analysis to ECG enhancement. Biomedical Signal Processing and Control 1(4), 289–298 (2006)
9. Kuncheva, L.I.: Fuzzy Classifier Design. Springer, Heidelberg (2000)
10. Lander, P., Berbari, E.J.: Performance assessment of optimal filtering of the high resolution electrocardiogram. Computers in Cardiology 25, 693–695 (1994)
11. Momot, A., Momot, M., Łęski, J.: Bayesian and empirical Bayesian approach to weighted averaging of ECG signal. Bulletin of the Polish Academy of Sciences, Technical Sciences 55(4), 341–350 (2007)

Fuzzy Clustering and Gene Ontology Based Decision Rules for Identification and Description of Gene Groups

Aleksandra Gruca, Michał Kozielski, and Marek Sikora

Abstract. The paper presents results of the research verifying whether gene clustering that takes under consideration both gene expression values and similarity of GO terms improves a quality of rule-based description of the gene groups. The obtained results show that application of the Conditional Robust Fuzzy C-Medoids algorithm enables to obtain gene groups similar to the groups determined by domain experts. However, the differences observed in clustering influences a description quality of the groups. The rules determined cover more genes retaining their statistical significance. The rules induction and post-processing method presented in the paper takes under consideration, among others, a hierarchy of GO terms and a compound measure that evaluates the generated rules. The approach presented is unique, it makes possible to limit a number of rules determined considerably and to obtain rules that reflect varied biological knowledge even if they cover the same genes.

Keywords: clustering, decision rules, microarray analysis.

1 Introduction

The analysis of the data obtained in the DNA microarray experiment is a complex process involving application of many different methods including statistical analysis and data mining techniques. Such analysis usually consist of identification of differentially expressed genes, application of the algorithms grouping together genes with similar expression patterns and application of the methods for interpretation of the biological functions of the coexpressed genes.

One of the most popular tools used for annotation of the genes and gene products is Gene Ontology (GO) database which provides functional annotation of genes [2].

Aleksandra Gruca · Michał Kozielski · Marek Sikora
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail:{Aleksandra.Gruca,Michal.Kozielski,Marek.Sikora}@polsl.pl

Gene Ontology database includes hierarchical and structured vocabulary that is used to describe genes and gene products. The information in database is represented in a form of three disjoint directed acyclic graphs (DAGs) describing biological process, molecular function and cellular component. Each node of the DAG is called GO term and it is a single unit describing gene or gene product. The structure of database is hierarchical, GO terms that are close to the root describe general concepts and as a DAG is traversed from the root to the leaves, the concepts represented by GO terms are more specific.

In this paper we present a method that enables to describe gene groups by means of decision rules based on GO terms. Typical analysis that use decision rules to process results of the DNA microarray experiments involve inducing the rules with expression values in a rule conditional part [4, 12]. In our approach we use decision rules to describe the results of DNA microarray experiment. In such case, conditional part of the rule includes GO terms.

For each decision rule we compute its statistical significance and as a result of analysis we present only statistically significant rules. Since we are interested in such rules only, covering whole genes belonging to a group by means of determined rules is usually not possible. Even if the obtained rules do not describe all genes from the group, the result set includes so many rules that it is very difficult for an expert to interpret them.

Therefore, to improve the quality of rule-based gene group descriptions we propose a new method of rules evaluation and filtration. Wishing to increase a covering degree of gene groups by determined rules we propose a method of clustering the results of DNA microarray experiments which combines two sources of information: gene expression values and GO terms of that genes. We apply proposed approach to several well-known fuzzy clustering algorithms and compute decision rules for obtained clusters.

1.1 Gene Ontology

Formally, Gene Ontology is a directed acyclic graph $GO = (A, \leq)$, where A is a set of terms describing genes and gene products and $\leq$ is a binary relation on A such that genes described by a_j are a subset of genes described by a_i, denoted $a_j \leq a_i$, if and only if there exists a path $a_i, a_{i+1}, \ldots, a_{j-1}, a_j$ such that $a_{m-1} \leq a_m$ for $m = i+1, i+2, \ldots, j-1, j$ (we can find here an analogy with the inverse Hasse diagram for ordering relations). Relation $\leq$ is an order relation (reflexive, antisymmetric and transitive).

Each GO term has a level assigned which is defined in the following way: the ith level of the graph is formed by all GO terms $a \in A$ for which there exists a path $(root, a_1, \ldots, a_{i-1}, a_i)$ such that: $a_1 \leq root$, $a_m \leq a_{m-1}$ for $m = 2, 3, \ldots, i-1$ and $a_i \leq a_{i-1}$ (in other words there exists a path of length i from the root to that term).

GO terms can be used as a tool for gene annotations. Each annotation is an association between gene and the GO term describing it, so if we have a set of genes G there are some nodes in GO graph that are annotated with genes from the set G.

Considering the relation $\leq$ and the definition of GO graph, if there are two GO terms such that: $a_j \leq a_i$, $a_i, a_j \in A$, we can assume that any gene that is annotated with the GO term a_j is also annotated with the GO term a_i. In this paper we construct a GO graph based on the such assumptions. We call this graph *GO-inc*.

2 Rule Induction

Let there be given: a set G of genes, a set A of GO terms that create *GO-inc* and n gene groups $(G(1), G(2), \ldots, G(n))$. It is possible to create a decision table $DT = (G, A \cup \{d\})$, where for all $a \in A$, $a\colon U \to \{0,1\}$, and for all $g \in G$, $d(g) \in \{G(1), G(2), \ldots, G(n)\}$. In this set we search for all statistically significant rules of the following form:

$$\text{IF } a_{i1} = 1 \text{ and } a_{i2} = 1 \text{ and } \ldots \text{ and } a_{ik} = 1 \text{ THEN } d = G(l), \qquad (1)$$

where: $\{a_{i1}, a_{i2}, \ldots, a_{ik}\} \subseteq A$, $G(l) \in \{G(1), G(2), \ldots, G(n)\}$.

A rule of the form (1) should be interpreted as follows: if a gene is simultaneously described by terms occurring in a premise of the rule, then it belongs to the gene group $G(l)$.

We denote a set of rules with identical conclusions by $RUL_{G(l)}$ and call the description of the gene group $G(l)$. A set of genes which are described by terms occurring in a premise of a rule r we denote by $match(r)$. By $supp(r)$ we denote these genes belonging to $match(r)$ which also belong to the gene group occurring in the conclusion of r. For each rule $r \in RUL_{G(l)}$ the accuracy of r is given by the formula $acc(r) = supp(r)/match(r)$ and coverage of r is defined as $cov(r) = supp(r)/|G(l)|$. We use hypergeometric test to verify whether a determined rule is statistically significant.

In the field of our interests are all rules with the value p-value less or equal to a value established by a user. In worst case we have to determine $2^{|A|} - 1$ rules, what is impossible in the case of big number of considered ontological terms. Therefore, the EXPLORE [15] algorithm proposed by Stefanowski is more suitable for our aims. For our purposes the algorithm underwent a few modifications. Searching space of potential candidates for rules is made by means of a procedure that is iteratively repeated for each gene group. The main part of the algorithm generates premises with increasing number of terms, beginning from premises containing one GO term. While a rule – candidate achieves desired p-value, it is added to a result rule set and a conjunction is widen on. If, for a given premise, all GO terms were already considered, then a new GO term is selected (not selected yet) and a new rule creation begins. In order to narrow the searching space the following solutions were applied:

- After adding a term a to the rule premise ϕ, no terms lying on any path (from root to leaf on the ontology) that leads to the element a are considered. Let us notice that for any term $b \in A$ for which $b \leq a$ or $a \leq b$, the conjunction $b \wedge a \wedge \phi$ amount

to $a \wedge \phi$ or to $b \wedge \phi$. There is no point to consider conjunction $b \wedge \phi$ because it was considered during induction of a rule including GO term b.

- Assuming that currently created rule has the form $\phi \to \psi$, the term a will be added to its premise forming the rule $\phi \wedge a \to \psi$, if $acc(\phi \to \psi) < acc(a \to \psi)$. The condition limits a number of GO terms added that do not contribute to improving rule accuracy.

The rule set determined in this way may be large. Therefore, a method of rules evaluation and filtration is required. Apart from statistical significance, the quality of a rule $r \in RUL_{G(l)}$ is also important [1, 14]. To asses the rules quality we modified WS^{Yails} measure:

$$mWS^{Yails}(r) = (0.5 + 0.25acc(r))acc(r) + (0.5 - 0.25cov(r))cov(r). \qquad (2)$$

Another quality criterion of the rules determined is the number of GO terms occurring in a rule premise. By $Lenght(r)$ we denote the normalized number of GO terms occurring in the premise of r. We assume that the bigger number of GO terms occurring in the rule premise the more information is included in the rule (we remind that terms occurring in a premise do not lie on common path in ontology graph).

The last quality criterion is a level of GO terms occurring in the rule premise:

$$Depth(r) = \frac{\sum_{i=1}^{NoGOterms(r)} level(a_i)}{\sum_{i=1}^{NoGOterms(r)} max\_path(a_i)}, \qquad (3)$$

where: $level(a_i)$ is the level of a GO term a_i that occurs in the rule premise; $max\_path(a_i)$ is the longest path leading from the root to a leaf of GO that passes trough the node described by a_i, and $NoGOterms(r)$ is the number of GO terms occurring in the premise of r. From a description point of view we should prefer rules with premises including terms from as low level of the GO graph as possible.

Finally, a measure that enables to evaluate a rule quality is a product of all component measures:

$$Q(r) = mWS^{Yails}(r) \times Length(r) \times Depth(r) . \qquad (4)$$

A filtration algorithm that uses rules ranking obtained on basis of the above measure is executed in a loop. Starting from the best rule in the ranking all rules covering the same set of genes (or its subset) are candidates to be removed from the result rule set. However, before any rule is removed its similarity to the reference rule is verified. That similarity is determined by (5). If a rule is similar to the reference rule in more than 50%, it is removed from the set of determined rules, otherwise it remains in an output rule set.

$$Similarity(r_1, r_2) = \frac{UniqGOterms(r_1, r_2) + UniqGOterms(r_2, r_1)}{NoGOterms(r_1) + NoGOterms(r_2)}, \qquad (5)$$

where: *UniqGOterms*(r_i, r_j) is a number of unique GO terms occurring in the rule r_i and not occurring in the rule r_j. The GO term a from the rule r_i is recognized as the unique if it does not occur directly in the rule r_j and there is no path in GO graph that includes both term a and any term b from rule r_j premise.

3 Clustering Methods

In the work presented genes are the multi–represented data objects described by the expression values and by the annotations to Gene Ontology. In order to cluster genes considering these two sources of information a special approach is needed.

Distance of the genes described by means of numeric expression values may be calculated applying Euclidean distance or correlation coefficient [5]. Similarity of the genes described by means of GO terms encoded to the form of binary annotation table may be calculated applying the concept of *Information Content* $I(a)$ of an ontology term $a \in A$ given by the following formula:

$$I(a) = -\ln(P(a)), \tag{6}$$

where $P(a)$ is a ratio of a number of annotations to term a to a number of analysed genes.

In order to calculate the similarity $S_A(a_i, a_j)$ of the ontology terms *Information Content* of the terms and their common ancestor $I_{ca}(a_i, a_j)$ are applied in the following formula [9]:

$$S_A(a_i, a_j) = \frac{2I_{ca}(a_i, a_j)}{I(a_i) + I(a_j)}. \tag{7}$$

Next, the similarity $S_G(g_k, g_p)$ between genes g_k and g_p can be calculated according to the following formula [3]:

$$S_G(g_k, g_p) = (m_k + m_p)^{-1} \left(\sum_i \max_j (S_A(a_i, a_j)) + \sum_j \max_i (S_A(a_i, a_j)) \right), \tag{8}$$

where m_k is a number of annotations of gene g_k.

When being able to measure similarity or distance between analysed data objects, it is possible to apply one of the clustering algorithms. However, it must be the method suitable for complex multi-represented data. Thus, combinations of different fuzzy clustering algorithms were analysed.

Apart from the basic fuzzy clustering algorithm (FCM – Fuzzy C-Means), in our research we applied several modifications of the algorithm. The Conditional Fuzzy C-Means algorithm [13] is FCM based method which enables setting a condition on data objects modifying their impact on clustering process. Robust Fuzzy C-Medoids algorithm [8] enables clustering relational data (compared by means of similarity matrix) where computation of cluster prototypes is not possible. Conditional Robust Fuzzy C-Medoids algorithm is a modification of RFCMdd method enabling application of condition on data objects. Proximity-based Fuzzy C-Means

algorithm [10] is FCM based method which enables applying an expert knowledge in the form of proximity matrix to the clustering process.

Using the algorithms mentioned it is possible to propose three approaches to clustering genes described by microarray expression values and Gene Ontology annotations:

- cluster expression data by means of FCM algorithm and apply a resulting fuzzy partition matrix as a condition parameter to CRFCMdd algorithm run on ontology annotations, which is referenced further as CRFCMdd,
- cluster ontology annotations by means of RFCMdd algorithm and apply a resulting fuzzy partition matrix as a condition parameter to CFCM algorithm run on expression data, which is referenced further as CFCM,
- cluster expression data by means of PFCM algorithm applying a distance matrix calculated for ontology annotations as the proximity hints.

4 Experiments

Experiments were conducted on two freely available data sets: YEAST and HUMAN. The data set YEAST contains values of expression levels for budding yeast *Saccharomyces cerevisiae* measured in several DNA microarray experiments [5]. Our analysis was performed on 274 genes from 10 top clusters presented in the paper [5]. The data set HUMAN contains values of expression levels of human fibroblast in response to serum [7]. In the paper [7], 517 YEAST sequences were reported and divided into 10 clusters. After translation of the sequences for unique gene names and removal sequences that are duplicated or that are currently considered to be invalid, we obtained set of 368 genes. Then, each gene from YEAST and HUMAN sets were described by GO terms from Biological Process (BP) ontology. There were some genes in the HUMAN data set that had no GO term from BP ontology assigned, so we removed them from further analysis. After that step we obtained set consisting of 296 objects. To induce decision rules we created decision tables on the basis of the *GO-inc* graph for BP ontology. We used GO terms from at least second (for HUMAN set) and third (for YEAST set) ontology level and describing at least five genes from our data sets. After removing from genes description GO terms that did not fulfill this condition we had to remove two more genes from HUMAN data set. Finally we obtained two decision tables: decision table for YEAST data set consisting of 274 objects (genes) described by 244 attributes (GO terms) and decision table for HUMAN data set consisting of 294 objects described by 358 attributes.

The following parameter values were applied to the clustering algorithms analysed. All the clustering algorithms use a number of clusters to be created which was set to $c = 10$. The value of the parameter m impacting the fuzziness of the partition was set to $m = 1.1$. The quality of resulting fuzzy partition determined by each clustering algorithm was assessed on the basis of quality index presented in [11]. PFCM algorithm performs gradient optimization using additional parameter α

Table 1 YEAST results

Algorithm	[%] Average coverage	Before filtration		After filtration	
		Rules	Average p-val	Rules	Average p-val
Eisen	89.5	105306	0.00122	100	0.00080
FCM	93.0	111699	0.00182	108	0.00100
CRFCMdd	90.6	106750	0.00146	106	0.00107
CFCM	94.8	81210	0.00181	102	0.00102
PFCM	98.4	62184	0.00153	90	0.00103

Table 2 HUMAN results

Algorithm	[%] Average coverage	Before filtration		After filtration	
		Rules	Average p-val	Rules	Average p-val
Iyer	54.7	65582	0.00438	149	0.00416
FCM	58.2	74106	0.00487	107	0.00398
CRFCMdd	61.9	49780	0.00466	119	0.00463
CFCM	57.5	46076	0.00491	96	0.00495
PFCM	58.6	72720	0.00426	101	0.00366

which was set to $\alpha = 0.0001$. Clustering Gene Ontology data was performed only on Biological Process part of the ontology.

In case of YEAST data for genes similarity calculation during clusterization the correlation coefficient [5] was applied, and in case of HUMAN data the Euclidean distance was used. In the rule induction algorithm a number of terms occurring in a rule premise was limited to five, and the significance level was established on 0.01. Results of experiments on YEAST and HUMAN datasets are presented in Tables 1 and 2, respectively. Clustering results combined with expert (biological) analysis of the problem are presented in the first row and the results of clustering expression values only by means of FCM algorithm are presented in the second row of the tables.

Clustering results presented in the tables are quantitative results. Groups created by means of PFCM algorithm enable to generate the rules of the best coverage of the groups created. Average statistical significance of the rules does not differ significantly from results obtained by means of other clustering methods. A number of generated rules (after filtration) is also the smallest. However, to evaluate the quality of groups obtained exhaustively, quality analysis consisting of a full biological interpretation of the groups and the rules determined from them is needed. Such analysis is beyond the subject area of the paper. Assuming that a reference partition into groups is the partition presented by Eisen and Iyer, we may compare mutual covering of generated groups. The algorithm CRFCMdd that covers model groups to superlative degree, seems to be the best in the case of such comparison. It is also important that genes migration observed among groups is bigger for HUMAN set than for YEAST set.

The above observation confirms the fact that partition into groups of YEAST set proposed in [5] is probably the best possible partition. In case of HUMAN set and algorithm CRFCMdd, genes migrations among groups are meaningful. We also obtained significantly greater coverage. As a matter of fact, we could not determine a statistically significant rules for one of the groups from the obtained partition, however, this group contained very few objects.

Regardless of a clustering algorithm, the proposed rule induction, evaluation and filtration method that uses GO terms for groups description always enables to obtain not large set of rules (on average 10 rules per group) describing of gene groups. Such description is composed of the rules with a given (or better) statistical significance. The chosen rules induction algorithm guarantees that all possible combinations of ontological terms are considered. The applied method of rules evaluation and filtration guarantees that the rules filtered cover the same genes as input (unfiltered) rules and that induced rules are specific in the sense that they use GO terms from the lowest possible level in the ontology graph. It is important that each pair of rules has to differ in two ways: either cover different genes or differ from one another by at least 50% of terms (considering similarity of terms lying on the same path) occurring in premises. Such approach allows to obtain a description which includes various aspects of biological functions of described genes.

5 Conclusions

In this paper the proposal of considering both gene expression levels and their position in the ontology graph was presented. Several algorithms that enable to combine clustering such data representing different types of information were analyzed. Considering the similarity to the reference groups, the algorithm CRFCMdd combined with FCM turned out to be the best. In that way we obtained better gene groups coverage by statistically significant rules. In relation to the coverage and rules number better results were obtained using the PFCM algorithm, but genes migration among groups was then considerable and evaluation of partition quality requires deeper qualitative (biological) analysis.

The presented method of rules induction, evaluation and filtration appeared to be very effective. After filtration we obtained small rule sets having average statistical significance better than for the unfiltered rule set.

It is worth highlighting that the rules are generated only for description purposes. Due to the structure of the rules (only the descriptors corresponding to the terms having gene annotations were considered) a part of the rules determined though statistically significant is approximate and therefore genes classification by means of the rules determined could be incorrect in many cases.

Future research will concentrate on determining the rules including descriptors referencing GO terms which does not describe the analysed gene. In that case we will be interested in occurrence of this type of descriptors on the highest level of the ontology. The appropriately modified version of LEM algorithm [6] will be implemented in order to accelerate the calculations. The rules obtained by LEM

and EXPLORE algorithms after filtration will be compared. Considering clustering methods our research will focus on defining other than *Information Content* ([6]) measure of gene similarity.

References

1. An, A., Cercone, N.: Rule quality measures for rule induction systems description and evaluation. Computational Intelligence 17, 409–424 (2001)
2. Ashburner, M., Ball, C.A., Blake, J.A., et al.: Gene ontology: tool for the unification of biology. Nature Genetics 25, 25–29 (2000)
3. Azuaje, F., Wang, H., Bodenreider, O.: Ontology-driven similarity approaches to supporting gene functional assessment. In: Proceedings of the 18th Annual Bio-Ontologies Meeting, Michigan, US (2005)
4. Carmona-Sayez, P., Chagoyen, M., Rodriguez, A., Trelles, O., Carazo, J.M., Pascual-Montano, A.: Integerated analysis of gene expression by association rules discovery. BMC Bioinformatics 7(1), 54 (2006)
5. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Science 95, 14, 863–14, 868 (1998)
6. Grzymala-Busse, J.W., Ziarko, W.: Data mining based on rough sets. In: Wang, J. (ed.) Data Mining Opportunities and Challenges, pp. 142–173. IGI Publishing, Hershey (2003)
7. Iyer, V.R., Eisen, M.B., Ross, D.T., Schuler, G., Moore, T., Lee, J.C., Trent, J.M., Staudt, L.M., Hudson, J., Boguski, M.S., Lashkari, D., Shalon, D., Botstein, D., Brown, P.O.: The transcriptional program in the response of human fibroblasts to serum. Science 283, 83–87 (1999)
8. Krishnapuram, R., Joshi, A., Nasraoui, O., Yi, L.: Low-complexity fuzzy relational clustering algorithms for web mining. IEEE Transactions on Fuzzy Systems 9(4), 595–607 (2001)
9. Kustra, R., Zagdański, A.: Incorporating gene ontology in clustering gene expression data. In: Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems (2006)
10. Loia, V., Pedrycz, W., Senatore, S.: P-FCM: a proximity-based fuzzy clustering for user-centered web applications. International Journal of Approximate Reasoning 34, 121–144 (2003)
11. Łęski, J., Czogała, E.: A new artificial neural network based fuzzy inference system with moving consequents in if-then rules and its applications. Fuzzy Sets and Systems 108(3), 289–297 (1999)
12. Midelfart, H.: Supervised learning in gene ontology Part I: A rough set framework. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets IV. LNCS, vol. 3700, pp. 69–97. Springer, Heidelberg (2005)
13. Pedrycz, W.: Conditional fuzzy c-means. Pattern Recognition Letters 17, 625–631 (1996)
14. Sikora, M.: Rule quality measures in creation and reduction of data role models. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS, vol. 4259, pp. 716–725. Springer, Heidelberg (2006)
15. Stefanowski, J., Vanderpooten, D.: Induction of decision rules in classification and discovery-oriented perspectives. International Journal on Intelligent Systems 16(1), 13–27 (2001)

Estimation of the Number of Primordial Genes in a Compartment Model of RNA World

Dariusz Myszor and Krzysztof A. Cyran

Abstract. The Origin of life has been studied by researchers for many years. During this time a lot of computer models of early life were created and have given scientists a better view of this immemorial time. One of the simple models of early life proclaims that primitive genes (molecules) were enclosed in compartments (packages) which were submerged in the primordial broth. U. Niesert, D. Harnasch, and C. Bresch in the article 'Origins of Life Between Scylla and Charybdis' explained the basics of the model and predicted that there can be only 3 unlinked types of genes in a package. One of the important factor in the compartment model is NORM (Number of replicated molecules between two packages fission). We wanted to check whether NORM variation caused by environment changes, that certainly took place at this time, has an influence on the maximum number of unlinked types of genes in a package. Results of our researches, based on computer simulations, indicate that NORM variation has such an influence.

Keywords: RNA world, compartment model, primordial genes, computer simulations.

1 Introduction

First traces of life on the Earth are 3.5 billion years old [10], but we assume that life began 0.5 billion years earlier. At the beginning, life on our planet was completely

Dariusz Myszor · Krzysztof A. Cyran
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail:{dariusz.myszor,krzysztof.cyran}@polsl.pl

different from the present one. There is a lot of unanswered questions related to this period of time. We do not know where life started [2, 13], on the Earth's surface, in deep-sea vents [10] or maybe it came in meteorites (panspermia [4]), which were common guests on our young planet in this period. We do not know what was the first as well: metabolism or replication [11], or maybe metabolism and replications emerged in the same moment. We speculate whether the RNA world (world in which RNA molecules are the only available form of life and which lead to DNA creation), was the first one or it was preceded by other forms of life, such as the PNA world [12]. It is London at a brick that life exists so there must have been some beginning.

A vast amount of experiments pertaining to life's beginning are carried out by computers. Constant rise in computational power of these devices let us create and analyse more and more sophisticated models and elaborate the old ones.

In this article we are interested in a compartment model of life's beginning. The compartment model was created as an alternative to the hypercycle. In the hypercycle model every gene is responsible for encoding polypeptide supporting replication of the next gene in a cycle [3]. Since the inception of this theory, researchers have been arguing about stability and possibility of survival of such units. The package model is much simpler. Exact rules of the model are presented in [9]. One crucial gene, called the replicase [7] is responsible for replication of all genes in a protocell. At the beginning, this primitive 'replicase' could not achieve high fidelity level because of RNA strand length constraints. Those limitations in the length are being caused by high mutation level, which is caused by low fidelity of replication before the replicase had emerged. However, with time, and paradoxically thanks to series of luck mutations, fidelity of replicase should improve.

In 1987, U. Niesert used Univac 1100 to investigate this model properties [8]. In particular, she sought to determine the maximal amount of different types of genes in a package (MDTOG) under different mutation rates and NORM. Computer time was expensive, so there was a necessity to limit the amount of simulations. Now we have access to, computers with 4 core processors inside. We use ten devices with Intel Core 2 Quad 2.8 GHz. These powerful machines let us conduct enough simulations to formulate the problem in the language of statistics.

2   Model Description

In the package model, primordial genes are enclosed in primitive compartments – proto cells. A set of proto cells is denoted as population. Compartments contain many different types of genes. In one proto cell there might be many copies of the same type of gene. In order to survive the package must have at least one representative of every type of gene. All genes have equal replication rates. Once in a while a package is split into two progeny packages. This process is being called as package fission. During the fission, genes from the parent package are distributed randomly between progeny packages. If a progeny package does not have representatives of all gene types, it is removed. Genes are replicated between package fissions.

Number of replicated genes between package fission is denoted as NORM. It is difficult to imagine that NORM is constant in such a primitive cell, that is why NORM is varying in our simulations. Genes are replicated with some fidelity, during replication an error can take place. We distinguish two types of mutation:

- Parasite mutation – leads to disability of gene functionality. New gene is not harmful for the package; however it might be replicated so it reduces the amount of healthy gene replicas in a package. Parasite can never become a functional gene again.
- Lethal mutation – leads to creation of a gene with disabled functionality which causes instant death of the proto cell or it has a higher replication rate than other genes in a package (it will eliminate descendants of the package in a few generations).

We assume that there can be only harmful mutations, we do not analyse mutations that can lead to improvement in gene functionality. Lengths of genes in the model are modelled by the modification of the value of mutation rate per gene. The package can become a victim of a harmful event which leads to its death. Such an event is called an accident and it is determined instantly after package creation.

Above-mentioned processes are characterised by the following model parameters: parasite mutation rate (PMR), lethal mutation rate (LMR) and accident rate (AR).

3 Simulation

We repeated Niesert simulations with constant NORM and different levels of mutations. We also repeated simulations with NORM variation on the package level. Later we added a new type of NORM variation that should reflect changes in the environment. At the end we conducted simulations which connected NORM variation on the package level with NORM variation on the environmental level.

As a scenario, we denote simulation with given mutation rate and NORM variation. For every scenario presented in this article, we created 100 unlinked histories. Every history is simulated independently from the others. For every history we created 1000 consecutive generations. Size of every generation was constant and was set to 25 packages. In foster conditions the amount of packages can raise exponentially, which can consume all computer memory. That is why we set a limit for 25 packages. If there are more packages, after creation of the new generation, we have to reduce the number of packages in the generation. For every package we compute the prospective value proposed by Niesert and the weakest packages are removed. We assume that a package is viable only if it possesses at least one copy of every type of gene and it does not contain any lethal genes. We also assume that a history succeeds if the last generation contains at least one viable package. In order to succeed (to be approved) the scenario must have at least five successful histories. Five might seem small and negligible but in order to create the variety of nowadays life, one such a history was enough.
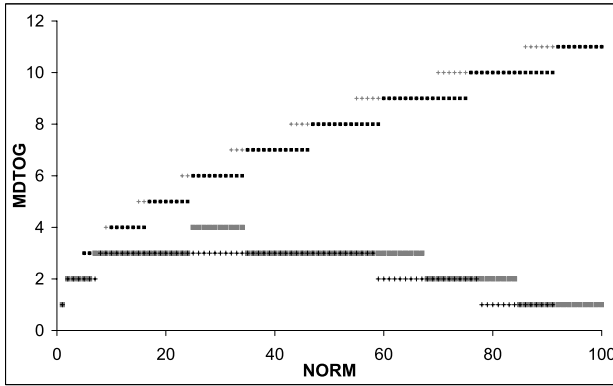
Fig. 1 MDTOG (maximal number of different types of genes in a package) as a function of NORM. Black dots and gray crosses represent turned off mutations and accidents. For gray dots and black cross PMR: 0.1, LMR: 0.01, AR: 0.01. In order to succeed, a scenario needs 5 (gray dots and gray crosses) or 95 (black dots and black crosses) successful histories

Prospective coefficient proposed by Niesert:

$$V = k^k \left(\prod_{i=1}^{k} \frac{n_i}{N} \right) \left(\frac{N}{T} \right)^k \log_2(N - k + 2),\tag{1}$$

where:

- k – the number of different genes,
- n_i – the number of copies of ith gene,
- n_p – the number of parasites in a package,
- $N = \sum_{i=1}^{k} n_i$,
- $T = N + n_p$.

Differences between the outcomes when 5 and 95 successful histories are required to approve a scenario might seem small for the case with turned off mutations, accident rate set to 0 and fixed NORM, but when we turn on mutations and the accident rate, the difference is better visible (Fig. 1).

3.1 NORM Variation

We simulated following types of variations – at the package level, at the generation level, both at the package and at the generation level. Variation at the package level was proposed by Niesert. According to the reference [8] the influence of such a variation should be low. We simulated variation with distribution different than the one used by Niesert. We used a normal distribution with mean set to the Base NORM and standard deviation equal to 15 or 30. Our results (Fig. 2) showed that there are small differences when mutations and accidents are turned off. More significant differences can be observed when mutations rates and accident rate are different
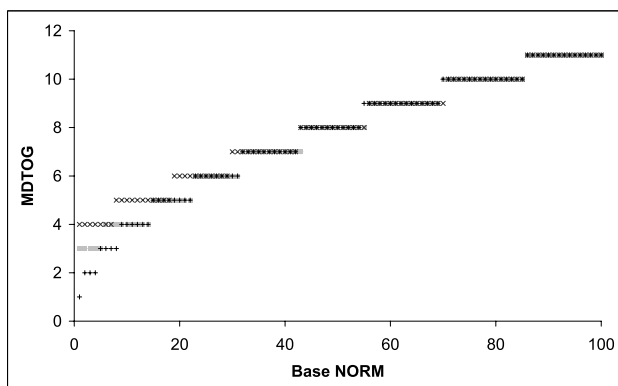
Fig. 2 MDTOG (maximal number of different types of genes in a package) as a function of Base NORM. Package NORM is ON. NORM has normal distribution, mean set to the Base NORM, standard deviation set to 15 (gray dots) and 30 (gray x), for crosses the variance is off. PMR: 0; LMR: 0; AR: 0
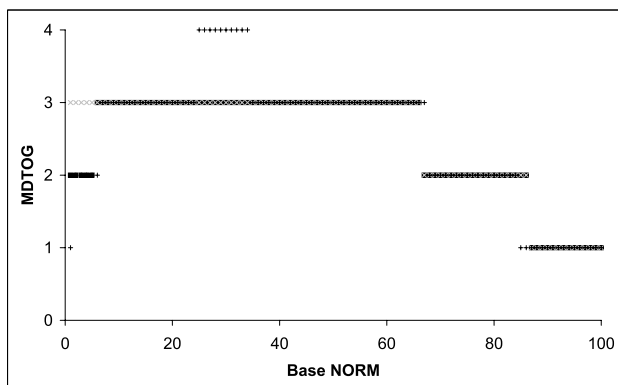
Fig. 3 MDTOG (maximal number of different types of genes in a package) as a function of Base NORM. Package NORM is ON. NORM has normal distribution, mean set to the Base NORM, standard deviation set to 15 (black dots) and 30 (gray x), for crosses the variance is off. PMR: 0.1; LMR: 0.01; AR: 0.01

from zero (Fig. 3). NORM variation at the package level might reduce the MDTOG. Greater variance leads to a greater reduction.

Variance at the generation level is a new feature in the model. We want to check whether changing environmental conditions represented by changes in the amount of replicated genes between generations, might have influence on the MDTOG. We used a normal distribution for NORM variance; there were two cases: a) mean was set to the Base NORM and standard deviation was fixed, set to a defined value (Figs. 4, 5), b) mean was set to the Base NORM and standard deviation depended
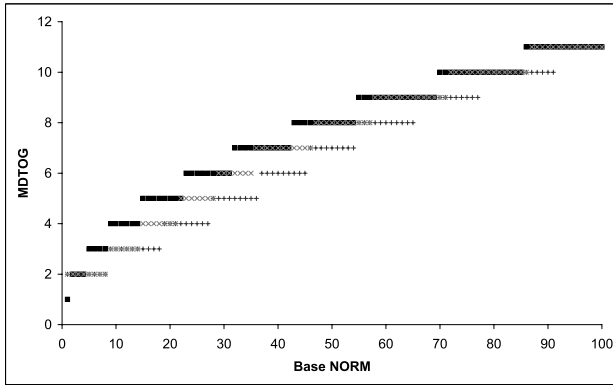
Fig. 4 MDTOG (maximal number of different types of genes in a package) as a function of Base NORM. Environmental NORM is ON. NORM has normal distribution, mean set to the Base NORM, standard deviation set to 15 (gray x) and 30 (gray crosses), for black dots NORM variance is off. PMR: 0; LMR: 0; AR: 0
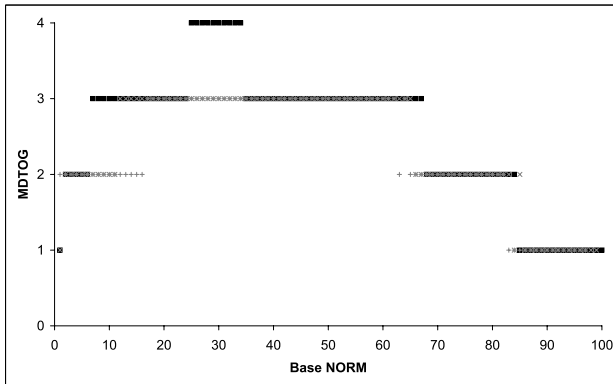
Fig. 5 MDTOG (maximal number of different types of genes in a package) as a function of Base NORM. Environmental NORM is ON. NORM has normal distribution, mean set to the Base NORM, standard deviation set to 15 (gray x) and 30 (gray crosses), for black dots NORM variance is off. PMR: 0.1; LMR: 0.01; AR: 0.01

on the Base NORM value and a constant value, standard deviation = (Base NORM value) / (constant value) (Figs. 6, 7).

Simulations with both variances (at the package level and at the generation level) turned on, are also novelty in the system. First we create environmental NORM, based on the environmental variance and then we use this value to create package NORM. We used normal distribution for both variances and conducted simulations for two cases: a) fixed standard deviation for environmental and package variances. Mean of environmental variance was set to the Base NORM and mean of package variance was set to the Environmental NORM (Figs. 8, 9) b) standard deviation
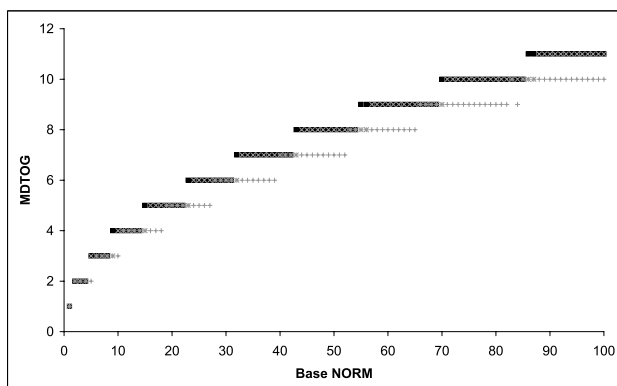
Fig. 6 MDTOG (maximal number of different types of genes in a package) as a function of Base NORM. Environmental NORM is ON. NORM has normal distribution, mean set to the Base NORM, standard deviation depends on the Base NORM; (gray x) environmental NORM standard deviation = (Base Norm)/5; (gray crosses) environmental NORM standard deviation = (Base Norm)/2. For black dots NORM variance is off. PMR: 0; LMR: 0; AR: 0
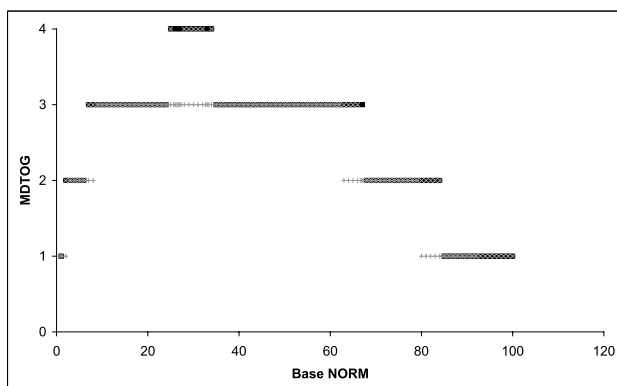
Fig. 7 MDTOG (maximal number of different types of genes in a package) as a function of Base NORM. Environmental NORM is ON. NORM has normal distribution, mean set to the Base NORM, standard deviation depends on the Base NORM; (gray x) environmental NORM standard deviation = (Base Norm)/5, (gray crosses) environmental NORM standard deviation = (Base Norm)/2. For black dots NORM variance is off. PMR: 0.1; LMR: 0.01; AR: 0.01

of environmental variance was based on the Base NORM divided by a constant and was equal to (Base NORM)/(constant value), mean was set to the Base NORM. Standard deviation of the package variance was based on the Environmental NORM divided by a constant and was equal to (Environmental NORM)/(constant value), mean was set to the Environmental NORM (Figs. 10, 11). When both variances have influence on the NORM value, number of MDTOG is lower than in scenario with variances turned off but higher than in scenario with only environmental NORM
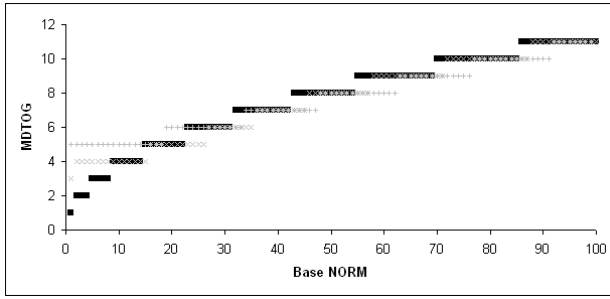
Fig. 8 MDTOG (maximal number of different types of genes in a package) as a function of Base NORM. Environmental NORM and package NORM are ON. Environmental and package NORM standard deviation set to 15 (gray x) and 30 (gray crosses), for black dots NORM variance is off. PMR: 0; LMR: 0; AR: 0
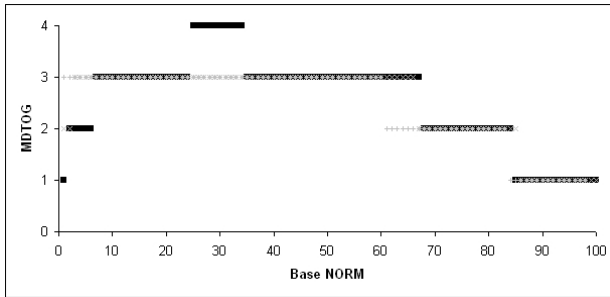
Fig. 9 MDTOG (maximal number of different types of genes in a package) as a function of Base NORM. Environmental NORM and package NORM are ON. Environmental and package NORM standard deviation set to 15 (gray x) and 30 (gray crosses), for black dots NORM variance is off. PMR: 0.1; LMR: 0.01; AR: 0.01

turned on. Variance on the package level is mitigating influence of the environmental variance (Fig. 12).

Graphs imply that environmental impact is significant, especially for a fixed standard deviation. Variance on the generation level reduces the MDTOG.

4   Results

We verified whether packages could persist without the replicase. We created series of simulations with NORM variance at the package level turned on (standard deviation was set to 15 or 30). Our experiment showed that without replicase, when the best PMR is estimated at 0.01 per nucleotide (LMR and AR set to 0), the compartment could have at most two MDTOG, each 50 nucleotides long (mutation rate 0.39 per gene) or one type of gene containing 100 nucleotides (mutation rate 0.63 per gene). For a greater mutation rate – 0.02 per nucleotide, there might be at

Fig. 10 MDTOG (maximal number of different types of genes in a package) as a function of Base NORM. Environmental NORM and package NORM are ON. Environmental NORM has normal distribution, mean set to the Base NORM, standard deviation depends on the Base NORM; package NORM has normal distribution, mean set to the Environmental NORM, standard deviation depends on the Environmental NORM; (gray crosses) environmental NORM standard deviation = (Base Norm)/5 and package NORM standard deviation = (Environmental NORM)/5; (gray x) environmental NORM standard deviation = (Base Norm)/2 and package NORM standard deviation = (Environmental NORM)/2. For black dots NORM variance is off. PMR: 0; LMR: 0; AR: 0

Fig. 11 MDTOG (maximal number of different types of genes in a package) as a function of Base NORM. Environmental NORM and package NORM are ON. Environmental NORM has normal distribution, mean set to the Base NORM, standard deviation depends on the Base NORM; package NORM has normal distribution, mean set to the Environmental NORM, standard deviation depends on the Environmental NORM; (gray crosses) environmental NORM standard deviation = (Base Norm)/5 and package NORM standard deviation = (Environmental NORM)/5; (gray x) environmental NORM standard deviation = (Base Norm)/2 and package NORM standard deviation = (Environmental NORM)/2. For black dots NORM variance is off. PMR: 0.1; LMR: 0.01; AR: 0.01

most one type of gene in a package, even if it had only 50 nucleotides. Our experiment showed that there is a limit in the length of a single gene and for PMR 0.01 per nucleotide this limit is close to 500 nucleotides (PMR 0.99 per gene, LMR: 0, AR: 0). For more realistic conditions with LMR set to 0.005 and AR set to 0.01, the maximal length of the gene is 200 nucleotides. Such an amount of information in the
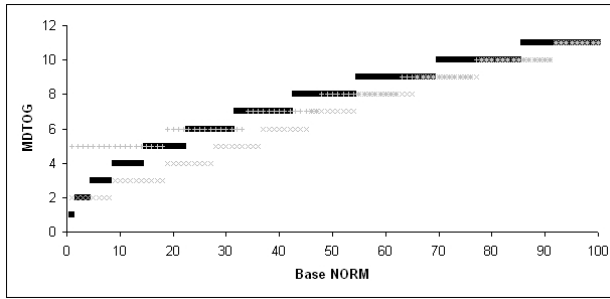
Fig. 12 Comparison of the influence of different variance types on MDTOG (maximal number of different types of genes in a package). Environmental NORM and package NORM variance on – gray crosses. Only environmental NORM variation on – gray x. Standard deviation set to 30. NORM variance off – black dots. PMR: 0; LMR: 0; AR: 0

package is similar to the amount of information contained in the single strand model. Those results are consistent with Cyran's researches concerning complexity threshold in the single strand model [1] being the modified version of Demetrius/Kimmel branching process model [5]. This single strand model describes an early stage of the RNA-world, directly following the stage of short oligonucleotides (of the length not exceeding 30 units) whose hypothetical evolution is presented by Ma [6]. The compartment model we consider in the paper describes the next stage at which the replicase ribosome is able to replicate some number of different genes. Cyran's research showed that changing conditions (different values of the accident rate, caused by the break of phosphodiester bonds in RNA molecules) had influence on the values of maximal amount of nucleotides in RNA – strand; these values were varying between 170 and 500 genes.

That is why we assumed that one gene in a package is the replicase. It can increase replication fidelity tenfold so mutation rate in the best case might be equal to 0.001 per nucleotide. If every gene in a package has 100 nucleotides, the mutation rate per gene is close to 0.1 and for 200 nucleotides close to 0.2. We conducted series of simulations for those values with environmental variation on/off and different LMR and AR. For typical values of LMR: 0.01 and AR: 0.01, when PMR was set to 0.1, MDTOG was equal to 4 for NORM variance off, and 3 for NORM variance on. When PMR was set to 0.2, the results are not so obvious: MDTOG is equal to 2 for environmental variance off, and 2 for environmental variance on. However, a package might contain 2 MDTOG within a narrower NORM range.

5   Discussion and Conclusions

We used computer simulations to demonstrate that the modified gene package model could not exist without simple replicase. When this type of gene is not present in a model, parasite mutation rate is too high and we end up with single strand model.

We confirmed the results obtained by Cyran about the maximal length of RNA – strand in the single strand model. Our results showed that in presence of replicase, without NORM variation, there might be even 4 MDTOG (100 nucleotides each) in a package for semi-optimistic mutation rates (PMR 0.001 per nucleotide, LMR 0.01, AR 0.01). However, NORM variance leads to limitation in the MDTOG. When one type of the NORM variation is on, there might be at most 3 MDTOG. Higher variance leads to greater reduction in MDTOG. It seems that environmental variance has a greater impact on the MDTOG than variance on the package level. Interestingly when both variances are on, variance on the package level seems to mitigate influence of the variance on the environmental level.

References

1. Cyran, K.A.: Information amount threshold in self-replicating RNA-protospecies: branching processes approach. International Journal of Mathematics and Computers in Simulation 3(1), 20–29 (2009)
2. Edwards, M.R.: From a soup or a seed? Pyritic metabolic complexes in the origin of life. Trends in Ecological Evolution 13, 179–181 (1998)
3. Eigen, M., Schuster, P.: The hypercycle – a principle of natural self-organization. Naturwissenschaften 64(11), 541–565 (1977)
4. Hoyle, F., Wickramasinghe, N.C.: Astronomical Origins of Life – Steps Towards Panspermia. Kluwer Academic Publishers, Dordrecht (1999)
5. Kimmel, M., Axelrod, D.: Branching Processes in Biology. Springer, New York (2002)
6. Ma, W., Yu, C., Zhang, W.: Monte Carlo simulation of early molecular evolution in the RNA world. BioSystems 90, 28–39 (2007)
7. McGinness, K.E., Joyce, G.F.: In search of replicase rybozyme – review. Chemical Biology 10, 5–14 (2003)
8. Niesert, U.: How many genes to start with? A computer simulation about the origin of life. Origins of Life Evolution of Biosphere 17(2), 155–169 (1987)
9. Niesert, U., Harnasch, D., Bresch, C.: Origin of life – between scylla and charybdis. Journal of Molecular Evolution 17(6), 348–353 (1981)
10. Orgel, L.E.: The origin of life – a review of facts and speculations. Trends in Biochemical Sciences 23(12), 491–495 (1998)
11. Pross, A.: Causation and the origin of life. Metabolism or replication first? Origins of Life and Evolution of the Biosphere 34(3), 307–321 (2004)
12. Schmidt, J.G., Christensen, L., Nielsen, P.E., Orgel, L.E.: Information transfer from DNA to peptide nucleic acids by template-directed syntheses. Nucleic Acids Research 25(23), 4792–4796 (1997)
13. Trevors, J.T.: Why on Earth: Self-assembly of the first bacterial cell to abundant and diverse bacterial species. World Journal of Microbiology and Biotechnology 15(3), 297–304 (1999)

Quasi Dominance Rough Set Approach in Testing for Traces of Natural Selection at Molecular Level

Krzysztof A. Cyran

Abstract. Testing for natural selection operating at molecular level has become one of the important issues in contemporary bioinformatics. In the paper the novel methodology called quasi dominance rough set approach (QDRSA) is proposed and applied for testing of balancing selection in four genes involved in human familial cancer. QDRSA can be considered as a hybrid of classical rough set approach (CRSA) and dominance rough set approach (DRSA). The advantages of QDRSA over CRSA and DRSA are illustrated for certain class of problems together with limitations of proposed methodology for other types of problems where CRSA or DRSA are better choice. The analysis of the reasons why QDRSA can produce decision algorithms yielding smaller error rates than DRSA is performed on the real world example, what shows that superiority of QDRSA in certain types of applications is of practical value.

Keywords: DRSA, CRSA, natural selection.

1 Introduction

Since the time of Kimura's famous book [7], the search for signatures of natural selection operating at the molecular level has become more and more important. It is so because neutral theory of evolution at molecular level does not deny the existence of selection observed at that level. It only states that the majority of observed genetic variation is caused by random fluctuation of allele frequencies in finite populations (effect of genetic drift) and by selectively neutral mutations.

If majority of mutations have been claimed to be neutral, then the next step should be to search for those which are not neutral. Indeed, several statistical tests, called

Krzysztof A. Cyran
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: `krzysztof.cyran@polsl.pl`

neutrality tests, have been developed and the neutral theory of evolution has been used as a null hypothesis for them. A statistically significant departure from this model can be therefore treated as a signature of natural selection operating in a gene under consideration.

Unfortunately, other reasons for departure from the neutral model are also possible and they also account for statistically significant signals in neutrality tests. These reasons include expansion of the population and geographical substructure of it with limited migration. Also recombination accounts for incorrect testing of the natural selection often suppressing the positive test signals even if the selection was present. Moreover, these effects affect various tests with diffrent strength, resulting in an interpretation puzzle instead of clear indication in the favor of the natural selection or against it.

Aforementioned difficulty in the interpretation of a battery of tests is the start point for machine learning methodology. The required assumption for successful application of machine learning is that a mosaic of test outcomes, making direct inference so troublesome, containes enough information to diferentiate between the existence of natural selection and its lack. The second prerequisite is the expert knowledge about presence of the selection for given combinations of neutrality test outcomes. Having those two it is possible in principle to train the knowledge retrieving system and, after successful testing, to use it for other genes for which the expert knowledge is unknown.

The paper presents a novel methodology called quasi dominant rough set approach (QDRSA) and compares its advantages and limitations with other rough set based methods: classical rough set approach (CRSA) and dominance based rough set approach (DRSA) using the aforementioned application. Such strategy, except for presentig theoretical aspects about the types of problems adequate for QDRSA, at the same time is able to demonstrate that the class of problems which can be solved with QDRSA is represented in real world applications similar to those considered in the paper.

2 Quasi Dominance Rough Set Approach

Since the first presentation of Rough Set Theory (RST) by Pawlak [8, 9] as an information retrieval system generating rules describing uncertain knowledge in a way alternative to Fuzzy Sets Methodology [12], many modifications of RST have been proposed. The most notable of them include Ziarko's Variable Precision Rough Set Model (VPRSM) [14], Dominance Rough Set Approach introduced by Greco, Matarazzo and Slowinski [5], and Near Set Approach (NST) developed by Peters [10].

The first is dedicated for large data sets where tolerated to some extent inconsistencies can be advanagous, the second is appropriate for attributes with inherent preference order and not necesserily discretized, and the latter uses the discovery of affinities between perceptual objects and perceptual granules that provide a basis for perceptual information systems useful in science and engineering. It is also

worthwhile to notice that there exists also methodology which incorporates Ziarko's idea of variable precision to DRSA methodology resulting in Variable Consistency Dominance Rough Set Approach (VCDRSA) [6].

There have been proposed also other modifications of RST, mainly changing the equivalence relation to weaker similarity relation [11], or defining equivalence relation in continuous attribute space without the need of discretization. Introduction of the structure onto the set of conditional attributes together with application of cluster analysis methodology for this purpose has been proposed by Cyran [1]. The applicability for the problem solved also with the use of CRSA in [3] has been demonstrated in the case study, but it is worth to say that the domain of possible use of modified indiscernibility relation extends to all problems with continuous attributes.

When considering modifications and improvements of Classical Rough Set Approach (CRSA) defined by Pawlak in [9] it may be of some interest to discuss the relation between given enhanced approach and the original CRSA. Basically there are two kinds of this relation: the first is when the modified approach is more general than the CRSA and then the CRSA is a special case of it, and the second is when the modified approach uses the inspiration from CRSA but in fact it defines a new methodology which cannot be reduced to CRSA.

The example of the first type is VPRSM, because CRSA is a special case of VPRSM with precision parameter set to one. Also the modified indisrenibility relation as defined by Cyran in [1] is more general than the original one, since the latter is a special case of the first. Contrary to these examples, the DRSA is such enhancement which cannot be reduced to classical rough sets: it is inspired by the notions present in RST, but the introduction of dominance relation for preference-ordered attributes (called criteria) instead of equivalence relation present in CRSA is the reason why CRSA cannot be derived from DRSA as its special case.

The DRSA is claimed to have many advantages over CRSA in applications with natural preference-ordered attributes. Not denying this statment in general, it is possible to demonstrate the example of such information system with preference-ordered attributes, which, when treated as a decision table, can yield better (in the sense of decision error) decision algorithm A than that generated by DRSA (A_{DRSA}). The superiority of A is also true (however in the sense of generality level) when the aforementioned algorithm A is compared with the algorithm A_{CRSA} obtained by application of CRSA. The quasi dominance rough set approach is the framework within which the algorithm A can be derived. That is why algorithm A will be referred to as A_{QDRSA}.

QDRSA can be considered as a hybrid of CRSA and DRSA. Like DRSA it is dedicated for problems with preference-ordered attributes, but contrary to DRSA, it does not resign from the classical indiscernibility. Therefore the relation I_{CRSA} and I_{QDRSA} are identical. It follows, that for the Information System $S = (U, Q, V_q, f)$ in which $Q = C \cup \{d\}$ and for any $x, y \in U$ the I_{QDRSA} is defined as

$$x I_{\mathrm{QDRSA}} y \Longleftrightarrow \forall q \in C f(x, q) = f(y, q). \tag{1}$$

The notions of lower and upper approximations, quality of approximation, (relative) cores, (relative) reducts and (relative) value reducts are therefore defined in QDRSA like in CRSA.

The consequence of this assumption is that QDRSA, like CRSA, requires discrete values of attributes. This is different from DRSA where corresponding notions rely on preference relation and this apprach does not require discrete attributes.

Similarly to DRSA (and contrary to CRSA), QDRSA is dedicated for problems with preference-ordered attributes, however, because it relies on (1) these attributes need to be of the discrete type. While in some problems it is a clear limitation, in others, namely in such which deal with attributes having inherently discrete nature, the use of classical indiscernibility relation (1) can be advantageous. The illustrative example, concerning real world application in evolutionary genetics, explaines this in more detail. Here, the second limitation of the QDRSA will be given. This limitation is the two-valued domain of the decision attribute $V_d = \{c0, c1\}$, where $c0 < c1$.

Certainly, aforementioned constraint excludes QDRSA from being applied in many problems having more complex decisions. However, there is a vast class of applications for which the binary decision is natural and sufficient. In such cases, if the preference-order is in adddition naturally assigned to the decision, the application of QDRSA can give better effects that either CRSA (which does not take into consideration the preference order) or DRSA (which resignes from indiscernibility relation, what, as it will be shown, can lead to suboptimal solutions).

In general, the types of decision rules obtained in QDRSA are identical to those generated by DRSA. However, because the decision attribute recognizes only two classes and due to relying on indiscernibility (instead of preference) relation, only two types (out of five possible in DRSA) are generated in QDRSA. These decision rules are of the types:

```
if q1 is at least v1 and
   q2 is at least v2 and
       ....            and
   qn is at least vn then
   decision is at least c1
```

and

```
if q1 is at most v1 and
   q2 is at most v2 and
       ....          and
   qn is at most vn then
   decision is at most c0
```

Certainly if only two classes are recognized the conlusions of the two above types of rules can be changed to *decision is c1* or *decision is c0* for the first and the second type respectively. However, for consistency with DRSA, the full syntax with phrases *at least* and *at most* will be used.

The conditions of the decision rules in QDRSA can be obtained from conditions of the correspnding rules in CRSA by introduction of the preference of attribute

values to these conditions. First, it requires the change of equalities to phrases like *at least* for the first type conclusion and *at most* for the second type conclusion. Second, it requires selection of minimal set of conditions (considering all decision rules for the given class), since for example the condition *q1 is at least 2* in one rule and *q1 is at least 3* in the other, are subject for dominance relation. This relation is crucial in DRSA: in QDRSA it is also important, but its realm is reduced to the final stage of the information retrival, as shown above. Therefore in QDRSA but not in DRSA the notion of relative value reduct is exploited with its full potential.

It is also worth to notice that not neccesserily the limitation of the types of decision rules to only two aforementioned is a drawback. For example, the lack of the fifth type of the decision rules possibly generated by DRSA is in fact a pure advantage in all problems with binary decision, since sensless in such conditions decision rules of the type

```
if ... then decision is at least c0 and at most c1
```

are never generated (contrary to DRSA which in certain situations can generate such rules).

In the subsequent text the syntax of QDRSA rules will be a little different. In this slightly modified syntax, the notation of the two types of rules available in QDRSA is more compact:

```
if q1 >= v1 and q2 >= v2 and .... and qn >= vn then at_least.C1
if q1 <= v1 and q2 <= v2 and .... and qn <= vn then at_most.C0
```

3   Illustrative Example

Human evolution at molecular level is reflected in the genome record. Some genes were under strong pressure of natural selection, while genetic variation in others is mainly the result of genetic drift and selectively neutral mutations. If the gene under consideration is exhibiting signatures of natural selection then some variants of it must be more or less fit to the environment. Very often it is associated with some disorder having genetic background, but in some cases it is responsible for the development of the species. The best known example of the latter is the ASPM gene responsible for the brain size in primates, including Humans [13].

There is also third type of selection in which heterozygotes (i.e., organisms having different alleles at two homologues chromosomes) are more fit than any homozygotes (i.e., organisms having identical variants at both homologues chromosomes). This is the case with human sickle cell anemia which is caused by two identical copies of mutated allele. However if this allele is present in heterozygote together with wild-type allele, then the carrier of one copy of mutant allele, not only does not suffer sickle cell anemia, but also is able to generate successful immune response to the malaria. Therefore, on malaria endemic regions the mutant allele is frequent, despite in homozygotes it is responsible for severe disorder.

The type of selection, described above, is called overdominance selection. It is one of the cases of balancing selection – the other case, called underdominance selection is proven to be unstable and the mutant alelle is relatively quickly eliminated

from the population. In the case of balancing selection caused by overdominance mechanism the mutant allele is kept in population for very long time, and sometimes it is even reflected by between-species polymorphism.

3.1 Neutrality Tests

Population geneticists have developed quite a number of statistical neutrality tests which serve to deny at given significance level the neutral Kimura's model. Positive signals generated by them can be interpreted as caused be the presence of natural selection. In the study we consider Tajima's T, Fu's D^* and F_s, Wall's Q and B, Kelly's Z_{nS} and Strobeck's S tests. The definition of them is beyond the scope of the paper, but they are summarized in [4]

When given gene is tested with the use of aforementioned tests, some of them can give positive while others negative signals. Moreover, positive signals can be caused by population expansion or gegraphical structure of the population. On the other hand the signatures of actual natural seletion can be supressed by the recombination. All these factors make the proper interpretation hard and not necesserily univocal.

Cyran and Kimmel have developed multinull hypotheses methodology (partially published in [4] and lately further improved) capable for the reliable interpretation of the test outcomes in the context of natural selection. However, since the method requires modified null hypotheses, the critical values of the tests are unknown and the huge amount of computer simulations must be carried out for estimation of these values.

Therefore, the author proposed application of artificial intelligence (AI) based methods for the interpretation based solely on the test outcomes against classical null hypotheses. In this methodology the battery of tests outcomes is considered as a set of conditional attributes and the expert knowledge is delivered from application of the multinull hypotheses for some small amount of genes. After crossvalidation of the model, the decision concerning other genes can be done based on testing only against classical null hypotheses and application of decision algorithm inferred with AI methodology.

As AI techniques, among others, the rough set approaches were applied. The comparison of CRSA with DRSA for this particular purpose is described in [2] where it is proved that neither CRSA nor DRSA generates decision algorithm which is optimal for the problem considered. The proof is done by a simple demonstration of another algorithm which is Pareto-preferred over both mentioned approaches. This algorithm can be obtained with QDRSA as presented below.

3.2 Decision Algorithms

Consider the information system $S = (U, Q, V_q, f)$ in which $Q = C \cup \{d\}$ The application of CRSA generates the following decision algorithm, referred here to as

*Algorithm*CRSA (Fig. 1). The outcomes of neutrality tests are designated as *NS*, *S*, and *SS* for non-significant, significant, and strongly significant, respectively.

```
BAL_SEL_DETECTED   = False
BAL_SEL_UNDETECTED = False
CONTRADICTION      = False
NO_DECISION        = False
if T = SS or (T = S and D* = S) or ZnS = S then
   BAL_SEL_DETECTED = True
if T = NS or (T = S and D* = NS and ZnS = NS) then
   BAL_SEL_UNDETECTED = True
if BAL_SEL_DETECTED and
   BAL_SEL_UNDETECTED) then
   CONTRADICTION = True
if not(BAL_SEL_DETECTED) and
   not(BAL_SEL_UNDETECTED) or
   CONTRADICTION then
   NO_DECISION = True
```

Fig. 1 *Algorithm*CRSA, adopted from [2]

The algorithm generated by DRSA, called *Algorithm*DRSA is shown in Fig. 2.

```
at_least.BAL_SEL_DETECTED  = False
at_most.BAL_SEL_UNDETECTED = False
CONTRADICTION              = False
NO_DECISION                = False
if T >= SS or (T >= S and D* >= S) or ZnS >= S then
   at_least.BAL_SEL_DETECTED = True
if T <= NS or (T <= S and D* <= NS and ZnS <= NS) then
   at_most.BAL_SEL_UNDETECTED = True
if at_least.BAL_SEL_DETECTED and
   at_most.BAL_SEL_UNDETECTED then
   CONTRADICTION = True
if not(at_least.BAL_SEL_DETECTED)
   and not(at_most.BAL_SEL_UNDETECTED) or
   CONTRADICTION then
   NO_DECISION = True
```

Fig. 2 *Algorithm*DRSA, adopted from [2]

It happened that the algorithm generated by QDRSA *Algorithm*QDRSA is identical to *Algorithm*DRSA when the whole universe *U* of the Information System *S* is used for algorithm genration. However, if the universe of the Information System *S* is

divided into two sets of rules: those used for information retrival in the process of generating the decision algorithm, and those left for testing, then the resulting algorithms generated by DRSA and QDRSA are different in some cases. Below we present only these algorithms which differ between the two approaches.

If the information about RECQL gene is excluded from the information system S and it is left for testing then the DRSA and QDRSA generate the algorithms $Algorithm_{DRSA}(-RECQL)$ and $Algorithm_{QDRSA}(-RECQL)$ respectively. Since the general structure of both algorithms is identical to that of $Algorithm_{DRSA}$, only two crucial if-then rules (the ones after four initialization assignments, and before two contradiction/no-decision determining if-then rules) are presented in Figs. 3 and 4.

```
if (T >= S and D* >= S) or Zns >= S then
    at_least.BAL_SEL_DETECTED = True
if T <= NS or (D* <= NS and Zns <= NS) then
    at_most.BAL_SEL_UNDETECTED = True
```

Fig. 3 $Algorithm_{DRSA}(-RECQL)$

```
if {T >= SS} or
    (T >= S and D* >= S) or Zns >= S   then
    at_least.BAL_SEL_DETECTED = True
if T <= NS or (D* <= NS and Zns <= NS) then
    at_most.BAL_SEL_UNDETECTED = True
```

Fig. 4 $Algorithm_{QDRSA}(-RECQL)$

It is visible that the difference is in the existence of one more condition in the rule describing the detection of balancing selection. This condition reads 'if the outcome of Tajima test is at least strongly statistically significant'. It occurs in $Algorithm_{QDRSA}(-RECQL)$ because condition $T = SS$ is a result of application of relative value reduct for one of the rules in the Information System $S(-RECQL)$ analyzed with CRSA. After changing it to $T >= SS$ when QDRSA is applied it is still not dominated by any other conditions detecting balancing selection. Since it is not dominated it must remain in the final decision algorithm presented above.

However, this is not the case in DRSA. This latter approach, when considering the dominance of decision rules for the class *at-least.BAL-SEL*, compares original (i.e., not reduced with relative value reduct) condition (A) $D^* >= S$ and $T >= SS$ and $Z_{nS} >= S$ with another original condition (B) $D^* >= S$ and $T >= S$ and $Z_{nS} >= NS$, instead of comparing (like QDRSA does) the condition (a) $T >= SS$ with condition (b) $D^* >= S$ and $T >= S$ being the results of application of relative value reducts in QDRSA sense to the original conditions (A) and (B), respectively.

It is clear that rule with condition (A) is dominated by rule with condition (B), and therefore condition (A) seemed to be redundant in DRSA sense for the

class *at-least.BAL-SEL*. However, rule with condition (a) is not dominated by rule with condition (b) and this is the reason why condition (a) is present in the $Algorithm_{QDRSA}(-RECQL)$ while it is absent in $Algorithm_{DRSA}(-RECQL)$. Conditions (B) and (b) in both approaches are necessary and reduced to condition (b) present in both algorithms.

Finally, let us consider what is the influence of inclusion of the condition $T >= SS$ to the $Algorithm_{QDRSA}(-RECQL)$. When this algorithm is applied for the interpretation of neutrality tests for RECQL gene (i.e., the gene which was not present in the Information System $S(-RECQL)$ used for automatic information retrival) for four populations the decision error is reduced from 0.25 to 0. When the full jack-knife method of crossvalidation is applied, the decision error is reduced from 0.313 with DRSA, what seems rather unacceptable, to 0.125 with QDRSA. We have to mention that at the same time QDRSA *NO-DECISION* results have increased from 0 to 0.188, however in the case of screening procedure for which this methodology is intended, the unsure decision is also an indication for more detailed study with the use of multi-null hypotheses methodology.

4 Discussion and Conclusions

DRSA is no doubt a powerfull tool for information retrieval from data representing preference ordered criteria. However, if the problem can be naturally reduced to discrete criteria and binary preference-ordered decision, then this sophisticated construction, designed to be as universal as possible, can be less efficient than QDRSA, dedicated for such type of applications.

The real world illustration is an example that such class of applications is of practical value, at least in all problems with automatic interpretation of a battery of statistical tests. The genetic example with neutrality tests is only one of them. Certainly, many other areas exist having similar properties from the information retrival point of view. In presented illustration the information preserved in the combination of neutrality tests has been retrieved by a novel method called QDRSA.

The comparison of QDRSA with CRSA gives the favor to the first when the preference-order is present in conditional and decision attributes. The resulting decision algorithms in QDRSA are more general, i.e they cover more points of the input space. Moreover, in many cases, because of possible domination of some QDRSA conditions over some other ones, the decision algoriths are shorter as comapred to CRSA. However, because the domination is checked after the application of relative value reducts, the negative effect (characteristic to DRSA) of ommiting the important condition from the decision algorithm (as it was shown in the illustrative example) is not present in QDRSA.

'Artificial intelligence, branching processes, and coalescent methods in the studies concerning RNA-world and Humans evolution'.

References

1. Cyran, K.A.: Modified indiscernibility relation in the theory of rough sets with real-valued attributes: application to recognition of Fraunhofer diffraction patterns. In: Peters, J.F., Skowron, A., Rybiński, H. (eds.) Transactions on Rough Sets IX. LNCS, vol. 5390, pp. 14–34. Springer, Heidelberg (2008)
2. Cyran, K.A.: Classical and dominance based rough sets in the search for genes under balancing selection. In: Transactions on Rough Sets X. LNCS. Springer, Heidelberg (2009) (in press)
3. Cyran, K.A., Mrozek, A.: Rough sets in hybrid methods for pattern recognition. International Journal of Intelligent Systems 16(2), 149–168 (2001)
4. Cyran, K.A., Polanska, J., Kimmel, M.: Testing for signatures of natural selection at molecular genes level. Journal of Medical Informatics and Technologies 8, 31–39 (2004)
5. Greco, S., Matarazzo, B., Slowinski, R.: Rough approximation of preference relation by dominance relations. European Journal of Operational Research 117, 63–83 (1999)
6. Greco, S., Matarazzo, B., Slowinski, R., Stefanowski, J.: Variable consistency model of Dominance-based Rough Sets Approach. In: Ziarko, W.P., Yao, Y. (eds.) RSCTC 2000. LNCS, vol. 2005, pp. 170–181. Springer, Heidelberg (2001)
7. Kimura, M.: The Neutral Theory of Molecular Evolution. Cambridge University Press, Cambridge (1983)
8. Pawlak, Z.: Rough sets. International Journal of Information and Computer Sciences 11, 341–356 (1982)
9. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1992)
10. Peters, J.F.: Near sets. general theory about nearness of objects. Applied Mathematical Sciences 1(53), 2609–2629 (2007)
11. Slowinski, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Transactions on Data and Knowledge Engineering 12(2), 331–336 (2000)
12. Zadeh, L.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)
13. Zhang, J.: Evolution of the Human ASPM gene, a major determinant of brain size. Genetics 165, 2063–2070 (2003)
14. Ziarko, W.: Variable precision rough sets model. Journal of Computer and Systems Sciences 46(1), 39–59 (1993)

Part IV
Decision Support, Rule Inferrence and Representation

The Way of Rules Representation in Composited Knowledge Bases

Agnieszka Nowak and Alicja Wakulicz-Deja

Abstract. The aim of the paper is to find the proper rules representation in composited knowledge bases (CKB). Such representation lets for improving the efficiency of inference processes. To achieve it we can build a modular, hierarchically organized rule base using cluster analysis method is the way of achieving efficient method of rules representation. There are lot of clusters representations, but the most often used is the method based on centroid. If descriptive methods are not sufficient then it is possible to use methods based on visualisation of rules – PCA. With minimal additional effort it provides a roadmap for how to reduce a complex rule set to a lower dimension to reveal the sometimes hidden, simplified structure that often underlies it.

Keywords: composited knowledge bases, knowledge representation, clusters of rules, PCA.

1 Introduction

Knowledge-Based Systems (KBS) are productive tools of Artificial Intelligence (AI) working in a narrow domain to impart quality, effectiveness, and knowledge-oriented approach in decisionmaking process. Being a product of fifth generation computer technology, KBS possess characteristics like providing a high intelligence level, assisting people to discover and develop unknown fields, offering vast knowledge base, aiding management activities and solving social problems in better way. Acquiring new perceptions by simulating unknown situations and offering

Agnieszka Nowak · Alicja Wakulicz-Deja
Institute of Computer Science, University of Silesia,
Będzińska 39, 41-200 Sosnowiec, Poland
e-mail:{agnieszka.nowak, wakulicz}@us.edu.pl
http://zsi.tech.us.edu.pl

significant software productivity improvement or reducing cost and time to develop computerized systems are others advatages of such systems. One of the main components of KBS is the knowledge base, in which domain knowledge, knowledge about knowledge, factual data and procedural rules and so on are available. The inference engine is another component, which infers new knowledge and utilizes existing knowledge for decision-making and problem solving [12]. Unfortunately if we use (possibly different) tools for automatic rules acquisition and/or extraction, the number of rule can rapidly grow. For modern problems, knowledge bases can count up to hundreds or thousands of the rules. One way to understand collected knowledge better is a smart organization of such knowledge. We may reorganize the knowledge base from set of not related rules, to groups of *similar rules* (using cluster analysis) or *decision units* [7, 8]. *Composited knowledge bases* (CKB) are just sets of clusters of similar rules. The formal definition of such sets was presented in [9, 10, 11]. Thanks clustering conditionally similar rules, possibly only small subset of rules need to be checked for a given facts [9]. It optimizes the inference processes because only one chosen cluster is searched, that with the highest similarity value (similarity to the given set of facts). The created structure calls *hierarchical* because applied algorithm of agglomerative hierarchical clustering builds the tree (called *dendrogram*) which shows the hierarchy of rules. Such a tree has all features of the binary tree, thus we can simplify notes that the time efficiency of searching such tree is $O(\log_2 n)$. The hierarchy is a very simple and natural form of presentation the real structure and relationships between data in large data sets. When the number of attributes in data grow it is difficult to understand such data in a simply way. Moreover, it os difficult to represent such complex data. Nowadays, whene graphical methods are very meaningful, finding methods of visualization of the collected knowledge in a form easily understandable by humans is very important. Such methods we want to use to represent data from CKB systems. Our goal is to develop a method for the allocation of large databases on the rule-consistent, and functionally useful subgroups. The method of partitioning is intended to be a division of superstructure rule's model without forcing modification of existing representation rules' methods.

1.1 VSM – Vector Space Model as a Method of Representation of Composited Knowledge Bases

We can use a *Vector Space Model* as a method to represent rules in composited KB. We will say that in such a model, rules are similarly expressed as rows in a matrix, where columns represent unique attributes and the intersection of a column and a row is a value of given attribute.In this model, each rule r is considered to be a vector in the attribute-space. In its simplest form, each rule is represented by the value of attribute vector $\overrightarrow{r} = (v_1^j, v_2^j, \ldots, v_n^j)$, where v_i^j is the jth value of the ith attribute in the rule r. Then we can represent the jth value of ith attribute (v_i^j) in rule r_k as a^i if

$a_i \in r_k$ or as 0 if $a_i \notin r_k$. a^i represents the value of attribute a_i in rule r_k. Combining all rule vectors creates an attribute-rule matrix. We propose to use this representation technique in rule-based application, especially in *inference engine* process with rule browsing interfaces. Such a representation method is useful for both: single rule and cluster of rules, which can be represent as a centroid vector, $\overrightarrow{c} = \frac{1}{|R|}\sum_{r \in R}\overrightarrow{r}$, and it is nothing more than the vector obtained by averaging the values of the various attributes present in the rules of R [8].

2 The Cluster Representation

Generalizing we propose to use for rules clusters one of the following representations:

- descriptive representation for cluster (the centroid)- using *VSM* model,
- graphical representation using dendrograms (*AHC*), clusters's distribution (*k*-means, *k*-medoid).

2.1 Descriptive Forms Rules Clusters Representation

A *centroid* (center of gravity, for example, the average of vector objects group) or medoid (object, the distance to which all other objects in the group is the smallest) are the most often used descriptive representations for rules clusters. We can also use a vector of most occurring features in the group, the most distinctive features of vector groups from each other or a boundary characteristic of a group for cluster representation. For numerical data (with n objects of d-dimensions in a group $\{X_i\}$, $i = 1, 2, \ldots, n$) it is possible to use *centroid* $(c_m = (\sum_{i=1}^{n} X_i)/n)$, *radius*: the square root of the average distance of all objects in the group to the middle of the group (centroid) $R_m = \sqrt{(\sum_{i=1}^{n}(X_i - c_m)^2)/n}$ and *diameter*: square root of the average distance between all pairs of objects in the group: $D_m = \sqrt{(\sum_{i=1}^{n}\sum_{i=1}^{n}(X_i - X_j)^2)/(n(n-1))}$. Radius R_m and diameter D_m are used to determine the concentration of objects in the group around the midpoint. Representation by means of cluster points of boundary is proposed for clusters of long, irregularly shaped, this approach fails. It is a set of points selected from the edge of cluster representation to its shape. Number of points used to grow along with the increasing complexity of shape (Fig. 1).

Fig. 1 Boundary characteristic of the cluster
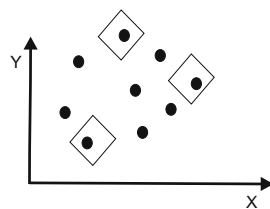
Fig. 2 Representation
through branches in the
classification tree
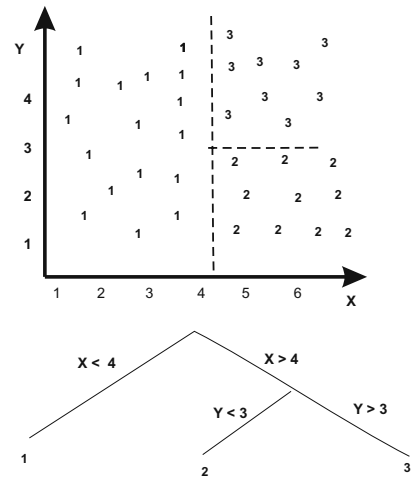
Representation through branches in the classification tree is presented in Fig. 2. Representation by groups describe their actions logical is based on the assumption that each path in the tree classification from the highest to the lowest level corresponds to a logical expressions:

$$1 : [X < 4], \quad 2 : [X > 4][Y < 3], \quad \text{and } 3 : [X > 4][Y > 3].$$

The assumption of the representation based on the taxonomy of symbolic value is to focus on the distribution (homogeneous groups) and to find the characteristics of these classes (their description by means of appropriate concepts). Natural way to characterize the classes are rules for conditions that must be met by objects belonging to them [4]. It is based on using *variable-valued logic* (concepts: selector and complex) and so-called conceptual cluster analysis (*conceptual clustering*). Each cluster is characterized by *complex* – the conjunction of attribute values of objects belonging to them (e.g., [*color=green*] ∧ [*taste=sweet*] ∧ [*shape=round*] [6].

2.1.1 The Advantages and Disadvantages of the *VSM* Model

The disadvantages of *VSM* model are: omitting information about the validity of attributes in the rules, assuming the independence of the attributes, the fact that the number of dimensions must be known in advance and problem with high-dimensional data (medical (e.g., EEG), financial (stock market indices) multimedia (pictures, movies), purchases made using credit cards, time series). The advantage is representation of even very complex data sets, which using effective artificial intelligence techniques (clustering) giving the smart data organization and optimizing the systems with such data. Moreover, operations on vectors (counting the distance,
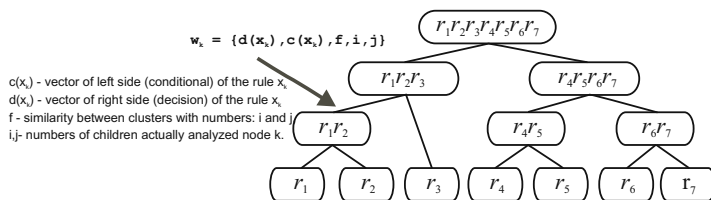
Fig. 3 The dendrogram

weighing, etc.) are easier and more efficient in calculation from the competitive representation, for example, based on a graphical model.

2.2 The Graphical Representation of Clusters – Dendrograms for Hierarchical Clustering Methods

Dendrogram is a tree received as a result of the hierarchical cluster analysis method (both techniques: agglomerative and divisive). In the first case, the construction goes from bottom to top of tree, instead of the the second case, where tree construction goes from the top to the bottom of the tree.

The problem with dendrogram is that it does not show the relationship of proximity combined objects (rules). Moreover, large collections of attributes brings a risk, so-called 'dimensionality curse' (concept probably invented by mathematician Richard Bellman). With increasing dimensionality the distance to the nearest point moves closer to the furthest distance. This effect can already be visible for 10–15 dimensions [1].

3 Graphical Methods for Representation of Multidimensional Data

The task of unsupervised learning is to describe – a sort of 'clarification' – observed data input on the basis of only themselves. Such a task can be defined as the task of detecting the internal structure of a data set or between those data [2].

Multidimensional data analysis is a task of complicated calculation – the more the greater is the dimension of data. This requires an increase in the number of cases. It is impossible to their visualization. Therefore, it is reasonable to reduce the data aggregation, clustering analysis, cross-examination of volatility. When measuring real data the phenomenon of a few variables (dimensions) of this data is that some of them are independent to another. An important problem of large data exploration is to reduce the dimensions, both in terms of the number of observation and the number of features. For this problem, we can approach in two ways: *feature selection* or and feature *extraction*. *Feature selection* in space of objects (rules) is to select small subset of attributes, which gives the greatest possible chance to distinguish

the rules [3]. Feature extraction uses all available information in order to transition
to the new spaces, with reduced number of dimensions. This transformation may
be linear or nonlinear. Examples of linear transformation is Principal Components
Analysis (PCA) and Singular Value Decomposition (SVD).

3.1 PCA as a Graphical Representation Method

The PCA features are following:

- reduces the data (hence often used to compress data),
- transforms k correlated variables into the k main components,
- graphically presents the structure of multi-dimensional data set on a plane with
  the minimum distortion of information,
- removes noise in observed data.

The method transports the information contained in a number of variables to small
number of dimensions (method presented by Hotelling in 1933 [5]). With minimal
additional effort PCA provides a roadmap for how to reduce a complex data set
to a lower dimension to reveal the sometimes hidden, simplified structure that often
underlie it. The other main advantage of PCA is that once it has found these patterns
in the data, and compresses the data, ie. by reducing the number of dimensions,
without much loss of information (used in image compression).

4 The Overview of the Experiments

The experiments consist of two procedures: using PCA method for the set of col-
lected objects and for the set of rules extracted from such a set of objects. For the
very well known set of iris objects with 150 cases and 4 attributes, the set of rules
is extracted by an exhaustive algorithm. It consists of 246 elements. In both cases
we first clustering objects using AHC method and then we use the PCA method
for visualization achieved structure in a biplot. A *biplot* is plot which aims to rep-
resent both the observations and variables of a matrix of multivariate data on the
same plot. The result of PCA visualization for the set of iris objects is presented as
a biplot in Fig. 4a, whereas result of PCA visualization for the set of rules extracted
from iris data set is presented in Fig. 4b. We can see that rules visualization using
PCA projection is a method simple to understand the data. It is well known that iris
data set classify all objects from this set to one of three given classes. In the picture
it is possible to see that objects are distributed to such three groups. Two of four
given attributes are correlated together. In Fig. 4a it is viewed as parallel factors.
Rules extracted form the set of objects are the set of data without the noisy data,
and data that are not important. That is why in Fig. 4b all objects presented are well
distributed (well separated).

(a)   (b)
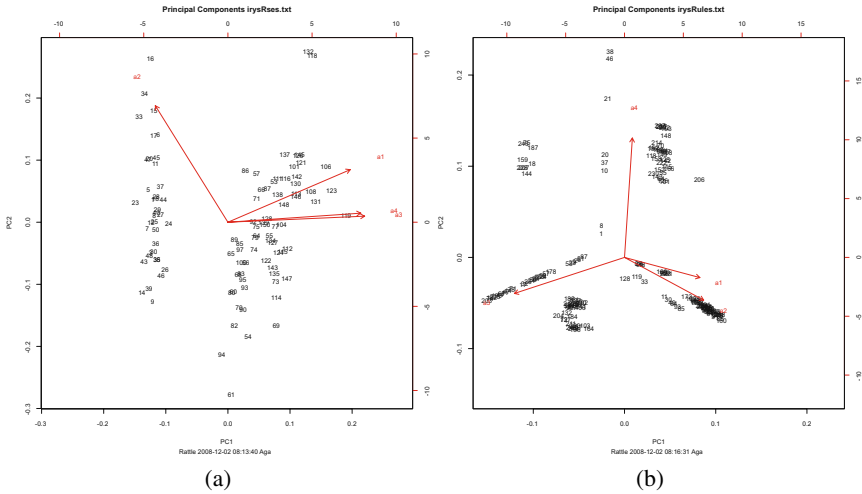
Fig. 4 (a) Biplot for IRIS data set **(b)** Biplot for rules extracted from IRIS data set

5   Future Researches – Using Both: Cluster Analysis and PCA?

In the future we want to check if it is possible to use both method for exploration data in composited knowledge bases. After using *PCA* method, that reduces unimportant data, we will clustering the achieved set. In our opinion clustering without dimensions reduction may give us not exactly such results as we want to get, because we clustering potentially correlated data. If we first reduce data to some smaller number of dimensions, then clustering such reduced data let us achieve very well clustered groups of objects. High dimensionality data often consist of redundancy of attributes. Values of objects are not equally distributed of all directions of highdimensional coordinate system, but they are concentrate to some subspaces. Finding such subspaces (with usually much smaller dimension than original space) let us to see some regularities, invisible or impossible to see in original coordinate system.

6   Conclusions

There are a lot of methods of representation of knowledge in composited knowledge bases. Modularization of given by clustering rules allows us to optimize the efficiency of inference process. These clusters of rules can be represent by a descriptive methods but such methods are not optimal representation for complex data. The VSM model with centroid as a geometrical center (mean) of a given group not always characterizes the cluster of objects properly. The symbolical taxonomy lets us build optimal characteristics of clusters, fulfilling the criterions: universality of object's characteristics, the simplicity of objects's characteristics, the similarity of objects, the separability of clusters and high quality of discrimination. It needs to

measure the quality of distribution. Even the dendrogram as a graphical representation of clusters is not sufficient. It does not show how similar are objects together. That is why we propose to use PCA method for visualization created structure of rules clusters. Factorial has got an exploration character. They separate variables hidden a posteriori. The PCA let to reduce data by replace correlated variables to the new data set. The reduction is based on the compromise between reduction of structure and the quality of such transposition. In our opinion method that is a connection of different conceptions of clusters representations would be optimal solution. It is obviously expensive way of knowledge representation but in our opinion it is necessary to properly interpret and understand the knowledge of presented system.

References

1. Beyer, K.S., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is 'Nearest Neighbor' meaningful? In: Beeri, C., Bruneman, P. (eds.) ICDT 1999. LNCS, vol. 1540, pp. 217–235. Springer, Heidelberg (1998)
2. Ćwik, J., Koronacki, J.: Statistical learning systems. Wydawnictwa Naukowo-Techniczne (2005) (in Polish)
3. Friedman, J.H., Meulman, J.J.: Clustering objects on subsets of attributes. Royal Statistical Society 66, 815–849 (2004)
4. Gatnar, E.: Symbolical methods of data classification. Państwowe Wydawnictwo Naukowe (1998) (in Polish)
5. Hotelling, H.: Analysis of a complex of statistical variables into principal components. Journal of Education Psychology 24 (1933)
6. Michalski, R.S.: Variable-valued logic and its application to pattern recognition and machine learning. North-Holland, Amsterdam (1977)
7. Nowak, A., Simiński, R., Wakulicz-Deja, A.: Towards modular representation of knowledge base. Advances in Soft Computing 5, 421–428 (2006)
8. Nowak, A., Simiński, R., Wakulicz-Deja, A.: Knowledge representation for composited knowledge bases. In: Kłopotek, M.A., Przepiórkowski, A., Wierzchoń, S.T., Trojanowski, K. (eds.) Intelligent Information Systems, pp. 405–414 (2008)
9. Nowak, A., Wakulicz-Deja, A.: The concept of the hierarchical clustering algorithms for rules based systems. Advances in Soft Computing 31, 565–570 (2005)
10. Nowak, A., Wakulicz-Deja, A.: The inference processes on clustered rules. Advances in Soft Computing 5, 403–411 (2006)
11. Nowak, A., Wakulicz-Deja, A.: The inference processes on composited knowledge bases. In: Kłopotek, M.A., Przepiórkowski, A., Wierzchoń, S.T., Trojanowski, K. (eds.) Intelligent Information Systems, pp. 415–422 (2008)
12. Sajja, P.S.: Multi-agent system for knowledge-based access to distributed databases. Interdisciplinary Journal of Information, Knowledge and Management 3, 1–9 (2008)

Clustering of Partial Decision Rules

Agnieszka Nowak and Beata Zielosko

Abstract. The aim of the paper is to cluster partial decision rules using Agglomerative Hierarchical Clustering (AHC) algorithm. Partial decision rules are constructed by greedy algorithm. We study how exact and partial decision rules clustered by AHC algorithm influence on inference process on knowledge bases. Clusters of rules are a way of modularization of knowledge bases in Decision Support Systems. The results of rules clustering are searched during the inference process only the most relevant rules, what makes the inference process faster. Results of experiments present how different factors (rule length, number of facts given as an input knowledge, value of α parameter used to construct partial decision rules, number of rules in a given knowledge base) can influence on the efficiency of inference process.

Keywords: rules clusters, partial decision rules, greedy algorithm, composited knowledge bases, knowledge representation.

1 Introduction

Knowledge based systems are created for specialized domains of competence, in which effective problem solving normally requires human expertise. In recent years, knowledge based systems technology has proven itself to be a valuable tool for solving hitherto intractable problems in domains such a telecommunication, aerospace, medicine and the computer industry itself. The goals of knowledge based systems are often more ambitious than conventional programs. They frequently perform not only as problem solvers but also as intelligent assistant and training aids, they help their creators and users to understand better own knowledge [4]. Proper knowledge

Agnieszka Nowak · Beata Zielosko
Institute of Computer Science, University of Silesia,
Będzińska 39, 41-200 Sosnowiec, Poland
e-mail:{agnieszka.nowak, beata.zielosko}@us.edu.pl
http://zsi.tech.us.edu.pl

acquisition is one of the main goal of systems based on knowledge. Usually they are implemented using knowledge acquired from human experts or discovered in data bases (for example using rough set theory [5, 13]). Rules are the most popular method of knowledge representation. Unfortunately if we use – possibly different – tools for automatic rules acquisition and/or extraction, the number of rules can rapidly grows. For modern problems, knowledge bases can count up to hundreds or thousands of rules. For such knowledge bases, number of possible inference paths is very high. In such cases knowledge engineer can not be totally aware that all possible rules interactions are legal and provide to expected results. It brings problems with inference efficiency and interpretation of inference results. For example, if we consider forward reasoning, a lot of fired rules forming a lot of new facts that are sometimes difficult to properly interpret and they may be useless for user [7, 8]. The problem of efficiency may be important in technical applications, especially in real-time systems. Inference methods are well known, there are well described algorithms for forward and backward reasoning and some of their modifications [4]. We believe that the increase of efficiency should not rely on the modification of these algorithms but on modification of data structures used by them. We build knowledge base from partial decision rules constructed by greedy algorithm. Partial decision rules consist of smaller number of attributes than exact rules [14]. We can say that such rules are less complex and better for understand. It is not the only improvement of the efficiency we can do. We propose to reorganize the knowledge base from set of not related rules, to groups of *similar rules* (using cluster analysis method). Thanks to clustering of conditional parts of similar rules, possibly only small subset of rules need to be checked for a given facts, which influence on performing of inference process [7, 11]. The paper consists of five sections. In Sect. 2 main notions of partial decision rules and greedy algorithm are considered. In Sect. 3 description of performing of inference on clusters of rules is presented. In Sect. 4 results of experiments with real-life decision tables are considered. Section 5 contains conclusions.

2 Partial Decision Rules

Decision rules can be considered as a way of knowledge represenatation. In applications we often deal with decision tables which contain noisy data. In this case, exact decision rules can be 'over-fitted', i.e., depend essentially on the noise. So, instead of exact decision rules with many attributes, it is more appropriate to work with partial decision rules with small number of attributes, which separate almost all different rows with different decisions [14].

The problems of construction decision rules with minimal number of attributes, are NP-hard. Therefore, we should consider approximate polynomial algorithms.

In [5] we adapted well known greedy algorithm for set cover problem to construction of partial decision rules. From obtained bounds on greedy algorithm accuracy and results proved in [6] it follows, that under some natural assumptions on the class

NP, the greedy algorithm is close to the best polynomial approximate algorithms, for minimization of partial decision rule length.

Now, main notions for partial decision rules are presented.

Let T be a table with n rows labeled with decisions and m columns labeled with attributes (names of attributes) $a_1, \ldots, a_m$. This table is filled by values of attributes. The table T is called a *decision table* [13].

Two rows are called *different* if they have different values at the intersection with at least one column a_i.

Let $r = (b_1, \ldots, b_m)$ be a row from T labeled with a decision d. We will denote by $U(T, r)$ the set of rows from T which are different (in at least one column a_i) from r and are labeled with decisions different from d. We will use the parameter α to denote a real number such that $0 \le \alpha < 1$. A decision rule

$$(a_{i_1} = b_{i_1}) \wedge \ldots \wedge (a_{i_t} = b_{i_t}) \to d$$

is an α-*decision rule* for the row r of decision table T if attributes $a_{i_1}, \ldots, a_{i_t}$ separate from r at least $\lceil (1 - \alpha)|U(T, r)| \rceil$ rows from $U(T, r)$. We will say that an attribute a_i *separates* a row $r' \in U(T, r)$ from the row r, if the rows r and r' have different values at the intersection with the column a_i.

For example, 0.01-decision rule means that attributes contained in this rule should separate from r at least 99% of rows from $U(T, r)$. If α is equal to 0 we have an exact decision rule. Algorithm 1 (presented in Fig. 1) describes the greedy algorithm with threshold α which constructs an α-decision rule for the row r of decision table T.

Input : Decision table T with conditional attributes $a_1, \ldots, a_m$, row $r = (b_1, \ldots, b_m)$ of T labeled by the decision d and real number α, $0 \le \alpha < 1$.
Output: α-decision rule for the row r of decision table T
$Q \longleftarrow \emptyset$;
while *attributes from Q separate from r less than* $(1 - \alpha)|U(T, r)|$ *rows from* $U(T, r)$ **do**
    select $a_i \in \{a_1, \ldots, a_m\}$ with minimal index i such that a_i separates from r the maximal number of rows from $U(T, r)$ unseparated by attributes from Q $Q \longleftarrow Q \cup \{a_i\}$;
end
return $\bigwedge_{a_i \in Q}(a_i = b_i) \to d$;

Fig. 1 Greedy algorithm for partial decision rule construction

3 The Hierarchical Structure of Knowledge Base

It is known that cluster analysis brings useful technique to reorganize the knowledge base to hierarchical structure of rules. The hierarchy is a very simple and natural form of presentation the real structure and relationships between data in large data sets. Instead of one long list of all rules in knowledge base, better results we achieve if we build composited knowledge base as a set of groups of similar rules. That is why we used agglomerative hierarchical clustering algorithm to build a tree of rules

clusters (called *dendrogram*). Such a tree has all features of binary tree, so we notes that the time efficiency of searching such trees is $O(\log_2 n)$. Such an optimization of the inference processes is possible because small part of whole knowledge base is analyzed only (the most relevant to the given set of facts (input knowledge)) [8].

3.1 The Knowledge Base Structure

Having both: X as a set of rules in given knowledge base and F_{sim} as similarity function connected with set of rules from X, we may build hierarchically organized model of knowledge base. Such model for n rules $X = \{x_1, \ldots, x_n\}$, where each rule uses set of attributes A and values of such attributes V ($V = \cup_{a \in A} V_a$ is a set of values of attribute a), is represent as labeled binary tree $Tree = \{w_1, \ldots, w_{2n-1}\} = \bigcup_{k=1}^{2n-1} \{w_k\}$, created by clustering rules using such similarity function. Labels of nodes in such tree are fives: $\{d(x_k), c(x_k), f, i, j\}$, where $c(x_i) \in V_1 \times V_2 \times \cdots \times V_m$ is a vector of left hand-side (conditional part) of the rule x_k and $d(x_i) \in V_1 \times V_2 \times \cdots \times V_m$ is a vector of right hand-side (decision part) of such rule, and $f = F_{\text{sim}} : X \times X \longrightarrow [0 \ldots 1]$ is value of similarity between two clustered rules (or rules) x_i and x_j. Elements i and j are numbers of children actually analyzed kth node.

3.2 Agglomerative Hierarchical Clustering of Rules

Agglomerative algorithm (given in Fig. 2) starts with each object being a separate itself, and successively merge groups according to a distance measure. The clustering may stop when all objects are in a single group (classical AHC). We could also use modified AHC, so called mAHC, which is widely presented in [7, 12].

Algorithm gets as input a finite set O of n objects and a matrix of pairwise distances between these objects. This means that executing of clustering algorithm is completely independent on distances between the objects were computed. The algorithm starts with a trivial clustering c_0 with n singleton clusters. At each iteration phase two clusters c_i, c_j with highest similarity in c_{k-1} are searched and merged. A new clustering C_k is formed by removing these two clusters and adding the new merged cluster, e.g., C_k is C_{k-1} with clusters c_i and c_j merged. It is continued until

---

Input : A set O of n objects and a matrix of similarities between the objects.
Output: Clusterings $C_0, C_1, \ldots, C_{n-1}$ of the input set O;
$\quad\quad C_0 =$ the trivial clustering of n objects in the set input set O;
while $|C| > 1$ **do**
$\quad\quad$ find $c_i, c_j \in C_{k-1}$ where similarity $s(c_i, c_j)$ is maximal; $C_k = (C_{k-1} \backslash \{c_i, c_j\}) \cup (c_i \cup c_j)$;
$\quad\quad$ calculate similarity $s(c_i, c_j) \; \forall \, c_i, c_j \in C_k$;
end
return $C_0, C_1, \ldots, C_{n-1}$;

Fig. 2 Agglomerative hierarchical clustering algorithm for rules in knowledge bases

there is only one cluster left. The output of the algorithm is the sequence of clusterings $C_0, C_1, \ldots, C_{n-1}$. A centroid as the representative of created group-cluster is calculated as the average distance all objects till the given cluster. Various similarity and distance metrics were analyzed for clustering, because it is very important to use proper metrics [2]. Analysis of clustering efficiency is presented in [3]. The results show that only the *Gower's* measure is effective for such different types of data that are in composited knowledge bases.

3.3 Inference Process on Hierarchical Knowledge Base

In decision support systems without rules clusters, rule interpreter has to check each rule, one by one, and firing these which exactly match to given observations. It takes $O(n)$ time efficiency, for n as the number of rules, whereas in decision support systems with hierarchical structure of knowledge base the time efficiency is minimalized to $O(\log 2n - 1)$, where n is the number of rules clusters. In this situation rule interpreter does not have to check each rule – one by one. The process of inference is based on searching binary tree with rules clusters and choosing nodes with highest similarity value. There are various types of searching tree techniques. We can choose one of two given methods of searching trees: so called *'the best node in all tree'* or *'the minimal value of coefficient'*. Both are widely presented in [7, 12].

4 Experiments with Data from UCI Machine Learning Repository

We did experiment with 4 data sets ('zoo', 'lymphography', 'spect_all', 'nursery') from [1]. For every data set we use the greedy algorithm for partial decision rules constraction with parameter $\alpha \in \{0.0, 0.001, 0.01, 0.1\}$. Next, for every data set of rules we build knowledge base and use AHC algorithm for inference process. Table 1 presents results of experiments. With each experimental set we note number of attributes (N_{atr}), average, minimal and maximal length of rule ($rl_{avg}, rl_{min}, rl_{max}$), value of α parameter, number of rules (N_R), number of nodes searched in the dendrogram using 'the best node in all tree' method (N_n) and different number of input knowledge (facts)(N_{data}). For such items we checked number of relevent rules in whole knowledge base (N_{rr}), number of relevant rules successively searched (N_{rs}), and number of searched rules that are not relevant (N_{nrs}). These information let us to calculate the *precision* of the proposed method (*Precision*) and the percent of the whole knowledge base, which is really searched (%KB). It is a measure of probability of searching only relevant rules during the inference process. It was considered further in [10]. For us, as it is in Information Retrieval, a perfect precision equal to 1.0 means that every result retrieved by a search was relevant (but says nothing about whether all relevant rules were retrieved, simply bacause there is no need to search each relevant rule).

Table 1 Experimental results

set	N_{atr}	rl_{avg}	rl_{min}	rl_{max}	α	N_R	N_n	N_{data}	N_{rr}	N_{rs}	N_{nrs}	Precision	% KB
Zoo	16+1	1.48	1	4	0	101	20	1	41	1	0	1	10.05%
Zoo	16+1	1.48	1	4	0	101	14	2	41	1	0	1	7.03%
Zoo	16+1	1.48	1	4	0	101	12	3	1	1	0	1	6.03%
Zoo	16+1	1.48	1	4	0.001	101	20	1	41	1	0	1	10.05%
Zoo	16+1	1.48	1	4	0.001	101	10	2	4	1	0	1	5.03%
Zoo	16+1	1.48	1	4	0.001	101	12	4	1	1	0	1	6.03%
Zoo	16+1	1.48	1	4	0.01	101	18	1	41	1	0	1	9.05%
Zoo	16+1	1.48	1	4	0.01	101	12	2	13	1	0	1	6.03%
Zoo	16+1	1.48	1	4	0.01	101	12	4	1	1	0	1	6.03%
Zoo	16+1	1.059	1	2	0.1	101	12	1	8	1	0	1	6.03%
Zoo	16+1	1.059	1	2	0.1	101	8	2	1	1	0	1	4.02%
Zoo	16+1	1.059	1	2	0.2	101	12	1	8	1	0	1	6.03%
Zoo	16+1	1.059	1	2	0.2	101	10	2	2	1	0	1	5.03%
Zoo	16+1	1	1	1	0.5	101	14	1	41	1	0	1	7.03%
lymphography	18+1	2.12	1	4	0.01	148	12	1	7	1	0	1	4.07%
lymphography	18+1	2.12	1	4	0.01	148	12	2	1	1	0	1	4.07%
lymphography	18+1	2.12	1	4	0.01	148	16	3	3	1	0	1	5.42%
lymphography	18+1	1.48	1	2	0.1	148	18	2	2	1	0	1	6.10%
lymphography	18+1	1.06	1	2	0.2	148	20	1	11	1	0	1	6.78%
lymphography	18+1	1.06	1	2	0.2	148	14	2	2	1	0	1	4.75%
lymphography	18+1	1	1	1	0.5	148	26	1	11	1	0	1	8.81%
spect_all	23+1	3.18	1	10	0	267	16	1	38	1	0	1	4.50%
spect_all	23+1	3.18	1	10	0	267	12	9	6	1	0	1	1.13%
spect_all	23+1	3.18	1	10	0	267	4	9	6	1	0	1	1.13%
spect_all	23+1	3.18	1	10	0.001	267	12	9	6	1	0	1	1.50%
spect_all	23+1	3.18	1	10	0.001	267	24	9	1	1	0	1	1.50%
spect_all	23+1	3.18	1	10	0.01	267	8	9	1	1	0	1	1.88%
spect_all	23+1	1.55	1	7	0.1	267	26	1	38	1	0	1	3.00%
spect_all	23+1	1.29	1	5	0.2	267	16	5	19	1	0	1	2.63%
spect_all	23+1	1.29	1	5	0.2	267	16	2	19	1	0	1	3.75%
spect_all	23+1	1.29	1	5	0.2	267	16	1	6	1	0	1	1.50%
spect_all	23+1	1.022	1	2	0.5	267	14	1	34	1	0	1	3.00%
spect_all	23+1	1.022	1	2	0.5	267	20	1	11	1	0	1	2.25%
spect_all	23+1	1.022	1	2	0.5	267	8	2	6	1	0	1	0.75%
nursery	8+1	3.27	1	8	0	1000	22	5	4	1	0	1	0.91%
nursery	8+1	3.27	1	8	0	1000	20	1	333	1	0	1	0.99%
nursery	8+1	3.27	1	8	0	1000	20	5	4	1	0	1	0.99%
nursery	8+1	2.89	1	6	0.001	1000	16	1	216	1	0	1	1.25%
nursery	8+1	2.36	1	4	0.001	1000	12	5	1	1	0	1	1.66%
nursery	8+1	1.66	1	2	0.01	1000	10	3	2	1	0	1	1.99%
nursery	8+1	1.66	1	2	0.01	1000	16	1	216	1	0	1	1.25%
nursery	8+1	1.14	1	2	0.2	1000	18	1	333	1	0	1	1.11%
nursery	8+1	1.14	1	2	0.2	1000	22	1	576	1	0	1	0.91%
nursery	8+1	1.14	1	2	0.2	1000	16	1	91	1	0	1	1.25%
nursery	8+1	1	1	1	0.5	1000	18	1	333	1	0	1	1.11%
nursery	8+1	1	1	1	0.5	1000	16	1	91	1	0	1	1.25%
nursery	8+1	1	1	1	0.5	1000	22	1	576	1	0	1	0.91%

If we use clustering rules before inference process in given knowledge base, then instead of all set of rules only small percent of the rules is searched. In large knowledge bases we can significantly reduce the number of rules relevant to the inference initial data. Reduction in the case of backward reasoning is less significant than in case of forward reasoning and depends on selected goal, and need to search solution in small subset of rules. Results of experiments show that inference process is sucessful, if for given set of facts the relevant rules were found and fired, without

searching whole knowlege base. As it can be seen in the table, the more rules the knowledge base consists of, the less part of it is really searched. For datasets with 1000 of rules it is even less than 1% of whole knowledge base. Different datasets were analyzed, with different type of data, and various number of attributes and their values. Also different values of parameter α were checked. We can see that with the growth of parameter α the length of rules is fall of. If the value of α is bigger then the length of rules is longer. For high value of α (e.g., $\alpha = 0.5$) rules are the shortest what makes inference process faster. It is because the relevant rule has small number of conditions (for $\alpha = 0.5$ each rule consists of only one condition) that must be checked with set of facts in given knowledge base. For longer rules the time of rule checking is longer. For $\alpha = 0$ in datasets like *'spect_all'* or *'nursery'* maximal lenght of rule is even 10. It is obvious that it makes the inference process slower and less efficient. The best results we can achive for large knowledge bases with high value of α, because in this case the small percent of whole knowledge base is searched and the inference process made on searched rules is really fast.

5   Conclusions

In our opinion modularization methods presented in this paper allow us to optimize the efficiency of inference process. Using modular representation we can limit the number of rules to process during the inference. Thanks to properties of rules cluster we can perform inference algorithm optimizations, depending on user requirements. The length of rules influence on the efficiency of inference process on knowledge base. Exact rules can be overfitted or depending on the noise. So, instead of exact rules with many attributes we use partial rules with smaller number of attributes, which separates from a given row almost all other rows with different decision. Based on results from [5] we used greedy algorithm for partial decision rules construction, because it was proved that under some natural assumptions on the class NP, the greedy algorithm is close to the best polynomial approximate algorithms, for minimization of partial decision rule length. We made first test on real-world rules sets which confirm our theoretical expectation. Hierarchical organization of rule knowledge base always leads to decrease of the number of rules necessary to process during inference and global inference efficiency grows.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
2. Ćwik, J., Koronacki, J.: Statistical learning systems. Wydawnictwa Naukowo-Techniczne (2005) (in Polish)
3. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, New York (1990)

4. Luger, G.: Artificial Intelligence, Structures and Strategies for Complex Problem Solving. Addison-Wesley, Reading (2002)
5. Moshkov, M.J., Piliszczuk, M., Zielosko, B.: On partial covers, reducts and decision rules. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets VIII. LNCS, vol. 5084, pp. 251–288. Springer, Heidelberg (2008)
6. Moshkov, M.J., Piliszczuk, M., Zielosko, B.: Partial covers, reducts and decision rules in rough sets: theory and applications. Studies in Computational Intelligence, vol. 145. Springer, Heidelberg (2009)
7. Nowak, A., Simiński, R., Wakulicz-Deja, A.: Towards modular representation of knowledge base. Advances in Soft Computing 5, 421–428 (2006)
8. Nowak, A., Simiński, R., Wakulicz-Deja, A.: Knowledge representation for composited knowledge bases. In: Kłopotek, M.A., Przepiórkowski, A., Wierzchoń, S.T., Trojanowski, K. (eds.) Intelligent Information Systems, pp. 405–414 (2008)
9. Nowak, A., Wakulicz-Deja, A.: The concept of the hierarchical clustering algorithms for rules based systems. Advances in Soft Computing 31, 565–570 (2005)
10. Nowak, A., Wakulicz-Deja, A.: The analysis of inference efficiency in composited knowledge bases. In: Proceedings of Decision Support Systems Conference, pp. 101–108. Zakopane, Poland (2008) (in Polish)
11. Nowak, A., Wakulicz-Deja, A.: The inference processes on composited knowledge bases. In: Kłopotek, M.A., Przepiórkowski, A., Wierzchoń, S.T., Trojanowski, K. (eds.) Intelligent Information Systems, pp. 415–422 (2008)
12. Nowak, A., Wakulicz-Deja, A., Bachliński, S.: Optimization of speech recognition by clustering of phones. Fundamenta Informaticae 72, 283–293 (2006)
13. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
14. Zielosko, B., Piliszczuk, M.: Greedy algorithm for attribute reduction. Fundamenta Informaticae 85(1-4), 549–561 (2008)

Decision Trees Constructing
over Multiple Data Streams

Jerzy Martyna

Abstract. In this paper, we consider the decision trees-building question over multiple data streams. The online data mining over each data stream is often computationally prohibitive. Therefore, we propose an analysis for the problem of a large specimen. It concerns a case of satisfactorily large cardinality of samples. With the help of a given algorithm for the decision tree construction, we obtained data multistream decision trees. Experimental results demonstrate that the received decision trees can be efficient tools for the classification analysis.

Keywords: data mining, decision trees.

1 Introduction

One of the most interesting methods of data mining are decision trees, which were introduced by Breiman et al. [3]. According to this approach the decision trees combine classification with application in which they are used. To the best known decision trees belong CART [3], ID3 [10] and C4.5 [11].

Currently, the studies in the decision tree classification are concentrated on the new model of data sets which are in the form of continuous data streams. Such features of data sets are generated by some applications which are associated with the network data [4] or generated by the sensor networks [12], etc. These data streams come from the observation of financial transactions conducted by stock markets [13], etc.

Jerzy Martyna

Institute of Computer Science, Jagiellonian University,
Lojasiewicza 6, 30-348 Cracow, Poland
e-mail: martyna@softlab.ii.uj.edu.pl

The goal of this paper is to find the set of a collection of items (or any objects), where the occurrence count is at least greater than a certain threshold determined for each stream. In practice, due to the limited space and the need for real-time analysis, it is impossible to exactly count all the collections for each of the incoming items (transactions) in the stream. Hence, the threshold value determined by use of the proposed analysis allows for each stream to selective sampling across time. Thus, we can find the decision tree with a defined error margin in the collections of larger sizes. As a result, the suggested method guarantees the minimum error for frequent sets of different sizes.

The paper is organized as follows. Section 2 presents a brief problem formulation of a decision tree construction. Then, Sect. 3 introduces our approach to the threshold value analysis for each stream separately. Section 4 describes the procedure of the decision tree building for data streams. Section 5 concludes the paper.

2   The Question of Decision Tree Construction

In this section, we give background information on the decision tree construction question.

Let $D = \{d_1, d_2, \ldots, d_N\}$ be a data set, where $d_i = \langle \overline{X}, c \rangle \in \mathbf{X} \times C$. We assume that $\mathbf{X} = \langle X_1, X_2, \ldots, X_m \rangle$ is the datum associated with the instance and c is the class label. Each X_j is called an attribute of data instances. $\mathbf{X}$ is the domain of data instances. The domain of an attribute can either be a categorical set or a numerical set. Let C be a domain of the class label.

One of the criteria used for the partition of the input data is the application of entropy for a given training data set S, $S = \{g(X_1), g(X_2), \ldots, g(X_m)\}$, where $g(X_i)$ denotes the best gain possible using this attribute for splitting the node. If we have m attributes, we wish to determine i, such that

$$g(X_i) \geq \max_{j \in \{1, \ldots, m\} - \{i\}} g(X_j) . \tag{1}$$

The gain of the information can be expressed through use of the entropy, namely

$$g(X_i) = H(X_m) - H_S(X_m) , \tag{2}$$

where $H_S(X_m)$ is the mean application of the information and can be computed as a weighted sum of entropy for single subsets of attributes according to the possible partition of S. $H(X_m)$ is the entropy before partition of S and can be defined as

$$H(X_m) = -\sum_j p_j \times \log_2(p_j) , \tag{3}$$

where p_j is the probability of relation instance $\mathbf{X}$.

3 The Decision Tree Construction over Data Streams

3.1 The Study of Data Items Distribution and the Sample Size Establishment

The first problem which must be solved in the data stream analysis is associated with the data items distribution. We propose here the method for obtaining the limiting distribution of items occurring in the data streams. We recall that the application of limiting distributions is based on the assumption about the size of sample n. An open question is how large n must be in order for a limiting distribution.

Let $X_1^j, X_2^j, \ldots, X_n^j$ be independent random items from the data stream j with the uniform distribution. Let us assume that we have a moment of dimensionality k, namely $m_k^j = E(X^{j,k})$. Thus, we have the following statistics

$$A_k^j = \frac{1}{n} \sum_{r=1}^{n} X_r^k . \tag{4}$$

Then, it is easily seen that

$$E(A_k^j) = \frac{1}{n} \sum_{r=1}^{n} E(X_r^{j,k}) . \tag{5}$$

Since for $r = 1, 2, \ldots, n$ is satisfied $E(X_r^{j,k}) = E(X^{j,k})$, we may obtain

$$E(A_k^j) = \frac{1}{n} \sum_{r=1}^{n} E(X_r^k) = m_k . \tag{6}$$

We define statistics A_k^j as the moment of dimensionality k from the sample of the jth data stream.

According to the Khintchine's theorem [9] we can get the result that for each k the sequence $\{A_k^j\}$ of the moments of dimensionality k from the sample of the jth data stream is statistically converged to moment m_k^j from the general population, when $n^j \to \infty$. From the Slucki's theorem [5], the central moments $\{B_k^j\}$ from the sample are defined by

$$B_k^j = \frac{1}{n^j} \sum_{r=1}^{n^j} (X_r^j - A_r^j)^k , \tag{7}$$

where $n^j \to \infty$ is stochastically converged to the central moment μ_k^j from the general population. Thus, the variance of sample by $n^j \to \infty$ is stochastically converged to the variance of the general population.

Assuming that the finite moment m_{2k}^j exists with dimensionality $2k$ of a random variable X^j, we can get the variance of the random variable A_k^j. Firstly, for dimensionality $r = 1, 2, \ldots, n$ we obtain the variance of the random variable $X_r^{j,k}$, namely

$$\mathrm{var}(X_r^{j,k}) = E(X^{j,2k}) - [E(X^{j,k})]^2 = m_{2k}^j - (m_k^j)^2. \tag{8}$$

Through the independence of variables $X_1^j, X_2^j, \ldots, X_n^j$ this gives the following dependence

$$\text{var}(A_k^j) = \frac{1}{(n^j)^2} \sum_{r=1}^{n^j} \text{var}(X_r^{j,k}) = \frac{n^j}{(n^j)^2} \text{var}(X^{j,k}) = \frac{m_{2k}^j - (m_k^j)^2}{n^j}. \tag{9}$$

From the theorem of Leavy-Lindberg the sequence of distribution functions of standarized random variables can be obtained for $n^j \to \infty$, namely

$$Y_n^j = \frac{\sum_{r=1}^{n^j} X_r^{j,k} - n^j \times m_k^j}{\sqrt{n^j \times \text{var}(X^{j,k})}} = \frac{A_k^j - m_k^j}{\sqrt{m_{2k}^j - (m_k^j)^2}} \sqrt{n^j}. \tag{10}$$

This is convergent with the limiting distribution function $\Phi^j(X)$. Let us note that for the finite moment m_{2k}^j of the general population the distribution of moment A_k^j with dimensionality k from the sample is asymptotically normal:

$$N\left(m_k^j; \sqrt{\frac{m_{2k}^j - (m_k^j)^2}{n^j}}\right).$$

Finally, the sample size n^j of the jth data stream can be calculated as follows. Let the probability that the data from the sample indicates the defined mark (stamp) be equal to α.

   Thus, assuming that A_k^j has an asymptotic normal distribution

$$N\left(m_k^j; \sqrt{\frac{m_{2k}^j - (m_k^j)^2}{n^j}}\right)$$

it seeks the probability $P(|A_k^j - \alpha| \geq \varepsilon)$, where ε is a defined constant. In a practical sense it is important to assume here that ε is the difference between the greatest gain of information and the least gain of information or, in other words, the greatest possible entropy reduction to be obtained. With the help of the table of normal distribution, we can find the probability $F^j = P(|A_k^j - \alpha| \geq \varepsilon)$. Thus, the sample size n^j for the jth data stream is given by

$$n^j = \frac{(F^j)^2}{m_{2k}^j - (m_k^j)^2}. \tag{11}$$

3.2   The Decision Tree Induction from the Data Stream

Algorithms for the induction of decision trees from the data streams are based on the scalable approach. As one of the most effective induction methods, we used a

technique called Rain-Forest [6]. It adapts the main memory and can be used with very large training sets of a million of tuples. The method uses an AVC-set (where AVC stands for 'Attribute-Value, Classlabel') for each attribute, at each tree node, describing the training tuples at the node. The main idea of this method is based on the AVC-set creation for attribute at the node. For the exemplary data given in Table 1, the AVC-sets describing this data are presented in Table 2.

Table 1 An exemplary class-labeled training tuples

Identifier	Age	Cash_in_hand	Credit_rating	Has_credit
1	youth	low	weak	no
2	middle_aged	high	fair	yes
3	senior	low	weak	no
4	middle_aged	medium	fair	yes
5	youth	high	fair	yes
6	youth	low	weak	no
7	midle_aged	low	weak	no
8	senior	medium	weak	no
9	senior	high	fair	yes
10	middle_aged	low	fair	yes

Table 2 The AVC-sets describing the data of Table 1

Age	Has_credit	
	yes	no
youth	1	3
middle_aged	3	1
senior	1	2

Cash_	Has_credit	
in_hand	yes	no
low	1	4
medium	1	1
high	3	0

Credit_	Has_credit	
rating	yes	no
fair	5	0
weak	0	5

Let S be the set of training instances that allow for the choice of attribute a to build the decision tree. Let S_0 be a subset of training instances for which the values of attribute a are not defined. We assume that the measure of the information gain be

$$g(S \mid a) = \frac{\mid S - S_0 \mid}{\mid S \mid} g(S - S_0 \mid a). \tag{12}$$

If attribute a was selected for the test at node N, the subset of the training set S_0 is partitioned. Thus, we assign the weight responding to the probability of appearance defined by the attribute of value a. All the probabilities are computed on the ground

of the frequency of appearance of different values of the attribute value among the instances at the node. For example, for the subset S_j the weight is given by

$$w_j = \frac{|S_j|}{|S - S_0|} \; . \tag{13}$$

3.3 An Algorithm for Constructing the Decision Trees for Multiple Data Streams

The presented algorithm is two passes of the sample. In the first pass fixed the size of sample is fixed. In the second pass histograms are created for all the established class labels, taking into consideration attributes of each class label. The use of histograms to estimate the number of the established value of attributes is one of the techniques applied in the analysis of data stream, described a.o. by Haas et al. [7]. Further, we compute for each class label relevant attributes. It is the selected class label for which the largest gain of information was found. With the help of (13) we compute the weight for each value of the attribute in the current set of instances. On the basis of the weights for particular values of the attribute the ones with the largest weight is determined. It stands on a leaf of the constructed decision tree for which it forms a rule.

Further, in each pass the valuation of the constructed decision tree is performed using the error ratio computation. We assumed that the error ratio is equal to the the percentage of the falsely classified tuples by the constructed tree. If in succeeding pass the next sample of the error ratio has less value than the previously computed error ratio, according to the Hulton method [8] it is replaced.

Three conditions as the stop criterion were accepted:

1. the current set of intances is empty,
2. all the instances belong to the same class,
3. the set of the training examples is empty.

If one or more of the above given conditions occur, the decision tree is stopped. The algorithm of the decision tree construction for multiple data streams is given in Fig. 1.

4 Experimental Results

The data sets we used for our experiments were obtained from the Physiological Data Modeling Contest [2] held as a part of the International Conference on Machine Learning, 2004. These data were obtained from the observations collected with the use of the BodyMedia [1] wearable body sensors.

In our experiments, two streams were created from two separate sets. The first had 50 000 instances and seven class labels and the second had 32 673 instances and

```
procedure mining_streaming_data;
begin
   for j = 1 to J do
   while not stop_condition_satisfied do
      if first_pass(j) then
         find_the_cardinality(n(j)));
         determine_classlabel(k(j));
      endif;
      if second_pass(j) then
         for each k(j)) compute histogram (h(k, j)); endfor;
         find g := max[g(h, k(j))];
         use g to get w(k, j);
         find a = max[w(k, j)];
         form the rule of partition in the leaf;
         construct(current_tree);
      endif;
      tree_pruning;
      compute_error_ratio(current_tree);
      if (error_ratio(current_tree) - error_ratio(previous_tree) < ε) then
         replace(previous_tree_with_current_tree);
         else decision_tree(previous_tree);
      endif;
   endfor;
end;
```

Fig. 1 Pseudocode of algorithm for decision tree construction on streaming data

four class labels. The continuously-valued attributes were discretized with the help of the WEKA software.

For given data two data streams were built: the first one created for testing the transverse accelerometr reading, and the second one for studying the heat flux in various situations of human motion. The first data stream was characterized by the following attributes: age, near body temperature, pedometer indication, longitudinal accelerometer. For the second data stream we used such attributes as follows: age, heat flux, near body temperature, skin temperature. With the help of (11) and assuming that the value of parameter α is equal to 0.6 were determined the sizes of the sampled training data for both streams. The sample sizes are 136 and 125 for the first and the second stream, respectively.

The decision tree built for the first stream allows us to appreciate human motion for the indication of the transverse accelerometer in dependence on human's age, near body temperature, etc. The second decision tree formed for the second data stream determined the heat flux during the motion under various parameters. For both data streams seven and five label classes were determined, respectively. Histograms for established label classes were used to compute the information gain. These values were applied for the decision trees construction. The observed change in the trees contained change of the total number of nodes between three and eleven. All the trees had an assumed error of less than 10%. Figure 2a illustrates an
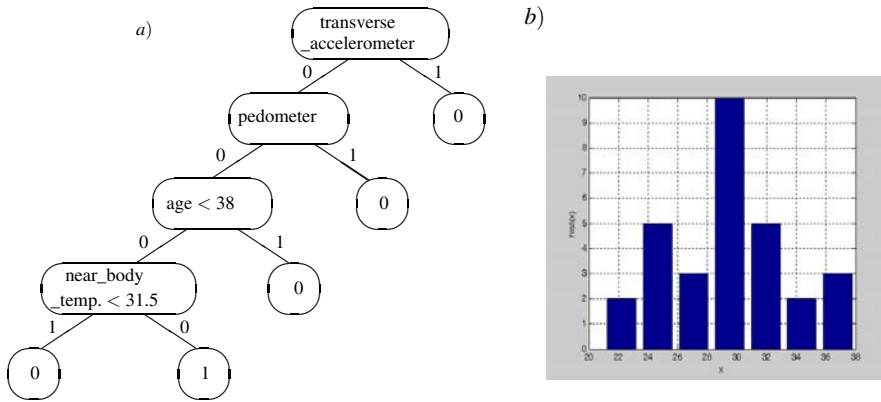
Fig. 2 (a) An exemplary decision tree built for first data stream. **(b)** Histogram for established label classes

exemplary decision tree built for first data stream. The corresponding histogram for established label classes is given in Fig. 2b.

5   Conclusions

This paper has focused on the construction of the decision tree over multiple data streams. This includes the study of data stream distribution and the sample size for a given data stream finding, as well as a decision tree algorithm over multiple streams building. We have developed a technique of the error ratio minimization, which is a necessary condition for obtaining the guaranteed quality classification.

References

1. Bodymedia: Home page (2004), http://www.bodymedia.com/index.jsp
2. Bodymedia: Physiological data modeling contest (2004),
   http://www.cs.utexas.edu/users/sherstov/pdmc
3. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Chapman & Hall/CRC Press, Boca Raton (1984)
4. Cranor, C., Johnson, T., Spatascek, O., Shkapenyuk, V.: Gigascope: A Stream Database for Network Applications. SIGMOD (2003)
5. Fisz, M.: Probability Theory and Mathematical Statistics. John Wiley & Sons, Chichester (1963)
6. Gehrke, J., Ramakrishnan, R., Ganti, V.: Rainforest - a framework for fast decision tree construction of large datasets. In: Proceedings of the 24th International Conference on Very Large Databases, pp. 416–427 (1998)
7. Haas, P., Naughton, J., Seshadri, P., Stokes, L.: Sampling-based estimation of the number of distinct values of an attribute. In: Proceedings of the International Conference on Very Large Databases, pp. 311–366 (1995)

8. Hulton, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 97–106 (2001)
9. Khintchine, A.Y.: Mathematical Methods in the Theory of Queueing. Griffin, London (1960)
10. Quinlan, J.R.: Introduction of decision trees. Machine Learning 1, 81–106 (1986)
11. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kauffman, San Francisco (1993)
12. Yao, Y., Gehrke, J.: Query processing in sensor networks. In: Proceedings of the Biennial Conference on Innovative Data Systems Research (2003)
13. Zhu, Y., Shasha, D.: StatStream: Statistical monitoring of thousand of data streams in real time. In: Proceedings of the 28th International Conference on Very Large Databases (2002)

Decision Tree Induction Methods
for Distributed Environment

Krzysztof Walkowiak and Michał Woźniak

Abstract. Since the amount of information is rapidly growing, there is an over-whelming interest in efficient distributed computing systems including Grids, public-resource computing systems, P2P systems and cloud computing. In this paper we take a detailed look at the problem of modeling and optimization of network computing systems for parallel decision tree induction methods. First, we present a comprehensive discussion on mentioned induction methods with a special focus on their parallel versions. Next, we propose a generic optimization model of a network computing system that can be used for distributed implementation of parallel decision trees. To illustrate our work we provide results of numerical experiments showing that the distributed approach enables significant improvement of the system throughput.

Keywords: decision tree, parallel machine learning, distributed computing.

1 Introduction

Problem of pattern recognition is accompanying our whole life. We start to learn how to recognize simple objects like 'dog', 'flower', 'car' when we are young and more sophisticated ones when we are growing up. Therefore methods of automatic classification is one of the main trend in Artificial Intelligence. The aim of such task is to classify the object to one of the predefined categories, on the basis of observation of its features [7]. Such methods are applied to the many practical areas like prediction of customer behavior, fraud detection, medical diagnosis etc. Numerous approaches have been proposed to construct efficient, high quality classifiers like neural networks, statistical and symbolic learning [15]. Among the different

Krzysztof Walkowiak · Michał Woźniak
Chair of Systems and Computer Networks, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail:{krzysztof.walkowiak,michal.wozniak}@pwr.wroc.pl

concepts and methods of machine learning, a decision tree induction is both attrac-
tive and efficient. To effectively deal with huge databases, we need time-efficient
parallel decision tree induction methods which can use a distributed network envi-
ronment. Such systems have been gaining much attention in recent years due to the
growing needs for various kinds of excessive computations related for instance to:
financial modeling, medical data analysis, experimental data acquisition, earthquake
simulation, and climate/weather modeling, astrophysics and many others [8, 16, 22].
In this paper we propose a new Integer Programming optimization model related to
network computing systems with a special focus on parallel decision tree induction
methods.

2   Related Works

A main aim of each pattern recognition algorithm is to construct a classifier which is
able to assign object to appropriate class on the basis of the observed features. There
are many methods of training classifiers. They solve the optimization problem how
to teach an efficient classifier on the basis of learning set, which consists of elements
representing feature values and corresponding class. In this case 'efficient' means
high-quality (i.e., classifier which average cost of misclassification is the lowest),
cheap in exploitation (i.e., cost of feature acquisition is small) and cheap in con-
struction (i.e., cost of it learning, like time, is small also). From this point of view
decision tree induction methods are very attractive approach which have been used
for several years [15]. Mentioned methods propose an approximation discrete func-
tion method which is adopted to the classification task. Many decision-tree algo-
rithms have been developed. The most famous are CART [3], ID3 and its extension
C4.5 [19]. ID3 introduces information entropy as the splitting attribute's choosing
measure. It trains a tree from root to leaf, a top-down sequence. The central choice
in the ID3 algorithm is selecting 'the best' attribute (which attribute to test at each
node in the tree). The proposed algorithm uses the information gain that measures
how well the given attribute separates the training examples according to the target
classification. As we mentioned above the C4.5 algorithm is an extended version
of ID3. It improves appropriate attribute selection measure, avoids data overfitting,
reduces error pruning, handles attributes with different weight, improves computing
efficiency, handles missing value data and continuous attributes, and performs other
functions. C4.5 instead of information gain in ID3 use an information gain ratio
[19]. Thanks that concept we can obtain so-called univariate decision tree, which
tests only single attribute in each node.

The second approach to decision tree construction is a multivariate decision tree
which offers simper structure than univariate one. We can suppose that simple struc-
ture is less susceptible to overfitting. Moreover univariate decision tree induction
uses greedy search methods. As a criterion local discriminant power is used. In [5]
Cover proved that for the set of attributes the best pair of features can not consist of
two best two independent features. It could consists of two another features. There
are many propositions how to construct multivariate decision tree. Some of them

suggests to use classifier in each node, e.g., LMDT uses linear classifier [4], in [12] authors propose to use Bayesian one. Interesting approaches was presented in [13], where proposition called LMT uses ID3 algorithm for discrete features and then linear regression for the rest set of features. Another approach to multivariate decision tree induction suggests to use traditional or heuristic feature selection methods in each node instead of evaluation of discriminant power of features [2, 6].

3   Parallelization of Decision Tree

Quinlan notes that computational complexity of ID3 (for discrete attribute) at each node of tree is $O(N_{LS} N_A)$, where N_{LS} is number of examples in learning set and N_A is number of attributes in the node. For continuous attributes the computational complexity is over quadratic in the size of learning set [17]. For such case to speed examination of the candidates, ID3 sorts examples using the continuous attributes as the sort key. The computational complexity of mentioned operation is $O(N_{LS} \log_2 N_{LS})$, what needs very long time for large dataset. Another time-consuming problem is decision tree pruning, which protects decision tree from overtraining. Its computational complexity is hard to estimate because it depends on decision tree size. To effectively deal with huge databases, we need time-efficient parallel decision tree induction methods which can use a distributed network environment. There are some proposition of parallel decision tree algorithm. SLIQ [14] and its extension SPRINT [20] used a pre-sorting technique in tree-growing phase and proposed new pruning procedures. In [11] data distributed parallel formulation of C4.5 was shown. Author used only frequency statistics from data to choose the best attribute. Parallel algorithm SPIES of decision tree induction with almost linear speedup was presented in [10]. In [21] synchronous and partitioned decision tree induction algorithms were shown. Additionally authors compared these propositions and formulate hybrid algorithm. The interesting research was presented in [23] where authors proposed parallel versions of univariate decision tree algorithm:

- Model-driven (node-based) parallelization is the most natural strategy of decision tree induction algorithms. Each node of tree represented by fine described terms of task (such as data subset, attributes used in this point, etc.) can be placed into queue and picked up by busy processor.
- Data-driven parallelization is based on an idea that training set is partitioned into subsets associated with separate processors. The part of decision tree induction algorithm is defined by master processor and realized by processor owning data subset necessary in this step. Data-driven decision tree algorithm parallelization is the most natural method in environments, where learning set is divided into several subsets dislocated physically. Main idea is to do as many computations locally as possible and exchange a small amount of partial results data. In real problems two cases could be observed:

  - distribution by samples,
  - distribution by features.

Experimental results of experiments which evaluated dependencies between speedup and number of processors were shown in cited work also. Mentioned algorithms concentrated their attention on constructing decision tree for a given learning set. If new data is coming the algorithms have to start from the beginning, because structure of decision tree is hard to modify. For many practical problems (where databases grow slow) this feature of methods is not disadvantage but for fast grooving database it could be a problem. Very interesting proposition of parallel decision tree algorithm for streaming data could be found in [1]. Proposed method builds a decision tree using horizontal parallelism on the ba-sis of on-line method for building histograms from coming data. These histograms are used to create new nodes of tree. Authors of mentioned paper showed that classification error of their proposition for distributed version of decision tree induction was slightly worse than original one, but value of error bound was accept-able for practical implementation. Some interesting observations and useful tips for decision tree construction on the basis of streaming data also could be found in [10]. When decision tree algorithms are applied to huge database they require significantly more computations. If we want to use multivariate decision tree induction algorithms in this case then numbers of computations dramatically grow. This observation caused that an idea of parallel decision tree induction algorithm was introduced early [20]. Another reason of parallelization needs may be data distribution and cost (usually time) of data relocation [11].

4 Model of Network Computing System

In this section we present an optimization model of a network computing system. Assumptions of the model follow from previous paper on these subject and real network computing systems. The model is generic, i.e., various network computing systems, e.g., grids, public resource computing systems fit to our model. The network computing system consists of clusters – represented as nodes $v = 1, 2, \ldots, V$ – connected to the overlay network. Each node is described by the download and upload capacity denoted as d_v and u_v, respectively. The maximum processing rate of node v, i.e., the number of uniform computational tasks that node v can calculate in one second is denoted as p_v. Furthermore, we are given v – the processing cost of one computational uniform task in node v. The transfer cost between nodes w and v is denoted by ξ_{wv}. In the network computing systems a set of computational projects $r = 1, 2, \ldots, R$ are to be computed. Each project described by the following parameters. The number of uniform computational tasks in project r is denoted by n_r. Each project has a source node that produces the input data and one or more destination nodes that wants to receive the output data, i.e., results of computations. For simplicity we assume that the uniform task for each project has the same computational requirement expressed in FLOPS. However, the values of the input and the output data transmit rate are specific for each computational project following from particular features of the project. Constants a_r and b_r denote the transmit rate of input data and output data, respectively, per one task in project r and are given in b/s. The

workflow of computational tasks is as follows. The input data is transferred from the source node to one or more computing nodes that processes the data. Next, the output data is sent from the computing node to one or more destination nodes. The input and the output data associated with the project is continuously generated and transmitted. Thus, computational and network resources can be reserved in the system according to offline optimization. To formulate the problem we use the notation proposed in [18].

Problem 1 (Network Computing System Cost Problem (NCSCP))

- Indices:

  - $v, w = 1, 2, \ldots, V$ overlay nodes (peers),
  - $r = 1, 2, \ldots, R$ projects.

- Constants:

  - p_v – maximum processing rate of node v (number of computational tasks that node v can calculate in one second),
  - d_v – download capacity of node v (b/s),
  - u_v – upload capacity of node v (b/s),
  - n_r – number of tasks in project r,
  - a_r – transmit rate of input data per one task in project r [b/s],
  - b_r – transmit rate of output data per one task in project r [b/s],
  - $s(r, v) = 1$ if v is the source node of project r; 0 otherwise,
  - $t(r, v) = 1$ if v is the destination node of project r; 0 otherwise,
  - Ψ_v – processing cost of one computational task in node v,
  - ξ_{wv} – transfer cost of 1 b/s from node w to node v,
  - M – large number.

- Variables:

  - x_{rv} – the number of tasks of project r calculated on node v (integer).

- Objective:

$$\min F = \Sigma_r \Sigma_v x_{rv} \Psi_v + \Sigma_r \Sigma_{w:s(r,w)=1} \Sigma_{v:v \neq w} a_r x_{rv} \xi_{wv} + \tag{1}$$
$$+ \Sigma_r \Sigma_{w:t(r,w)=1} \Sigma_{v:v \neq w} b_r x_{rv} \xi_{vw}$$

subject to

$$\Sigma_r x_{rv} \leq p_v, \qquad\qquad v = 1, \ldots, V, \tag{2}$$
$$\Sigma_r (1 - s(r,v)) a_r x_{rv} + \Sigma_{r:t(r,v)=1} b_r (n_r - x_{rv}) \leq d_v, \quad v = 1, \ldots, V, \tag{3}$$
$$\Sigma_{r:s(r,v)=1} a_r (n_r - x_{rv}) + \Sigma_r (t_r - t(r,v)) b_r x_{rv} \leq u_v, \quad v = 1, \ldots, V, \tag{4}$$
$$\Sigma_v x_{rv} = n_r, \qquad\qquad r = 1, \ldots, R. \tag{5}$$

The objective (1) is the cost of the system compromising the computing cost and the transfer cost. Since each node has a limited processing speed (power) dedicated to computations of the considered job, we add the constraint (2). Conditions (3) and

(4) are download and upload capacity constraints, respectively. (5) assures that for each project $r = 1, 2, \ldots, R$ all task are assigned for computing.

5 Results

The NCSCP problem is an NP-complete Integer Programming problem, since it can be reduced to the knapsack problem. To obtain the optimal solution we applied the CPLEX 11.0 solver [9] including effective branch-and-cut algorithm. We created several network computing systems consisting of 50 computing nodes. The processing limit and access link capacity of each node were generated at random or according to assumptions of our experiments. We created also a number of project sets, consisting of 30 projects. Parameters of these projects (i.e. number of tasks, transmit rate of input and output data rate, source and destination nodes) were selected randomly or according to assumptions of our experiments. An important feature of decision tree algorithms and many other algorithms related to machine learning methods is that the input data is much larger than the output data. Therefore, we assume that in a part computational projects the transmit rate of input data (constant a_r) is set to be much larger than the transmit rate of output data (constant b_r). In other projects the transmit rate of input and output data is selected at random. Let APR denote the asymmetric project ratio in the overall number of projects. For instance, if APR = 0.33, then 10 of 30 projects are asymmetric and 20 of 30 projects are other. We generated project sets with the following values of APR: 0.33, 0.67, and 1.00.

The left graph in Fig. 1 plots the cost as a function of node processing limit. Each curve in the graph represents a project set with different value of APR parameter. We can watch that increasing the processing limit reduces the network cost and the function is generally linear. Moreover, if the APR parameter grows, the system cost decreases. The reason for this lies in the fact that more asymmetric projects means lower capacity consumptions and consequently lower costs related to network transmissions. The second graph in Fig. 1 presents the cost as a function of symmetric
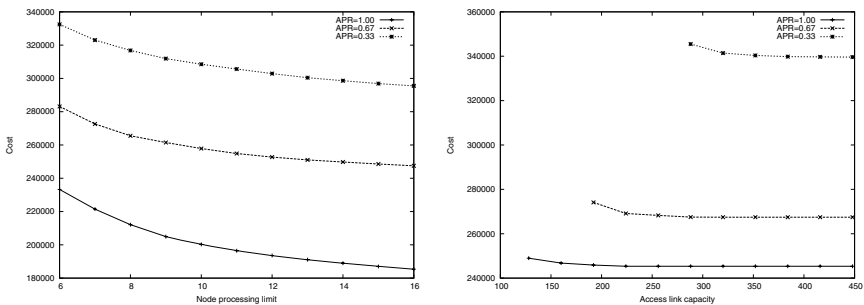
Fig. 1 The cost as a function of function of node processing limit and access link capacity

access link capacity. The values of processing limit are constant for each node in all tested cases, while the capacity of access links is increased consecutively by 32. Each curve in the graph represents a project set with different value of APR parameter. We can see that in the case of APR = 0.33 for small values of capacity (< 288), the project set cannot be process in the network computing system. Only in the case of APR = 1.00, the project set can be located in the computing system even for the smallest value of access link capacity. This follows from the fact that asymmetric projects have lower requirements of capacity.

Analysis of Fig. 1 indicates that increasing the link capacity enables reduction of the system cost, however the improvement is relatively low. In summary, the experiments show that the cost of the network computing systems depends on parameters related to both: computational projects and computing systems. Moreover, the processing limit have stronger influence on the system cost than the access link capacity.

6   Conclusions

Problems of decision tree induction in distributed computing environment have been presented in this work. Classifiers based on decision tree schema are both attractive and efficient but their computational complexity is high. In this paper we have studied how to model and optimize network computing systems for parallel decision tree induction methods. We have formulated a detailed Integer Programming model that reflects the workflow of parallel decision tree algorithm for streaming data. However, the presented model is generic and also other computational tasks fit to our approach. Results of experiments have shown that computations run in distributed environment using many nodes can reduce the system cost. The results of experiments and literature review of works connected with parallel decision tree induction methods encourage us to continue works on distributed methods of multivariate decision tree induction.

References

1. Ben-Haim, Y., Yom-Tov, E.: A streaming parallel decision tree algorithm. In: Proceedings of the PASCAL Workshop on Large Scale Learning Challenge, Helsinki, Finland (2008)
2. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. Artificial Intelligence 97(1-2), 245–271 (1997)
3. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth (1984)
4. Brodley, C.E., Utgoff, P.E.: Multivariate decision trees. Machine Learning 19(1), 45–77 (1995)

5. Cover, T.M.: The best two independent measurements are not the two best. IEEE Transactions on Systems, Man and Cybernetics 4(1), 116–117 (1974)
6. Dash, M., Liu, H.: Feature selection for classification. Intelligent Data Analysis 1(1-4), 131–156 (1997)
7. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Willey and Sons, New York (2001)
8. Foster, I., Iamnitchi, A.: On death, taxes and the convergence of peer-to-peer and grid computing. In: Kaashoek, M.F., Stoica, I. (eds.) IPTPS 2003. LNCS, vol. 2735, pp. 118–128. Springer, Heidelberg (2003)
9. ILOG: CPLEX 11.0. user's manual (2007)
10. Jin, R., Agrawal, G.: Communication and memory efficient parallel decision tree construction. In: Proceedings of the 3rd SIAM Conference on Data Mining, San Francisco, US, pp. 119–129 (2003)
11. Kufrin, R.: Decision trees on parallel processors. Parallel Processing for Artificial Intelligence 3, 279–306 (1997)
12. Kurzyński, M.: The optimal strategy of a tree classifier. Pattern Recognition 16(1), 81–87 (1983)
13. Landwehr, N., et al.: Logistic model trees. Machine Learning 95(1-2), 161–205 (2005)
14. Mehta, M., et al.: SLIQ: A fast scalable classifier for data mining. In: Proceedings of the 5th International Conference on Extending Database Technology, pp. 18–32. Avignon, France (1996)
15. Mitchell, T.M.: Machine Learning. McGraw-Hill Company, Incorporated, New York (1997)
16. Nabrzyski, J., Schopf, J., Węglarz, J.: Grid resource management: state of the art and future trends. Kluwer Academic Publishers, Boston (2004)
17. Paliouras, G., Bree, D.S.: The effect of numeric features on the scalability of inductive learning programs. In: Lavrač, N., Wrobel, S. (eds.) ECML 1995. LNCS, vol. 912, pp. 218–231. Springer, Heidelberg (1995)
18. Pióro, M., Medhi, D.: Routing, Flow, and Capacity Design in Communication and Computer Networks. Morgan Kaufman Publishers, San Francisco (2004)
19. Quinlan, J.R.: C4.5: Program for Machine Learning. Morgan Kaufman, San Mateo (1993)
20. Shafer, J., et al.: SPRINT: A scalable parallel classifier for data mining. In: Proceedings of the 22nd Conference on Very Large Databases, pp. 544–555 (1996)
21. Srivastava, A., et al.: Parallel formulations of decision tree classification algorithms. Data Mining and Knowledge Discovery 3(3), 237–261 (1999)
22. Taylor, I.: From P2P to Web services and grids: peers in a client/server world. Springer, Heidelberg (2005)
23. Yidiz, O.T., Dikmen, O.: Parallel univariate decision trees. Pattern Recognition Letters 28, 825–832 (2007)

Extensions of Multistage Decision Transition Systems: The Rough Set Perspective

Krzysztof Pancerz

Abstract. Multistage decision transition systems (MDTSs) were proposed to describe transitions among states observed in the given systems. If we are interested in sequences of changes of states, then we may represent such changes by means of polyadic transition relations over the sets of states. A given MDTS represents such a relation. Each object in MDTS is referred to as an episode. We can extend a given MDTS by adding new episodes to it. The important question is as follows: 'what is a consistency factor of a new episode added to MDTS with the knowledge included in MDTS?'. In the standard approach, if we wish to answer this question, then we compute the set of all minimal decision rules true in MDTS. Such a problem is NP-hard. In the paper, we present the answer to the question using a crucial notion of rough sets, namely a lower approximation of an appropriate set of episodes in MDTS. Moreover, we give some theoretical foundations of extensions of MDTS from the rough set point of view.

Keywords: MDTS, decision rules, consistency factor, NP-hard problem, rough approximation.

1 Introduction

Information systems can be used to represent the knowledge of the behavior of concurrent systems [6]. In this approach, an information system represented by a data table includes the knowledge of the global states of a given concurrent system *CS*.

Krzysztof Pancerz

Institute of Biomedical Informatics, University of Information Technology and Management, Sucharskiego 2, 35-225 Rzeszów, Poland

e-mail: `kpancerz@wsiz.rzeszow.pl`

and

Chair of Computer Science and Knowledge Engineering,

Zamość University of Management and Administration,

Akademicka 4, 22-400 Zamość, Poland

The columns of the table are labeled with names of attributes (treated as processes of *CS*). Each row labeled with an object (treated as a global state of *CS*) includes a record of attribute values (treated as local states of processes). In a general case, a concurrent system is a system consisting of some processes, whose local states can coexist together and they are partly independent. For example, we can treat systems consisting of economic processes, financial processes, biological processes, genetic processes, meteorological processes, etc. as concurrent systems. Dynamic information systems were proposed by Z. Suraj in 1998 [8] to represent the knowledge of states of concurrent systems and transitions between them. Transitions between states were described by binary transition relations. In this paper, we use a notion of multistage dynamic information systems [3]. These systems enable us to represent multistage transitions among states (called also episodes). Therefore, transitions among states are described by polyadic transition relations. To represent such relations multistage decision transition systems are used. We are especially interested in extensions (consistent and partially consistent) of multistage decision transition systems. A partially consistent extension of a given multistage decision transition system consists of new transitions among states which are totally consistent or consistent only to a certain degree (partially consistent) with the knowledge included in the original multistage decision transition system. The degree of consistency can be between 0 and 1, 0 for the total inconsistency and 1 for the total consistency. We assume that the knowledge included in multistage decision transition systems is expressed by transition rules, which are minimal decision rules understood from the rough set point of view. We use adequately defined lower approximations of sets of transitions in order to compute a degree of consistency of a given episode from any extension of a given multistage decision transition system. We can determine which transitions (episodes) in the original multistage decision transition system generate transition rules which are not satisfied by the tested episode from the extension. It is worth noting, that we do not calculate any transition rules in a multistage decision transition system. This is an important property from the computational complexity point of view, especially, if we have high dimensional data.

2 Rough Set Rudiments

First, we recall basic concepts of rough set theory (cf. [5, 7]) used in the paper.

A concept of an information system is one of the basic concepts of rough set theory. Information systems are used to represent some knowledge of elements of a universe of discourse. An *information system* is a pair $S = (U,A)$, where U is a set of *objects*, A is a set of *attributes*, i.e., $a : U \to V_a$ for $a \in A$, where V_a is called a value set of a. A *decision system* is a pair $S = (U,A)$, where $A = C \cup D$, $C \cap D = \emptyset$, and C is a set of *condition attributes*, D is a set of *decision attributes*. Any information (decision) system can be represented as a data table whose columns are labeled with attributes, rows are labeled with objects, and entries of the table are attribute values. For each object $u \in U$ in the information or decision system $S = (U,A)$, we define a signature of u by $inf_S(u) = \{(a,a(u)) : a \in A\}$.

Let $S = (U, A)$ be an information system. Each subset $B \subseteq A$ of attributes determines an equivalence relation on U, called an *indiscernibility relation Ind(B)*, defined as $Ind(B) = \{(u, v) \in U \times U : \forall_{a \in B} a(u) = a(v)\}$. The equivalence class containing $u \in U$ will be denoted by $[u]_B$.

Let $X \subseteq U$ and $B \subseteq A$. The *B-lower approximation* $\underline{B}X$ of X and the *B-upper approximation* $\overline{B}X$ of X are defined as $\underline{B}X = \{u \in U : [u]_B \subseteq X\}$ and $\overline{B}X = \{u \in U : [u]_B \cap X \neq \emptyset\}$, respectively.

With every information system $S = (U, A)$ we associate a formal language $L(S)$. Formulas of $L(S)$ are built from atomic formulas in the form (a, v), where $a \in A$ and $v \in V_a$, by means of propositional connectives: negation ($\neg$), disjunction ($\vee$), conjunction ($\wedge$), implication ($\Rightarrow$) and equivalence ($\Leftrightarrow$) in the standard way. The object $u \in U$ satisfies a formula ϕ of $L(S)$, denoted by $u \underset{S}{\models} \phi$ (or in short $u \models \phi$), if and only if the following conditions are satisfied: (1) $u \models (a, v)$ iff $a(u) = v$, (2) $u \models \neg\phi$ iff not $u \models \phi$, (3) $u \models \phi \vee \psi$ iff $u \models \phi$ or $u \models \psi$, (4) $u \models \phi \wedge \psi$ iff $u \models \phi$ and $u \models \psi$. As a corollary from the above conditions we get: (1) $u \models \phi \Rightarrow \psi$ iff $u \models \neg\phi \vee \psi$, (2) $u \models \phi \Leftrightarrow \psi$ iff $u \models \phi \Rightarrow \psi$ and $u \models \psi \Rightarrow \phi$. If ϕ is a formula of $L(S)$, then the set $|\phi|_S = \{u \in U : u \models \phi\}$ is called the meaning of formula ϕ in S.

A rule in the information system S is a formula of the form $\phi \Rightarrow \psi$, where ϕ and ψ are referred to as the predecessor and the successor of the rule, respectively. The rule $\phi \Rightarrow \psi$ is true in S if $|\phi|_S \subseteq |\psi|_S$. In our approach, we consider rules in the form $\phi \Rightarrow \psi$, where ϕ is a conjunction of atomic formulas of $L(S)$ and ψ is an atomic formula of $L(S)$. If $u \models (\phi \wedge \psi)$, then we say that the object u supports the rule $\phi \Rightarrow \psi$ or that the object generates the rule $\phi \Rightarrow \psi$. We say that the rule $\phi \Rightarrow \psi$ is satisfied by the object $u \in U$ (or the object $u \in U$ satisfies the rule $\phi \Rightarrow \psi$) if and only if $u \models (\phi \Rightarrow \psi)$. A rule is called minimal in S if and only if removing any atomic formula from ϕ results in a rule being not true in S. The set of all minimal rules true and realizable (i.e., such rules $\phi \Rightarrow \psi$ that $|\phi \wedge \psi|_S \neq \emptyset$) in S will be denoted by $Rul(S)$. By $Rul_a(S)$ we will denote the set of all rules from $Rul(S)$ having an atomic formula containing the attribute a in their successors.

3 Multistage Decision Transition Systems (MDTSs)

In general, a description of concurrent systems by means of information systems does not cover their dynamic behavior, i.e., an information system includes only the knowledge of global states observed in a given concurrent system. In [8], dynamic information systems have been proposed for a description of concurrent systems. A dynamic information system additionally includes information about transitions between global states observed in a given concurrent system. So, the dynamics is expressed by a transition relation defined in a dynamic information system and the term of a dynamic information system should be understood in this sense. Here, we give some crucial notions concerning dynamic information systems.

Definition 1. A *transition system* is a pair $TS = (U, T)$, where U is a nonempty set of states and $T \subseteq U \times U$ is a transition relation.

Definition 2. A *dynamic information system* is a tuple $DIS = (U, A, T)$, where $S = (U, A)$ is an information system called the *underlying system* of *DIS* and $TS = (U, T)$ is a transition system.

The underlying system includes global states of a given concurrent system whereas a transition system describes transitions between these global states.

   We can extend a notion of dynamic information systems to the so-called multistage dynamic information systems (in short, MDISs) [3] which uses the polyadic transition relation instead of the binary transition relation.

Definition 3. A *multistage transition system* is a pair $MTS = (U, T)$, where U is a nonempty set of states and $T \subseteq U^k$ is a polyadic transition relation, where $k > 2$.

Definition 4. A *multistage dynamic information system* is a tuple $MDIS = (U, A, T)$, where $S = (U, A)$ is an information system called the *underlying system* of *MDIS* and $MTS = (U, T)$ is a multistage transition system.

Each element of a multistage transition relation T in a multistage dynamic information system $MDIS = (U, A, T)$ is a sequence of global states (from the set U) which can be referred to as an episode.

Definition 5. Let $MDIS = (U, A, T)$ be a multistage dynamic information system, where $T \subseteq U^k$. Each element $(u^1, u^2, \ldots, u^k) \in T$, where $u^1, u^2, \ldots, u^k \in U$, is called an episode in *MDIS*.

A dynamic information system can be presented by means of data tables representing information systems in the Pawlak's sense (see [4]). In this case, each dynamic information system *DIS* is depicted by means of two data tables. The first data table represents an underlying system S of *DIS* that is, in fact, an information system. The second one represents a decision system that is further referred to as a decision transition system. This table represents transitions determined by a transition relation. Analogously, we can use a suitable data table to represent a multistage transition system. Such a table will represent the so-called multistage decision transition system.

Definition 6. Let $MTS = (U, T)$ be a multistage transition system. A *multistage decision transition system* is a pair $MDTS = (U_T, A^1 \cup A^2 \cup \cdots \cup A^k)$, where each $t \in U_T$ corresponds exactly to one element of the polyadic transition relation T whereas attributes from the set A^i determine global states of the ith domain of T, where $i = 1, 2, \ldots, k$.

Each object in a multistage decision transition system represents one episode in a given multistage dynamic information system. If k is fixed, we can talk about a k-adic transition relation, a k-stage transition system and a k-stage dynamic information system. For a given multistage decision transition system, we can consider its elementary decision transition subsystems defined as follows.

Definition 7. An *elementary decision transition subsystem* of a multistage decision transition system $MDTS = (U_T, A^1 \cup A^2 \cup \cdots \cup A^k)$ is a decision transition system $DTS(i, i+1) = (U_T, A^i \cup A^{i+1})$, where: $i \in \{1, \ldots, k-1\}$.

In an elementary decision transition subsystem, we can consider some rules called, in short, elementary transition rules which are, in fact, decision rules.

Definition 8. Let $DTS(i, i+1) = (U_T, A^i \cup A^{i+1})$ be an elementary decision transition subsystem. An *elementary transition rule* in $DTS(i, i+1)$ is a formula of a formal language $L(DTS(i, i+1))$ in the form $\phi|_{A^i} \Rightarrow \psi|_{A^{i+1}}$, where $\phi|_{A^i}$ and $\psi|_{A^{i+1}}$ are formulas of $L(DTS(i, i+1))$ restricted to the sets of attributes A^i and A^{i+1}, respectively.

4   Extensions of MDTSs

The extensions of dynamic information systems have been considered in [4, 9]. In this paper, we focus only on the extensions of multistage decision transition systems. Analogously to definition of extensions of information systems, we define an extension of a multistage decision transition system. So, any nontrivial extension of a given multistage decision transition system $MDTS = (U_T, A^1 \cup A^2 \cup \cdots \cup A^k)$ includes new episodes such that for each episode t^* we have $a(t^*) \in V_a$ for each $a \in (A^1 \cup A^2 \cup \cdots \cup A^k)$.

In this section, we are interested in computing a degree of consistency (called a consistency factor) of a given episode from any extension of a given multistage decision transition system $MDTS$ with the knowledge included in $MDTS$. We give a way of computing consistency factors. An approach proposed here does not involve computing any rules from $MDTS$.

Let $DTS(i, i+1) = (U_T, A^i \cup A^{i+1})$ be the elementary decision transition subsystem. For each attribute $a \in A^i$ and the new episode t^*, we can transform $DTS(i, i+1)$ into the system with irrelevant values of attributes. If $a(t^*) \neq a(t)$, where $t \in U_T$, then we replace $a(t)$ by the value * (denoting an irrelevant value). This means that we create a new system for which appropriate sets of attribute values are extended by the value *. The transformed system can be treated as an incomplete system. Therefore, instead of an indiscernibility relation and equivalence classes, we use a characteristic relation and characteristic sets (cf. [1]). For the transformed elementary decision transition subsystem $DTS(i, i+1) = (U_T, A^i \cup A^{i+1})$, we define a characteristic relation $R(A^i)$. $R(A^i)$ is a binary relation on U_T defined as follows:

$$R(A^i) = \{(t, v) \in U_T^2 : [\exists_{a \in A^i} a(t) \neq *] \wedge [\forall_{a \in A^i}(a(t) = * \vee a(t) = a(v))]\}.$$

For each $t \in U_T$, a characteristic set $K_{A^i}(t)$ has the form $K_{A^i}(t) = \{v \in U_T : (t, v) \in R(A^i)\}$. For any $X \subseteq U_T$, the A^i-lower approximation of X is determined as $\underline{A^i}X = \{t \in U_T : K_{A^i}(t) \neq \emptyset \wedge K_{A^i}(t) \subseteq X\}$.

Let $DTS(i, i+1) = (U_T, A^i \cup A^{i+1})$ be an elementary decision transition subsystem, $a \in A^{i+1}$, and $v_a \in V_a$. By $X_a^{v_a}$ we denote the subset of U_T such that $X_a^{v_a} = \{t \in U_T : a(t) = v_a\}$.

Let $DTS(i, i+1) = (U_T, A^i \cup A^{i+1})$ be an elementary decision transition subsystem, $DTS^*(i, i+1) = (U_T^*, A^i \cup A^{i+1})$ its extension, $t^* \in U_T^*$ a new episode from

DTS^*. The episode t^* satisfies each elementary transition rule in $DTS(i, i+1)$ if and only if $\forall_{a \in A^{i+1}} \forall_{v_a \in V_a} \left(\underline{A^i} X_a^{v_a} \neq \emptyset \Rightarrow a(t^*) = v_a \right)$.

Elementary transition rules (which are minimal and realizable in a given $DTS(i, i+1)$) are generated by episodes belonging to proper lower approximations. Each non-empty lower approximation includes episodes from $DTS(i, i+1)$ matched (at least partially) by a new episode with respect to attributes from the set A^i. Therefore, in order to satisfy proper elementary transition rules, a new episode has to have the same value of the attribute $a \in A^{i+1}$ as episodes belonging to the given lower approximation.

In order to compute a consistency factor of a given episode t^* from any extension of a given multistage decision transition system $MDTS = (U_T, A^1 \cup A^2 \cup \cdots \cup A^k)$ we create a family **DTS** of elementary decision transition subsystems, i.e., **DTS** $= \{DTS(i, i+1) = (U_T, A^i \cup A^{i+1})\}_{i=1,\dots,k-1}$. Next, the consistency factor $\xi_{DTS(i,i+1)}(t^*)$ of the episode t^* with the knowledge included in $DTS(i, i+1)$ is computed for each subsystem $DTS(i, i+1)$ from the family **DTS**. Finally, the consistency factor $\xi_{MDTS}(t^*)$ of the episode t^* with the knowledge included in $MDTS$ is calculated as:

$$\xi_{MDTS}(t^*) = \prod_{i=1}^{k-1} \xi_{DTS(i,i+1)}(t^*) \, .$$

The consistency factor $\xi_{DTS(i,i+1)}(t^*)$ of the episode t^* with the knowledge included in $DTS(i, i+1)$ is calculated as $\xi_{DTS(i,i+1)}(t^*) = 1 - \frac{\text{card}(\widetilde{U}_T)}{\text{card}(U_T)}$, where $\widetilde{U}_T = \bigcup_{a \in A^{i+1}} \bigcup_{v_a \in V_a} \{\underline{A^i} X_a^{v_a} : \underline{A^i} X_a^{v_a} \neq \emptyset \wedge a(t^*) \neq v_a\}$.

In the presented approach, computing a consistency factor for a given episode is based on determining importance (relevance) of rules extracted from the system $MDTS$ which are not satisfied by the new episode. We assume that if the importance of these rules is greater the consistency factor of a new episode with the knowledge included in $MDTS$ is smaller. The importance of rules is determined by the strength factor.

Example 1. As an example we take daily exchange rates between the Polish zloty and two currencies: US dollar (marked with u) and Euro (marked with e). Attributes correspond to currencies. The meaning of values of attributes is the following: -1 denotes decreasing a given exchange rate in relation to the previous exchange rate, 0 denotes remaining a given exchange rate on the same level in relation to the previous exchange rate, 1 denotes increasing a given exchange rate in relation to the previous exchange rate. To represent episodes (transitions among states) as sequences of three consecutive global states we build a multistage decision transition system $MDTS$ shown in Table 1a. We have five episodes $t_1, t_2, \dots, t_5$. We can say that attributes from the set A^1 determine global states in the time instant τ, attributes from the set A^2 determine global states in the time instant $\tau + 1$ and attributes from the set A^3 determine global states in the time instant $\tau + 2$.

In the multistage decision transition system $MDTS = (U_T, A^1 \cup A^2 \cup A^3)$, we can distinguish two elementary decision transition subsystems: $DTS(1, 2) = (U_T, A^1 \cup A^2)$ and $DTS(2, 3) = (U_T, A^2 \cup A^3)$. In the elementary decision transition

Table 1 **(a)** A multistage decision transition system *MDTS*. **(b)** A new episode

<div style="text-align:center">a)</div>

$U_T/A^1 \cup A^2 \cup A^3$	u^1	e^1	u^2	e^2	u^3	e^3
t_1	-1	-1	-1	-1	1	1
t_2	-1	-1	1	1	1	-1
t_3	1	1	1	-1	-1	1
t_4	1	-1	-1	1	-1	0
t_5	-1	1	-1	0	1	0

<div style="text-align:center">b)</div>

$U_T/A^1 \cup A^2 \cup A^3$	u^1	e^1	u^2	e^2	u^3	e^3
t^*	-1	-1	-1	0	1	1

subsystem $DTS(1,2)$, we have, for example, the following elementary transition rules: $(u^1, -1) \wedge (e^1, -1) \Rightarrow (u^2, -1)$, $(u^1, -1) \wedge (e^1, -1) \Rightarrow (e^2, -1)$ which are minimal, true and realizable. In the elementary decision transition subsystem $DTS(2,3)$, we have, for example, the following elementary transition rules: $(e^2, 0) \Rightarrow (e^3, 0)$, $(u^2, -1) \wedge (e^2, -1) \Rightarrow (e^3, 1)$ which are minimal, true and realizable.

Suppose we are given a new episode (multistage transition) presented in Table 1b. We are going to determine a degree of the possibility of appearance of this episode in our system. For the elementary decision transition subsystem $DTS(1,2)$, there do not exist lower approximations defined earlier. Therefore, the episode t^* satisfies all minimal transition rules true and realizable in the elementary decision transition subsystem $DTS(1,2)$. Hence, $\xi_{DTS(1,2)}(t^*) = 1$, i.e., the episode t^* is consistent to the degree 1 (or consistent in short) with the knowledge included in the elementary decision transition subsystem $DTS(1,2)$. In case of the elementary decision transition subsystem $DTS(2,3)$, we have $\underline{A^2}X_{e^3}^0 = \{t_5\}$ and $e^3(t_5) = 0$, but $e^3(t^*) = 1$. So, the episode t^* does not satisfy all minimal transition rules true and realizable in $DTS(2,3)$. The set $\widetilde{U}_T$ of episodes from $DTS(2,3)$ generating rules not satisfied by t^* consists of the episode t_5. Therefore, $\xi_{DTS(2,3)}(t^*) = 0.8$. Finally, we obtain the consistency factor $\xi_{MDTS}(t^*)$ of the episode t^* with the knowledge included in *MDTS* which is $\xi_{MDTS}(t^*) = \xi_{DTS(1,2)}(t^*) \times \xi_{DTS(2,3)}(t^*) = 0.8$. According to our approach we can say that the episode t^* is possible to appear to the degree 0.8 with respect to the knowledge included in the original multistage decision transition system *MDTS*.

5   Conclusions

In the paper, we proposed a new method of computing consistency factors of new episodes added to multistage decision transition systems. This method uses lower approximations of proper sets of episodes. Consistency factors can be useful to predicting degrees of possibilities of appearing episodes in the future. Important task for further research is to propose other ways of knowledge representation, and what follows, other ways of computing consistency factors. It is also necessary to examine our approach on real-life data.

References

1. Grzymała-Busse, J.W.: Data with missing attribute values: Generalization of indiscernibility relation and rule induction. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Świniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 78–95. Springer, Heidelberg (2004)
2. Moshkov, M., Skowron, A., Suraj, Z.: Maximal consistent extensions of information systems relative to their theories. Information Science 178, 2600–2620 (2008)
3. Pancerz, K.: Extensions of dynamic information systems in state prediction problems: the first study. In: Magdalena, L., et al. (eds.) Proceedings of the Conference on Information Processing and Management of Uncertainty, Malaga, Spain, pp. 101–108 (2008)
4. Pancerz, K., Suraj, Z.: Rough sets for discovering concurrent system models from data tables. In: Hassanien, A.E., et al. (eds.) Rough Computing – Theories, Technologies and Applications, Information Science Reference, Hershey, pp. 239–268 (2008)
5. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, Dordrecht (1991)
6. Pawlak, Z.: Concurrent versus sequential: the rough sets perspective. Bulletin of the EATCS 48, 178–190 (1992)
7. Pawlak, Z.: Some issues on rough sets. 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. 1–58. Springer, Heidelberg (2004)
8. Suraj, Z.: The synthesis problem of concurrent systems specified by dynamic information systems. In: Polkowski, L., et al. (eds.) Rough Sets in Knowledge Discovery, vol. 2, pp. 418–448. Physica-Verlag, Heidelberg (1998)
9. Suraj, Z., Pancerz, K.: Some remarks on computing consistent extensions of dynamic information systems. In: Kwasnicka, H., Paprzycki, M. (eds.) Proceedings of the Conference on Intelligent Systems Design and Applications, pp. 420–425. Wrocşaw, Poland (2005)
10. Suraj, Z., Pancerz, K.: A new method for computing partially consistent extensions of information systems: A rough set approach. In: Proceedings of the Conference on Information Processing and Management of Uncertainty, Paris, France, pp. 2618–2625 (2006)
11. Suraj, Z., Pancerz, K.: Towards efficient computing consistent and partially consistent extensions of information systems. Fundamenta Informaticae 79(3-4), 553–566 (2007)

Emotion Recognition Based on Dynamic Ensemble Feature Selection

Yong Yang, Guoyin Wang, and Hao Kong

Abstract. Human-computer intelligent interaction (HCII) is becoming more and more important in daily life, and emotion recognition is one of the important issues of HCII. In this paper, a novel emotion recognition method based on dynamic ensemble feature selection is proposed. Firstly, a feature selection algorithm is proposed based on rough set and domain-oriented data-driven data mining theory, which can get multiple reducts and candidate classifiers accordingly. Secondly, the nearest neighborhood of each unseen sample is found in a validation subset and the most accuracy classifier is selected from the candidate classifiers. In the end, the dynamically selected classifier is used to recognize each unseen sample. The proposed method is proved to be an effective and suitable method for emotion recognition according to the result of comparative experiments.

Keywords: emotion recognition, ensemble feature selection, rough set, domain oriented data driven data mining (3DM).

1 Introduction

In recent years there has been a growing interest in improving all aspects of interaction between humans and computers. It is argued that in order to truly achieve effective human-computer intelligent interaction (HCII), there is a need for computer to be able to interact with user naturally, similar to the way of human-human

Yong Yang · Guoyin Wang · Hao Kong
Institute of Computer Science & Technology,
Chongqing University of Posts and Telecommunications,
Chongqing, 400065, P.R. China
e-mail:{yangyong,wanggy}@cqupt.edu.cn,
e-mail: everystop@163.com

Yong Yang
School of Information Science and Technology, Southwest Jiaotong University,
Chengdou, 610031, P.R. China

interaction. HCII is becoming more and more important in such applications as smart home, smart office and virtual reality, and it will be popular in all aspects of daily life in the future [7]. To achieve the purpose of HCII, it is important for computers to recognize people's emotion states and give suitable feedback [6]. Consequently, emotion recognition is becoming a hot research topic in both industry and academic. There have been a lot of research works in this field in recent years and there have been some successful products such as the amazing robots produced in Japan. But there is still a long way to achieve a computer acting as a human since there are many problems unsolved in psychology and cognitive theories.

Usually, emotion recognition is studied with such methods as ANN, fuzzy sets, SVM, HMM, Rough Sets, etc, and the recognition rate often arrives at 64% to 98% [5]. Meanwhile, new methods are always pursued for emotion recognition for better results.

Ensemble learning has been a hot research topic in machine learning since 1990s' [2]. Ensemble methods are learning algorithms that construct a set of candidate classifiers and then classify new objects by integrating the prediction of the candidate classifiers. An ensemble system is often much more accurate than each candidate classifier. As a popular machine learning method, ensemble methods are often used in pattern recognition, network security, medical diagnosis, etc. [1, 3]. Bsides the popular ensemble methods as bagging [1], boosting [3]. Ensemble feature selection (EFS) is also a classical ensemble method. It takes different feature subset as the input features for a candidate classifier construction.

In this paper, a novel emotion recognition method based on EFS and rough set theory is proposed. Firstly, a novel feature selection algorithm is proposed based on rough set and domain_oriented data_driven data mining (3DM) theory, which can get multiple reducts and candidate classifiers accordingly. Secondly, the nearest neighborhood of each unseen sample is found in a validation subset and the most accuracy classifier is selected from the candidate classifiers. At last, the dynamically selected classifier is used to classify each unseen sample. The proposed method is proved to be efficient by experiment results.

The rest of this paper is organized as follows. In Sect. 2, a novel emotion recognition method based on dynamic ensemble feature selection is proposed. Simulation experiments and discussion are introduced in Sect. 3. Finally, conclusion and future works are discussed in Sect. 4.

2   Emotion Recognition Method Based on Dynamic Selection Ensemble Learning

2.1   Tolerance Relation Model for Continuous Value Information System

In this section, reduction algorithms of rough set theory are used for an emotion recognition method based on EFS. Since the facial features are measured and

continuous values. There should be a discretization process according to traditional rough set theory. Unfortunately, information might be lost or changed in the discretization process and the result would be affected. To solve this problem, a novel feature selection method based on tolerance relation for emotion recognition is proposed in this paper, which can avoid the process of discretization. Based on the idea of 3DM, a method for selecting suitable threshold of tolerance relation is also proposed.

Definition 1. A continuous value decision information system is defined as a quadruple $S = (U, R, V, f)$, where U is a finite set of objects and $R = C \cup D$ is a finite set of attributes, C is the condition attribute set and $D = \{d\}$ is the decision attribute set. $\forall c \in C$, c is a continuous value attribute, $\forall d \in D$, d is a continuous value attribute or discrete value attribute.

A facial expression information system is a continuous value decision information system according to Def. 1.

Since all facial attribute values are continuous value which are always imprecise in some extent, and the process of discretization may affect the result of emotion recognition, it might be suitable to consider that continuous values equal to each other in some range rather than exactly. Based on this idea, a novel method is proposed in this paper, which could avoid the process of discretization.

Definition 2. A binary relation $R(x, y)$ defined on an attribute set B is called a tolerance relation if it is:

- symmetrical: $\forall_{x,y \in U}(R(x, y) = R(y, x))$,
- reflective: $\forall_{x \in U}(R(x, x) = R(x, x))$.

A new relation for continuous value decision information systems is defined as follows.

Definition 3. Let an information system $S = (U, R, V, f)$ be a continuous value decision information system, a new relation $R(x, y)$ is defined as $R(x, y) = \{(x, y) | x \in U \wedge y \in U \wedge \forall_{a \in C}(|a_x - a_y| \leq \varepsilon, 0 \leq \varepsilon \leq 1)\}$.

It is easy to see that $R(x, y)$ is a tolerance relation according to Def. 2 since $R(x, y)$ is symmetrical and reflective. An equivalence relation constitutes a partition of U, but a tolerance relation constitutes a covering of U, and equivalence relation is a particular type of tolerance relation.

Definition 4. Let $R(x, y)$ be a tolerance relation according to Def. 3, $n_R(x_i) = \{x_j | x_j \in U \wedge \forall_{a \in C}(|a_{x_i} - a_{x_j}| \leq \varepsilon)\}$ is called the tolerance class of x_i, and $|n_R(x_i)| = |\{x_j | x_j \in n_R(x_i), 1 \leq j \leq U\}|$ the cardinality of the tolerance class of x_i.

According to Def. 4, $\forall x \in U$, the bigger the tolerance class of x is, the more uncertain it will be and the less knowledge it will contain, on the contrary, the smaller the tolerance class of x is, the less uncertain it will be and the more knowledge it will contain. Accordingly, the concepts of knowledge entropy and conditional entropy are defined as follows.

Definition 5. Let $U = \{x_1, x_2, \ldots, x_{|U|}\}, R(x_i, x_j)$ be a tolerance relation defined on an attribute set B, knowledge entropy $E(R)$ of relation R is defined as $E(R) = -\frac{1}{|U|} \sum_{i=1}^{|U|} \log_2 \frac{|n_R(x_i)|}{|U|}$.

Definition 6. Let R and Q be tolerance relations defined on $U, R \cup Q$ is a relation satisfying R and Q simultaneously, and it is a tolerance relation too. $\forall x_i \in U, n_{R \cup Q}(x_i) = n_R(x_i) \cap n_Q(x_i)$ therefore, the knowledge entropy of $R \cup Q$ can be defined as $E(R \cup Q) = -\frac{1}{|U|} \sum_{i=1}^{|U|} \log_2 \frac{|n_{R \cup Q}(x_i)|}{|U|}$.

Definition 7. Let R and Q be tolerance relations defined on U, the conditional entropy of R with respect to Q is defined as $E(Q|R) = E(R \cup Q) - E(R)$.

2.2 Parameter Selection for Tolerance Relation Model Based on 3DM

In this section, a novel attribute reduction algorithm is developed based on the idea of domain-oriented data-driven data mining (3DM).

3DM is a data mining theory proposed by Guoyin Wang [8, 9]. According to the idea of 3DM, knowledge could be expressed in many different ways, there should be some relationship between the different formats of the same knowledge. In order to keep the knowledge unchanged in a data mining process, the properties of the knowledge should remain unchanged during a knowledge transformation process [4]. Otherwise, there should be some mistake in the process of knowledge transformation. Based on this idea, knowledge reduction can be seen as a process of knowledge transformation, and in the process, the properties of the knowledge should be remained.

Based on the idea of 3DM, the indiscernibility is decided when a decision information table is given, and the ability should be unchanged in the process of attribute reduction.

Definition 8. Let $S = (U, R, V, f)$ be a continuous value decision information system, if $\forall_{x_i, x_j \in U} (d_{x_i} \neq d_{x_j} \to \exists_{a \in C} (a_{x_i} \neq a_{x_j}))$ there is certain discernability of the conditional attribute set with respect to the decision attribute set in the continuous value decision information system S.

The discernibility can be seen as a fundamental ability of a continuous information decision system. According to 3DM, the indiscernibility should be unchanged in the process of knowledge acquisition. Therefore, the indiscernibility should be held if feature selection is done on a continuous value decision information system based on tolerance relation. Based on the standpoint that attribute values could be equal in some range while not be equal exactly in a continuous value decision information system, according to Def. 8, $\forall_{x_i, x_j \in U} (d_{x_i} \neq d_{x_j} \to \exists_{a \in C} (|a_{x_i} - a_{x_j}| > \varepsilon))$ and according to Def. 4, $x_j \notin n_R(x_i), x_i \notin n_R(x_j), n_R(x_i) \neq n_R(x_j)$. Accordingly, the indiscernibility of a tolerance relation can be got.
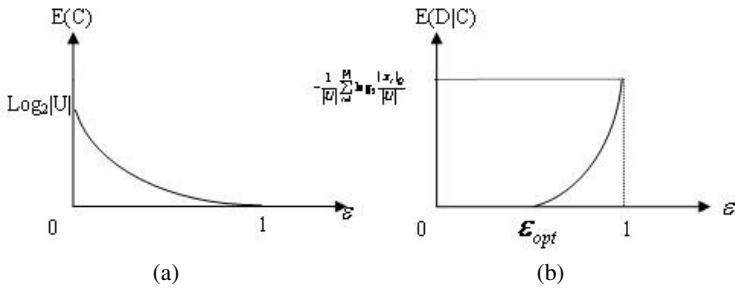
Fig. 1 **(a)** Relationship between $E(C)$ and ε. **(b)** Relationship between $E(D|C)$ and ε

Definition 9. Let $R(x,y)$ be a tolerance relation according to Def. 3, if $\forall_{x_i,x_j \in U}(d_{x_i} \neq d_{x_j} \rightarrow n_R(x_i) \neq n_R(x_j))$, $R(x,y)$ has the certain discernability. If $R(x,y)$ has certain discernability, according to Def. 9, $\forall_{x_i,x_j \in U}(n_R(x_i) = n_R(x_j) \rightarrow d_{x_i} = d_{x_j})$, therefore, $\forall_{x_i,x_j \in U}(x_i, x_j \in n_R(x_i) \rightarrow d_{x_i} = d_{x_j})$.

Based on these definitions above, we can get the theorem as follows. Detail proof is omited due to limited space.

Theorem 1. $E(D|C) = 0$ *is a necessary and sufficient condition of that there is an indiscernibility of the conditional attribute set with respect to the decision attribute set in a tolerance relation.*

The relationship between entropy, condition entropy and ε is shown in Fig. 1.

According to Fig. 1, if the threshold value of ε is ε_{opt}, it could make $E(D|C) = 0$ and the classification ability of the conditional attribute set with respect to decision attribute set is unchanged, at the same time, the tolerance class of x_i is the biggest with $E(D|C) = 0$. The knowledge granule of the conditional attribute set is the biggest in the case of ε_{opt}, the knowledge generalization is the best too.

Based on the discussion above, the suitable threshold ε_{opt} is found and the tolerance relation is set up accordingly. In this paper, the threshold of ε_{opt} is searched in $[0,1]$ based on binary search algorithm.

2.3 Algorithm for Finding Candidate Classifiers of Ensemble

Based on the tolerance relation model proposed above, a new algorithm for finding the core of a decision table is proposed as follows (Fig. 2).

After calculating the core of S using Algorithm 1 (Fig. 2), an algorithm for finding multiple reducts of S is proposed as follows.

Algorithm 2 (Fig. 3) could find multiple reducts of a decision table. Therefore, all the candidate classifiers could be generated accordingly. In this paper, SVM is used as the classifier, and all the classifiers take the same parameters.

Input:	A continuous value decision information system $S = (U, C \cup D, V, f)$, where U is a finite set of objects, C is the condition attribute set and $D = \{d\}$ is the decision attribute set
Output:	The core $Core_D(C)$ of S
Step 1	Compute ε_{opt}, then set up a tolerance relation model
Step 2	$Core_D(C) \leftarrow \emptyset$
Step 3	$\forall a_i \in C$ **if** $E(D|C) < E(D|C \setminus \{a_i\})$ **then** $Core_D(C) \leftarrow Core_D(C) \cup \{a_i\}$
Step 4	**return** $Core_D(C)$

Fig. 2 Algorithm 1 – An algorithm for finding core

Input:	A continuous value decision information system $S = (U, C \cup D, V, f)$
Output:	A set of reducts $REDU_i$
Step 1	Compute the core $Core_D(C)$ of decision table S using Algorithm 1
Step 2	$AR \leftarrow C \setminus Core_D(C); \quad REDU_i \leftarrow Core_D(C); \quad i \leftarrow 1$
Step 3	$\forall a_i \in AR$ compute $E(D|\{a_i\})$ and sort AR by $E(D|\{a_i\})$ ascendly
Step 4	**while** the attributes in $\cup_i REDU_i$ do not include all the attributes in C
Step 4.1	**while** $(E(D|REDU_i) \neq E(D|C))$
	$\forall a_j \in AR, REDU_i \leftarrow REDU_i \cup \{a_j\}, AR \leftarrow AR \setminus \{a_j\}$
	Compute $E(D|REDU_i)$
	if $(E(D|REDU_i) \neq E(D|C) \wedge REDU_i \setminus CORE_D(C) = AR)$
	$\quad i \leftarrow i - 1$
	$\quad$ **goto** Step 5
	endif
	endwhile
Step 4.2	$N \leftarrow |REDU_i|$
Step 4.3	**for** $j \leftarrow 0$ **to** $N - 1$
	if $a_j \in REDU_i$ **and** $a_j \notin CORE$ **then**
	$\quad REDU_i \leftarrow REDU_i \setminus \{a_j\}$
	$\quad$ Compute $E(D|REDU_i)$
	$\quad$ **if** $E(D|REDU_i) \neq E(D|C)$ **then**
	$\qquad REDU_i \leftarrow REDU_i \cup \{a_j\}$
	$\quad$ **endif**
	endif
	endfor
Step 4.4	$AR \leftarrow AR \setminus \{a\}, a \in REDU_i \wedge a = \min(E(D|\{a_j\})), a_j \in REDU_i$
Step 4.5	$i \leftarrow i + 1$
	endwhile
Step 5	**return** $REDU_i$

Fig. 3 Algorithm 2 – Algorithm for computing multiple reducts

2.4 Method of Dynamic Ensemble Static Selection

There are different ways for ensemble selection. Selective ensemble is a popular one. It selects the most diversity classifiers and integrates predictions of the selective classifiers. Unfortunately, it is difficult to define the measure of the diversity in real

Input:	A decision table $S = (U, C \cup D, V, f)$, and training subset, validation subset and testing subset
Output:	Classification output of the ensemble
Step 1	Find all the reducts of the training subset using Algorithm 2, and train all the candidate classifiers
Step 2	**for** each sample x in the test subset:
	for each reduct
	Calculate the K nearest neighborhood in the validation subset
	Classify the K nearest neighborhood by the candidate classifiers
	endfor
	endfor
Step 3	Classify x using the classifier with the highest correct classification ratio in Step 2, return classification result as the output of the ensemble

Fig. 4 Algorithm 3 – Algorithm of dynamic selection EFS

applications [11]. In this paper, a dynamic selection method is used instead of the statically selective method (Fig. 4).

3 Experiments and Discussion

In this section, three comparative experiments are done. In the first experiment, the proposed method is used. In the second experiment, all the classifiers are trained according to Algorithm 2 (Fig. 3), and the output of all the classifiers are combined according to the criterion of majority voting. This method is called integrate all in this paper. In the third experiment, a single reduct is found according to and a classifier is used based on the reduct. Since there are few open facial emotional dataset for all mankind, three facial emotional datasets are used in the experiments. The first dataset comes from the Cohn-Kanade AU-Coded Facial Expression (CKACFE) database and the dataset is a representation of western people in some extent. The second one is the Japanese female facial expression (JAFFE) database and it is a representation of eastern women in some extent The third one named CQUPTE is collected from 8 graduate students in Chongqing University of Posts and Communications in China, in which there are four female and four male. Detailed information of these datasets are listed in Table 1. In the experiments, 33 geometrical features of facial expression [10] are taken as features for emotion recognition.

Table 1 Three facial emotional datasets

Dataset Name	Samples	People	Emotion datasets
CKACFE	405	97	Happiness, Sadness, Surprise, Anger, Disgust, Fear, Neutral
JAFFE	213	10	Happiness, Sadness, Surprise, Anger, Disgust, Fear, Neutral
CQUPTE	653	8	Happiness, Sadness, Surprise, Anger, Disgust, Fear, Neutral

Table 2 Results of the comparative experiments

Dataset Name	Proposed method	Integrate all	Single classifier
CKACFE	0.7789418	0.736135	0.7739895
JAFFE	0.694773	0.664773	0.675635
CQUPTE	0.893968	0.8753145	0.878432
Average	0.789228	0.758741	0.776019

Each dataset are divided as train subset, validation subset and test subset according to 6:1:1, and 8-cross validation are taken. Results of the comparative experiments are shown in Table 2.

Through comparing all the three comparative experiment results, we can find that the proposed method is the most accurate one. Therefore, we can draw a conclusion that the proposed method is effective and a suitable method for emotion recognition.

Through comparing the proposed method and the method of integrating all the classifiers, we can find that the proposed method is superior to the other one. Therefore, we can draw a conclusion that dynamically selecting classifier from the candidates is more suitable for emotion recognition than the method of integrating all the candidate classifiers. Since the emotions are different from one and another, each candidate classifiers is suitable for a special subset of samples, the most suitable classifier for the unseen sample should be selected each time. Good results could not be got in some cases by combining the results of all the classifiers since some candidate classifiers may get conflict results for an unseen sample.

Through comparing the proposed method and the method of single classifier, we can find that the proposed method is superior to the other one too. Therefore, we can draw a conclusion that dynamically selecting classifier is more suitable for emotion recognition than selecting a static classifier. Although both methods use a single classifier for an unseen sample, the proposed method can get a better result since it uses the local properties of the unseen sample.

4   Conclusion and Future Work

In this paper, a novel emotion recognition method is proposed based on the ensemble learning, and it is based on the strategy of dynamic selection. Firstly, a feature selection method is proposed based on the rough set and domain oriented data driven data mining theory. It can get multiple reducts and candidate classifiers accordingly. Secondly, for each unseen sample, its nearest neighborhood is found and the most accuracy classifier is selected from the candidates. At last, the dynamically selected classifier is used for the unseen sample and the classification result is got. The proposed method is proved to be a suitable method for emotion recognition by the results of the comparative experiments. In the future, other strategies of dynamic selection will be studied.

References

1. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
2. Ditterrich, T.G.: Machine learning research: four current direction. Artificial Intelligence Magzine 4, 97–136 (1997)
3. Freund, Y.: Boosting a weak algorithm by majority. Information and Computation 121(2), 256–285 (1995)
4. Ohsuga, S.: Knowledge discovery as translation. In: Lin, T.Y., et al. (eds.) Foundations of Data Mining and Knowledge Discovery, pp. 1–19. Springer, Heidelberg (2005)
5. Picard, R.W.: Affective computing: Challenges. International Journal of Human-Computer Studies 59(1), 55–64 (2003)
6. Russell, B., Christian, P.: The role of affect and emotion in HCI. In: Peter, C., Beale, R. (eds.) Affect and Emotion in Human-Computer Interaction. LNCS, vol. 4868, pp. 1–11. Springer, Heidelberg (2008)
7. Scott, B., Clifford, N.: Emotion in hunam-computer interaction. In: Julie, A.J., Sears, A. (eds.) The Human-computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, pp. 81–93. Lawrence Erlbaum Associates Press, Mahwah (2003)
8. Wang, G.Y.: Introduction to 3DM: Domain-oriented data-driven data mining. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS, vol. 5009, pp. 25–26. Springer, Heidelberg (2008)
9. Wang, G.Y., Wang, Y.: Domain-oriented data-driven data mining: a new understanding for data mining. Journal of Chongqing University of Posts and Telecommunications 20(3), 266–271 (2008)
10. Yang, Y., Wang, G.Y., Chen, P.J., et al.: Feature selection in audiovisual emotion recognition based on 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. 283–294. Springer, Heidelberg (2007)
11. Zhou, Z.H.: Ensemble learning. In: Li, S.Z. (ed.) Encyclopedia of Biometrics, pp. 1–5. Springer, Heidelberg (2009)

Part V
Rough Fuzzy Investigations

On Construction of Partial Association Rules with Weights

Mikhail Ju. Moshkov, Marcin Piliszczuk, and Beata Zielosko

Abstract. This paper is devoted to the study of approximate algorithms for minimization of the total weight of attributes occurring in partial association rules. We consider mainly greedy algorithms with weights for construction of rules. The paper contains bounds on precision of these algorithms and bounds on the minimal weight of partial association rules based on an information obtained during the greedy algorithm run.

Keywords: partial association rules, greedy algorithms, weights of attributes.

1 Introduction

In this paper, we consider the case, where each attribute of information system [11] has its own weight, and we should minimize the total weight of attributes occurring in a partial association rule. If weights of attributes characterize time complexity of attribute value computation, then we try to minimize the total time complexity of computation of attributes from partial association rule. If weights characterize a risk

Mikhail Ju. Moshkov
Division of Mathematical and Computer Sciences and Engineering,
King Abdullah University of Science and Technology,
P.O. Box 55455, Jeddah 21534, Saudi Arabia
e-mail: `mikhail.moshkov@kaust.edu.sa`

Marcin Piliszczuk
ING Bank Śląski S.A.,
Sokolska 34, 40-086 Katowice, Poland
e-mail: `marcin.piliszczuk@ingbank.pl`

Beata Zielosko
Institute of Computer Science, University of Silesia,
Będzińska 39, 41-200 Sosnowiec, Poland
e-mail: `beata.zielosko@us.edu.pl`

of attribute value computation (as in medical or technical diagnosis), then we try to minimize the total risk, etc.

We consider not only exact but also partial (approximate) association rules since exact rules can be overfitted, i.e., dependent essentially on the noise or adjusted too much to the existing examples. This idea is not new. For years, in rough set theory and its extensions partial reducts and partial rules are studied intensively by J.G. Bazan, M.Ju. Moshkov, H.S. Nguyen, Z. Pawlak, M. Piliszczuk, M. Quafafou, A. Skowron, D. Ślęzak, J. Wróblewski, W. Ziarko, B. Zielosko and others (see, e.g., [2, 7, 8, 9, 10, 12, 13, 14, 16, 18, 19, 20, 21]).

We study three types of greedy algorithms with weights: standard greedy algorithm, greedy algorithm with two thresholds, and so-called extended greedy algorithm. Note that the considered approach to partial association rule construction is essentially different from the well known approach based on the mining of frequent itemsets [1, 5].

We prove that, under some natural assumptions on the class *NP*, the standard greedy algorithm is close to the best polynomial approximate algorithms for minimization of the total weight of attributes occurring in partial association rules.

We show that for a part of information systems the greedy algorithm with two thresholds can give us (by the choice of a value of the second threshold) better results than the standard greedy algorithm. We create a polynomial time extended greedy algorithm that uses advantages of the greedy algorithm with two thresholds, but allows us to avoid the consideration of infinite number of values of the second threshold.

We show that based on an information received during the greedy algorithm run it is possible to obtain nontrivial lower bound on the minimal total weight of attributes occurring in partial association rules. We obtain also new bounds on the precision of the greedy algorithm with two thresholds.

The proofs of the most part of statements mentioned in this paper are based on results for covers and partial covers [3, 4, 6, 7, 8, 15, 17].

2   Main Notions

An *information system I* is a table with n rows (corresponding to objects) and m columns labeled with attributes $a_1, \ldots, a_m$. This table is filled by nonnegative integers (values of attributes). Let w be a *weight function* for I which corresponds to each attribute a_i a natural number $w(a_i)$.

Let $r = (b_1, \ldots, b_m)$ be a row of I, and a_p be an attribute from the set $\{a_1, \ldots, a_m\}$. By $U(I, r, a_p)$ we denote the set of rows from I which are different from r in the column a_p and in at least one column a_j such that $j \in \{1, \ldots, m\} \setminus \{p\}$. We will say that an attribute a_i *separates* a row $r' \in U(I, r, a_p)$ from the row r if the rows r and r' have different numbers at the intersection with the column a_i. For $i = 1, \ldots, m$, $i \neq p$, we denote by $U(I, r, a_p, a_i)$ the set of rows from $U(I, r, a_p)$ which attribute a_i separates from the row r.

Input : Information system I with attributes $a_1,\ldots,a_m$, row $r = (b_1,\ldots,b_m)$ of I, attribute a_p of I, weight function w for I, and real numbers α and γ such that $0 \le \gamma \le \alpha < 1$.

Output: α-association rule for I, r, and a_p.

$Q \longleftarrow \emptyset$; $D \longleftarrow \emptyset$; $M \longleftarrow \lceil |U(I,r,a_p)|(1-\alpha)\rceil$; $N \longleftarrow \lceil |U(I,r,a_p)|(1-\gamma)\rceil$;

while $|D| < M$ **do**

> select $a_i \in \{a_1,\ldots,a_m\} \setminus \{a_p\}$ with minimal index i such that $U(I,r,a_p,a_i) \setminus D \ne \emptyset$ and the value
>
> $$\frac{w(a_i)}{\min\{|U(I,r,a_p,a_i) \setminus D|, N - |D|\}}$$
>
> is minimal
>
> $Q \longleftarrow Q \cup \{a_i\}$; $D \longleftarrow D \cup U(I,r,a_p,a_i)$;

end

return $\bigwedge_{a_i \in Q}(a_i = b_i) \to a_p = b_p$;

Fig. 1 Greedy algorithm with two thresholds α and γ for partial association rule construction

Let α be a real number and $0 \le \alpha < 1$. A rule

$$(a_{i_1} = b_{i_1}) \wedge \ldots \wedge (a_{i_t} = b_{i_t}) \to a_p = b_p \tag{1}$$

is called an *α-association rule* for I, r and a_p if $i_1,\ldots,i_t \in \{1,\ldots,m\} \setminus \{p\}$, and attributes $a_{i_1},\ldots,a_{i_t}$ separate from r at least $(1-\alpha)|U(I,r,a_p)|$ rows from $U(I,r,a_p)$ (such rules are also called *partial* association rules). The number $\sum_{j=1}^{t} w(a_{i_j})$ is called the *weight* of the considered association rule.

If $U(I,r,a_p) = \emptyset$, then for any $i_1,\ldots,i_t \in \{1,\ldots,m\} \setminus \{p\}$ the rule (1) is an α-association rule for I, r and a_p. The rule (1) with empty left-hand side (when $t = 0$) is also an α-association rule for I, r and a_p if $U(I,r,a_p) = \emptyset$. The weight of this rule is equal to 0.

For example, 0.01-association rule means that we should separate from r at least 99% of rows from $U(I,r,a_p)$. We denote by $L_{\min}(\alpha) = L_{\min}(\alpha,I,r,a_p,w)$ the minimal weight of α-association rule for I, r and a_p.

Let α,γ be real numbers such that $0 \le \gamma \le \alpha < 0$. We now describe a *greedy algorithm with thresholds α and γ* which constructs an α-association rule for given I, r, a_p and weight function w (see Fig. 1).

Let us denote by $L_{\text{greedy}}^{\gamma}(\alpha) = L_{\text{greedy}}^{\gamma}(\alpha,I,r,a_p,w)$ the weight of α-association rule constructed by the considered algorithm for given information system I, row r, attribute a_p and weight function w.

3 Precision of Standard Greedy Algorithm

Theorem 1. *Let I be an information system, r be a row of I, a_p be an attribute of I, $U(I,r,a_p) \ne \emptyset$, w be a weight function for I, and α be a real number such that $0 \le \alpha < 1$. Then $L_{\text{greedy}}^{\alpha}(\alpha) \le L_{\min}(\alpha)H\left(\lceil (1-\alpha)|U(I,r,a_p)|\rceil\right)$.*

Theorem 2. *Let I be an information system with m columns labeled with attributes $a_1, \ldots, a_m$, r be a row of I, a_p be an attribute of I, $U(I, r, a_p) \neq \emptyset$, w be a weight function for I, and α be a real number, $0 \leq \alpha < 1$. Then $L^{\alpha}_{\mathrm{greedy}}(\alpha) \leq L_{\min}(\alpha) H \left(\max_{i \in \{1, \ldots, m\} \setminus \{p\}} |U(I, r, a_p, a_i)| \right)$.*

4   Polynomial Approximate Algorithms

Let $0 \leq \alpha < 1$. We now consider the following problem: for a given information system I, row r of I, attribute a_p of I and weight function w for I it is required to find an α-association rule for I, r and a_p with minimal weight.

Theorem 3. *Let $0 \leq \alpha < 1$. Then the problem of construction of α-association rule with minimal weight is NP-hard.*

So we should consider polynomial approximate algorithms for minimization of α-association rule weight.

Theorem 4. *Let $\alpha \in \mathbb{R}$ and $0 \leq \alpha < 1$. If $NP \nsubseteq DTIME(n^{O(\log \log n)})$, then for any ε, $0 < \varepsilon < 1$, there is no polynomial algorithm that for a given information system I, row r of I, attribute a_p of I such that $U(I, r, a_p) \neq \emptyset$, and weight function w for I constructs an α-association rule for I, r and a_p which weight is at most $(1 - \varepsilon) L_{\min}(\alpha, I, r, a_p, w) \ln |U(I, r, a_p)|$.*

Theorem 5. *Let α be a real number such that $0 \leq \alpha < 1$. If $P \neq NP$, then there exists $\delta > 0$ such that there is no polynomial algorithm that for a given information system I, row r of I, attribute a_p of I such that $U(I, r, a_p) \neq \emptyset$, and weight function w for I constructs an α-association rule for I, r and a_p which weight is at most $\delta L_{\min}(\alpha, I, r, a_p, w) \ln |U(I, r, a_p)|$.*

From Theorem 2 it follows that $L^{\alpha}_{\mathrm{greedy}}(\alpha) \leq L_{\min}(\alpha)(1 + \ln |U(I, r, a_p)|)$. From this inequality and from Theorems 4 and 5 it follows that, under natural assumptions on the class *NP*, the standard greedy algorithm is close to the best polynomial approximate algorithms for minimization of partial association rule weight. However, we can try to improve the results of the work of standard greedy algorithm for some part of information systems.

5   Greedy Algorithm with Two Thresholds vs. Standard Greedy Algorithm

A *binary information system I* is a table with n rows (corresponding to objects) and m columns labeled with attributes $a_1, \ldots, a_m$. This table is filled by numbers from $\{0, 1\}$ (values of attributes). The number of such information systems is equal to 2^{mn}.

Let $r = (b_1, \ldots, b_m)$ be a row of I and a_p be an attribute of I. We will say that the triple (I, r, a_p) is a *minor triple over I* if the number of $\neg b_p$ in the column a_p

is at least $\frac{n}{2}$. We prove that, under some assumptions on the number of attributes and rows, for the most part of binary information systems I, for each minor triple (I, r, a_p) over I there exists a weight function w and real numbers $\alpha, \gamma, 0 \leq \gamma < \alpha < 1$, such that the weight of α-association rule constructed by the greedy algorithm with thresholds α and γ for I, r, a_p and w is less than the weight of α-association rule constructed by the greedy algorithm with equal thresholds α and α (by the standard greedy algorithm).

Theorem 6. *Let us consider binary information systems with $n \geq 8$ rows and m columns labeled with attributes $a_1, \ldots, a_m$. The fraction of information systems I, for each of which for each minor triple (I, r, a_p) over I there exists a weight function w for I and real numbers α, γ such that $0 \leq \gamma < \alpha < 1$ and $L^{\gamma}_{\mathrm{greedy}}(\alpha, I, r, a_p, w) < L^{\alpha}_{\mathrm{greedy}}(\alpha, I, r, a_p, w)$, is at least*

$$1 - \frac{n^3 m 2^{4m}}{n^{\frac{m}{2}}} - \frac{n^2 m}{2^m}.$$

One can show that if $m \leq n \leq \frac{2^{m/2}}{m}$, then the considered fraction tends to 1 as m tends to infinity.

6   Extended Greedy Algorithm

The results obtained in the previous section show that the greedy algorithm with two thresholds α and γ is of some interest. We now consider a polynomial modification of this algorithm which allows us to use advantages of the greedy algorithm with two thresholds without consideration of all values of the second threshold γ. We will say about this modification as about *extended* greedy algorithm.

Let I be an information system with m columns labeled with attributes $a_1, \ldots, a_m$, $r = (b_1, \ldots, b_m)$ be a row of I, a_p be an attribute of I, $U(I, r, a_p) \neq \emptyset$, w be a weight function for I and α be a real number such that $0 \leq \alpha < 1$.

It is impossible to consider effectively all γ such that $0 \leq \gamma \leq \alpha$. Instead of this, we can consider all natural N such that $M \leq N \leq |U(I, r, a_p)|$, where $M = \lceil |U(I, r, a_p)|(1 - \alpha) \rceil$ (see Fig. 1). For each $N \in \{M, \ldots, |U(I, r, a_p)|\}$, we apply algorithm from Fig. 1 with parameters M and N to I, r, a_p and w and after that choose an α-association rule with minimal weight among constructed α-association rules.

7   Bounds on $L_{\min}(\alpha)$ and $L^{\gamma}_{\mathrm{greedy}}(\alpha)$

We fix some information received during the run of greedy algorithm with two thresholds, and find the best lower bound on the value $L_{\min}(\alpha)$ depending on this information.

Let I be an information system, r be a row of I, a_p be an attribute of I such that $U(I,r,a_p) \neq \emptyset$, w be a weight function for I, and α, γ be real numbers such that $0 \leq \gamma \leq \alpha < 1$. We now apply the greedy algorithm with thresholds α and γ to the information system I, row r, attribute a_p and weight function w. Let during the construction of α-association rule the greedy algorithm choose sequentially attributes $a_{g_1}, \ldots, a_{g_t}$.

Let us denote $U(I,r,a_p,a_{g_0}) = \emptyset$ and $\delta_0 = 0$. For $i = 1, \ldots, t$, we denote $\delta_i = |U(I,r,a_p,a_{g_i}) \setminus (U(I,r,a_p,a_{g_0}) \cup \ldots \cup U(I,r,a_p,a_{g_{i-1}}))|$ and $w_i = w(a_{g_i})$. As information on the greedy algorithm run we will use numbers $M_L = M_L(\alpha,\gamma,I,r,a_p,w) = \lceil |U(I,r,a_p)|(1-\alpha) \rceil$ and $N_L = N_L(\alpha,\gamma,I,r,a_p,w) = \lceil |U(I,r,a_p)|(1-\gamma) \rceil$, and tuples $\Delta_L = \Delta_L(\alpha,\gamma,I,r,a_p,w) = (\delta_1, \ldots, \delta_t)$ and $W_L = W_L(\alpha,\gamma,I,r,a_p,w) = (w_1, \ldots, w_t)$.

For $i = 0, \ldots, t-1$, we denote

$$\rho_i = \left\lceil \frac{w_{i+1}(M_L - (\delta_0 + \cdots + \delta_i))}{\min\{\delta_{i+1}, N_L - (\delta_0 + \cdots + \delta_i)\}} \right\rceil.$$

Let us define parameter $\rho_L(\alpha,\gamma) = \rho_L(\alpha,\gamma,I,r,a_p,w)$ as follows:

$$\rho_L(\alpha,\gamma) = \max\{\rho_i : i = 0, \ldots, t-1\}.$$

We will show that $\rho_L(\alpha,\gamma)$ is the best lower bound on $L_{\min}(\alpha)$ depending on M_L, N_L, Δ_L and W_L.

Theorem 7. *For any information system I, row r of I, attribute a_p of I such that $U(I,r,a_p) \neq \emptyset$, weight function w for I, and real numbers α, γ, $0 \leq \gamma \leq \alpha < 1$, the inequality $L_{\min}(\alpha,I,r,a_p,w) \geq \rho_L(\alpha,\gamma,I,r,a_p,w)$ holds, and there exists an information system I', a row r' of I', an attribute a' of I' and a weight function w' for I' such that $U(I',r',a'_p) \neq \emptyset$ and*

$$M_L(\alpha,\gamma,I',r',a',w') = M_L(\alpha,\gamma,I,r,a_p,w),$$
$$N_L(\alpha,\gamma,I',r',a',w') = N_L(\alpha,\gamma,I,r,a_p,w),$$
$$\Delta_L(\alpha,\gamma,I',r',a',w') = \Delta_L(\alpha,\gamma,I,r,a_p,w),$$
$$W_L(\alpha,\gamma,I',r',a',w') = W_L(\alpha,\gamma,I,r,a_p,w),$$
$$\rho_L(\alpha,\gamma,I',r',a',w') = \rho_L(\alpha,\gamma,I,r,a_p,w),$$
$$L_{\min}(\alpha,I',r',a',w') = \rho_L(\alpha,\gamma,I',r',a',w').$$

Let us consider a property of the parameter $\rho_L(\alpha,\gamma)$ which is important for practical use of the bound from Theorem 7.

Proposition 1. *Let I be an information system, r be a row of I, a_p be an attribute of I, $U(I,r,a_p) \neq \emptyset$, w be a weight function for I, and α, γ be real numbers such that $0 \leq \gamma \leq \alpha < 1$. Then*

$$\rho_L(\alpha,\alpha,I,r,a_p,w) \geq \rho_L(\alpha,\gamma,I,r,a_p,w).$$

We now study some properties of the parameter $\rho_L(\alpha, \gamma)$ and obtain two upper bounds on the value $L^{\gamma}_{\text{greedy}}(\alpha)$ which do not depend directly on cardinality of the set $U(I, r, a_p)$ and cardinalities of subsets $U(I, r, a_p, a_i)$.

Theorem 8. *Let I be an information system, r be a row of I, a_p be an attribute of I, $U(I, r, a_p) \neq \emptyset$, w be a weight function for I, α, γ be real numbers and $0 \leq \gamma < \alpha < 1$. Then*

$$L^{\gamma}_{\text{greedy}}(\alpha, I, r, a_p, w) < \rho_L(\gamma, \gamma, I, r, a_p, w) \left(\ln \left(\frac{1 - \gamma}{\alpha - \gamma} \right) + 1 \right).$$

Corollary 1. *Let ε be a real number and $0 < \varepsilon < 1$. Then for any α, $\varepsilon \leq \alpha < 1$, the following inequalities hold: $\rho_L(\alpha, \alpha) \leq L_{\min}(\alpha) \leq L^{\alpha - \varepsilon}_{\text{greedy}}(\alpha) < \rho_L(\alpha - \varepsilon, \alpha - \varepsilon) \left(\ln \left(\frac{1}{\varepsilon} \right) + 1 \right).$*

For example, $\ln(\frac{1}{0.01}) + 1 < 5.61$ and $\ln(\frac{1}{0.1}) + 1 < 3.31$. The obtained results show that the lower bound $L_{\min}(\alpha) \geq \rho_L(\alpha, \alpha)$ is nontrivial.

Theorem 9. *Let I be an information system, r be a row of I, a_p be an attribute of I, $U(I, r, a_p) \neq \emptyset$, w be a weight function for I, α, γ be real numbers and $0 \leq \gamma < \alpha < 1$. Then*

$$L^{\gamma}_{\text{greedy}}(\alpha, I, r, a_p, w) < L_{\min}(\gamma, I, r, a_p, w) \left(\ln \left(\frac{1 - \gamma}{\alpha - \gamma} \right) + 1 \right).$$

Corollary 2. $L^{0.3}_{\text{greedy}}(0.5) < 2.26 L_{\min}(0.3)$, $L^{0.1}_{\text{greedy}}(0.2) < 3.20 L_{\min}(0.1)$, $L^{0.001}_{\text{greedy}}(0.01) < 5.71 L_{\min}(0.001)$, $L^{0}_{\text{greedy}}(0.001) < 7.91 L_{\min}(0)$.

Corollary 3. *Let $0 < \alpha < 1$. Then $L^{0}_{\text{greedy}}(\alpha) < L_{\min}(0) \left(\ln \left(\frac{1}{\alpha} \right) + 1 \right)$.*

Corollary 4. *Let ε be a real number, and $0 < \varepsilon < 1$. Then for any α such that $\varepsilon \leq \alpha < 1$ the following inequalities hold: $L_{\min}(\alpha) \leq L^{\alpha - \varepsilon}_{\text{greedy}}(\alpha) < L_{\min}(\alpha - \varepsilon) \left(\ln \left(\frac{1}{\varepsilon} \right) + 1 \right).$*

8 Conclusions

The paper is devoted mainly to the analysis of greedy algorithms with weights for partial association rule construction. Under some natural assumptions on the class *NP*, the standard and extended greedy algorithms are close to the best polynomial approximate algorithms for the minimization of the weight of partial association rules. For a part of information systems and weight functions, the extended algorithm constructs better rules than the standard algorithm. The obtained results show that the lower bound on the minimal weight of partial association rules, based on an information about the run of greedy algorithm with two thresholds, is nontrivial and can be used in practice.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) Proceedings of the 20th International Conference on Very Large Data Bases. Morgan Kaufmann, San Francisco (1994)
2. Bazan, J.G.: Discovery of decision rules by matching new objects against data tables. In: Polkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS (LNAI), vol. 1424, pp. 521–528. Springer, Heidelberg (1998)
3. Chvátal, V.: A greedy heuristic for the set-covering problem. Mathematics of Operations Research 4(3), 233–235 (1979)
4. Feige, U.: A threshold of $\ln n$ for approximating set cover (preliminary version). In: Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, Philadelphia, US (1996)
5. Goethals, B.: Frequent itemset mining implementations repository, http://fimi.cs.helsinki.fi
6. Kearns, M.J.: The computational complexity of machine learning. Massachussetts Institute of Technology Press, Cambridge (1990)
7. Moshkov, M.J., Piliszczuk, M., Zielosko, B.: On partial covers, reducts and decision rules with weights. 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. 211–246. Springer, Heidelberg (2007)
8. Moshkov, M.J., Piliszczuk, M., Zielosko, B.: Partial covers, reducts and decision rules in rough sets: theory and applications. Studies in Computational Intelligence, vol. 145. Springer, Heidelberg (2008)
9. Nguyen, H.S.: Approximate boolean reasoning: foundations and applications in data mining. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, pp. 334–506. Springer, Heidelberg (2006)
10. 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)
11. Pawlak, Z.: Rough sets – theoretical aspects of reasoning about data. Kluwer Academic Publishers, Dordrecht (1991)
12. Pawlak, Z.: Rough set elements. In: Polkowski, L., Skowron, A. (eds.) Rough sets in knowledge discovery 1. Methodology and applications. Studies in Fuzziness and Soft Computing, Physica-Verlag, Heidelberg (1998)
13. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177, 3–27; Rough sets: Some extensions. Information Sciences 177, 28–40; Rough sets and boolean reasoning. Information Sciences 177, 41–73 (2007)
14. Quafafou, M.: α-RST: a generalization of rough set theory. Information Sciences 124(1-4), 301–316 (2000)
15. Raz, R., Safra, S.: A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In: Proceedings of the 29th Annual ACM symposium on the theory of computing. El Paso, US (1997)
16. Skowron, A.: Rough sets in KDD. In: Shi, Z., Faltings, B., Musen, M. (eds.) Proceedings of the 16th IFIP World Computer Congress. Publishing House of Electronic Industry, Beijing (2000)
17. Slavík, P.: Approximation algorithms for set cover and related problems. Ph.D. thesis, University of New York, Buffalo, US (1998)

18. Ślęzak, D.: Normalized decision functions and measures for inconsistent decision tables analysis. Fundamenta Informaticae 44(3), 291–319 (2000)
19. Ślęzak, D.: Approximate entropy reducts. Fundamenta Informaticae 53(3-4), 365–390 (2002)
20. Wróblewski, J.: Ensembles of classifiers based on approximate reducts. Fundamenta Informaticae 47(3-4), 351–360 (2001)
21. Ziarko, W.: Analysis of uncertain information in the framework of variable precision rough sets. Foundations of Computing and Decision Sciences 18(3-4), 381–396 (1993)

Fuzzy Rough Entropy Clustering Algorithm Parametrization

Dariusz Małyszko and Jarosław Stepaniuk

Abstract. Image processing represents active research area that requires advanced and sophisticated methods capable of handling novel emerging imagery technologies. Adequate and precise capture and image interpretation is primarily based on proper image segmentation. Advances in correct image partitioning are still embracing new research areas such as fuzzy sets, rough sets and rough fuzzy sets. Present research concentrates on new rough entropy clustering algorithm Fuzzy RECA Rough Entropy Clustering Algorithm and extension relative to distance threshold and fuzzy threshold, namely its parameters having impact on rough entropy calculation. Different rough entropy measures are calculated and incorporated into Fuzzy RECA based clustering algorithm on satellite image data set. Presented results suggest that proposed fuzzy thresholds capture properly image properties and it is possible to take advantage of these characteristics in real image processing applications.

Keywords: granular computing, standard and fuzzy rough entropy clustering algorithm, image clustering, rough sets, fuzzy sets, rough fuzzy sets.

1 Introduction

Image segmentation presents the low-level image transformation routine concerned with image partitioning into distinct disjoint and homogenous regions. Clustering or data grouping describes distinct key procedure in image processing and segmentation. The presented research is based on combining the concept of rough sets and entropy measure in the area of image segmentation.

In the previous work [1, 2] new algorithmic scheme RECA in the area of rough entropy based partitioning routines has been proposed. Rough entropy clustering

Dariusz Małyszko · Jarosław Stepaniuk
Department of Computer Science, Białystok University of Technology,
Wiejska 45A, 15-351 Białystok, Poland
e-mail: {malyszko,jstepan}@wi.pb.edu.pl

incorporates the notion of rough entropy into clustering model taking advantage of dealing with some degree of uncertainty in analyzed data. Given predefined number of clusters, with each cluster lower and upper approximations are associated. Image points that are close to the cluster contribute their impact by increasing lower and upper cluster approximation value in case of their proximity only to that cluster or distribute uniformly their impact on some number of upper cluster approximations otherwise. After lower and upper approximation determination for all clusters, their roughness and rough entropy value calculation proceeds. On the base of entropy maximization law, the best segmentation is achieved in case of maximal entropy value. For this purpose, an evolutionary population of separate solutions is maintained with solutions with predefined number of cluster prototypes. For each solution, respective rough entropy measure is calculated and subsequently, new populations are created from parental solutions with high values of this fitness measure.

Additionally, an extension of RECA algorithm into fuzzy domain has been elaborated in the form of Fuzzy RECA. In Fuzzy RECA algorithm, the impact of each image point on upper cluster approximations of sufficiently close clusters is not constant and depends upon point's distance to these clusters. Upper cluster approximations are increased by fuzzy measure for all image points that are sufficiently close to more than one cluster center. The paper deals with this fuzzy distance approximation measures related with RECA algorithm in the form of standard distance difference measure with constant crisp increase of approximations, standard distance difference measure with constant fuzzy increase of approximations and fuzzy threshold measure.

In Sect. 2, RECA clustering algorithmic schemes and RECA parametrization has been presented. Image segmentation evaluation methods have been described in Sect. 3. Experimental setup and results have been put in Sect. 4. At the end brief concluding remarks have been appended.

2 Fuzzy RECA Clustering

2.1 General RECA Algorithm

Segmentation is the standard image partitioning process that results in determining and creation of disjoint and homogeneous image regions. Regions resulting from the image segmentation should be uniform and homogeneous with respect to some characteristics, regions interiors should be simple and without many small holes, adjacent regions should be significantly different with respect to the uniformity characteristics and each segment boundary should be comparatively simple and spatially accurate.

Rough entropy framework in image segmentation domain has been proposed in [3] and further extended in [2] as fully robust clustering algorithmic model. Rough entropy clustering has been based on the combining rough set theory together with the notion of entropy.

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

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

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 [4, 5, 6].

Shannon entropy notion describes uncertainty of the system and is defined as follows $E(p_1, \ldots, p_k) = \sum_{l=1}^{k} -p_l \times \log(p_l)$ where p_l represents probability of the state l and $l = 1, \ldots, k$. In this context, combined rough entropy formula is given as

$$RE(AS_B, \{X_1, \ldots, X_k\}) = \sum_{l=1}^{k} -\frac{e}{2} \times R(AS_B, X_l) \times \log(R(AS_B, X_l)), \tag{2}$$

where $R(AS_B, X_l)$ represents roughness of the cluster X_l, $l \in \{1, \ldots, k\}$ indexes the set of all clusters ($\bigcup_{l=1}^{k} X_l = U$ and for any $p \neq l$ and $p, l \in \{1, \ldots, k\}$ $X_p \cap X_l = \emptyset$).

Fuzzy membership value $\mu_{C_l}(x_i) \in [0,1]$ for the data point $x_i \in U$ in cluster C_l (equivalently X_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 μ represents fuzzifier value that should be greater than 1.0 and $d(x_i, C_l)$ denotes distance between data object x_i and cluster (center) C_l.

Let $d > 0$ be a given natural number. In image segmentation domain, data objects are represented as d-dimensional vectors. Most often, data attribute values are 8-bit integers in the range from 0 to 255. Number of dimensions depends upon concrete imagery type, most often images are represented as one-dimensional vectors ($d = 1$) in case of grey-scale images $x_i = (x_i^{\text{Grey}})$ with attribute set $A = \{\text{Grey}\}$ and three-dimensional vectors ($d = 3$) RGB $x_i = (x_i^{\text{R}}, x_i^{\text{G}}, x_i^{\text{B}})$ in case of color images with R, G, and B meaning respectively Red, Green, and Blue channels with attribute set $A = \{R, G, B\}$. In this setting, most often Euclidean distance is taken into consideration.

In order to discover optimal clustering solutions in the form of cluster centers an evolutionary searching algorithm has been employed. In evolutionary algorithm, each chromosome represents $k > 1$ d-dimensional cluster centers. In the experiments, chromosomes selected to mating pool undergo averaging cross-over operation and mutation operation with mutation probability set to 0.15 value. Mutation operation with 0.15 probability for each cluster center coordinate changed the coordinate value into random number in the range from 0 to 255.

2.2 Standard RECA

Standard *RECA* algorithm proposed in [2] incorporates computation of lower and upper approximations for the given cluster centers and considering these two set cardinalities during calculation of roughness and further rough entropy clustering measure. Rough Entropy Clustering Algorithm flow has been presented in Fig. 2. Rough measure general calculation routine has been given in Fig. 3. Exemplary borders for RECA algorithm have been presented in Figs. 1b and 1c compared to crisp border in Fig. 1a. In all presented algorithms, before calculations, lower and upper cluster approximations should be set to zero.

2.3 Fuzzy RECA – with Fuzzy Approximations

In distance difference based threshold RECA with fuzzy approximations, distances between the analyzed data object x_i and all clusters centers are computed. After

(a)   (b)   (c)
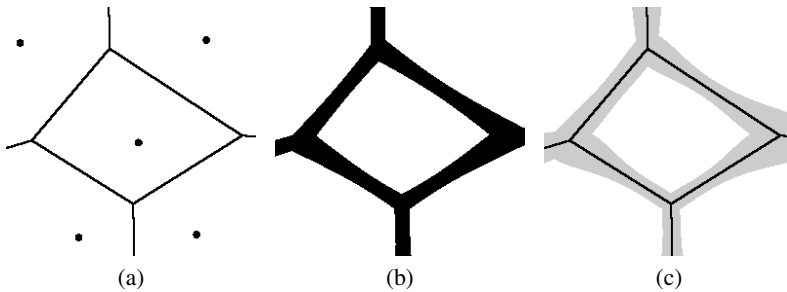
Fig. 1 **a** Cluster borders for RECA and selected centers. **b** Fuzzy RECA with $\varepsilon_{dist} = 15$. **c** Threshold RECA with $\varepsilon_{fuzz} = 0.21$. Number of clusters $k = 5$ in two dimensional attribute domain

---

Data: Input Image, k – number of clusters, *Size* – number of chromosomes in evolutionary
         population
Result: Optimal Cluster Centers
1. Create X population with *Size* random chromosomes (solutions) each encoding k cluster centers
repeat
  forall *chromosomes of X* **do**
      calculate their rough entropy measure values *RECA*
      Fuzzy_Rough_Entropy_Measure
  end
  create mating pool Y from parental X population
  apply selection, cross-over and mutation operators to Y population
  replace X population with Y population
until *termination criteria (most often predefined number of iterations)* ;

Fig. 2 General *RECA* Algorithm Flow

---

foreach *Data object x_i* **do**
    Determine the closest cluster C_l for x_i
    Increment Lower(C_l) and Upper(C_l) by 1.0
    foreach *Cluster Cluster $C_m \neq C_l$ with $|d(x_i, C_m) - d(x_i, C_l)| \leq \varepsilon_{\text{dist}}$* **do**
        Increment Upper(C_m) by 1.0
    end
end

for *$l = 1$ to k (number of data clusters)* **do**
    $roughness(C_l) = 1 - Lower(C_l)/Upper(C_l)$
Fuzzy_RE = 0
for *$l = 1$ to k (number of data clusters)* **do**
    $Fuzzy\_RE = Fuzzy\_RE - \frac{e}{2} \times roughness(C_l) \times \log(roughness(C_l))$

Fig. 3 General *RECA* – calculation of cluster Lower and Upper Approximations, Roughness and Fuzzy Rough Entropy

---

foreach *Data object x_i* **do**
    Determine the closest cluster C_l for x_i
    Increment Lower(C_l) and Upper(C_l) by $\mu_{C_l}(x_i)$
    foreach *Cluster $C_m \neq C_l$ with $|d(x_i, C_m) - d(x_i, C_l)| \leq \varepsilon_{\text{dist}}$* **do**
        Increment Upper(C_m) by $\mu_{C_m}(x_i)$
    end
end

Fig. 4 Fuzzy *RECA* – calculation of cluster Lower and Upper Approximations

distance calculations, data object is assigned to lower and upper approximation of the closest cluster (center) $d(x_i, C_l)$. Additionally, if difference between the distance to other cluster center(s) and the distance $d(x_i, C_l)$ is less than predefined distance threshold $\varepsilon_{\text{dist}}$ – this data object is additionally assigned to this cluster approximations. Approximations are increased by fuzzy membership value of the given data object to the cluster center. Fuzzy RECA algorithm flow is the same as presented in Figs. 2 and 3 with the exception of lower and upper approximation calculation that follows steps presented in Fig. 4. Exemplary borders for Fuzzy RECA algorithm, that are the same as in case of RECA algorithm have been presented in Fig. 1b compared to crisp borders in Fig. 1a.

2.4 Fuzzy Threshold RECA

In Fuzzy Threshold RECA, membership values of analyzed data object to clusters centers are computed. Afterwards, data object is assigned to lower and upper approximation of the closest cluster relative to membership value to this cluster center $\mu_{C_l}(x_i)$. Additionally, if fuzzy membership value to other cluster(s) not exceeds predefined fuzzy distance threshold $\varepsilon_{\text{fuzz}}$ – this data object is also assigned to this cluster approximations. Approximations are increased by fuzzy membership value of the given data object to the cluster(s). Lower and upper approximation calculation

---

foreach *Data object x_i* **do**
    Determine the closest cluster C_l for x_i
    Increment Lower(C_l) and Upper(C_l) by $\mu_{C_l}(x_i)$
    foreach *Cluster $C_m \neq C_l$ with $\mu_{C_m}(x_i) \geq \varepsilon_{\text{fuzz}}$* **do**
        Increment Upper(C_m) by $\mu_{C_m}(x_i)$
    end
end

Fig. 5 Fuzzy Threshold *RECA* – calculation of cluster Lower and Upper Approximations

follows steps presented in Fig. 5. Exemplary borders for Fuzzy Threshold RECA have been presented in Fig. 1c.

3 Image Segmentation Evaluation

In general, three distinctive approaches to cluster validity are possible. The first approach relies on external criteria that investigate the existence of some predefined structure in clustered data set. The second approach makes use of internal criteria and the clustering results are evaluated by quantities describing the data set such as proximity matrix etc. Approaches based on internal and external criteria make use of statistical tests and their disadvantage is high computational cost. The third approach makes use of relative criteria. In this research, the following two measures are taken into account.

3.1 *Quantitative Measure: β-Index*

Measure in the form of β-index denotes the ratio of the total variation and within-class variation. Define n_i as the number of pixels in the ith ($i = 1, 2, \ldots, k$) region from segmented image. Define X_{ij} as the gray value of jth pixel ($j = 1, \ldots, n_i$) in the region i and $\overline{X_i}$ the mean of n_i values of the ith region. The β-index is defined in the following way

$$\beta = \frac{\sum_{i=1}^{k} \sum_{j=1}^{n_i} \left(X_{ij} - \overline{X} \right)^2}{\sum_{i=1}^{k} \sum_{j=1}^{n_i} \left(X_{ij} - \overline{X_i} \right)^2} \,, \tag{4}$$

where n is the size of the image and $\overline{X}$ represents the mean value of the image pixel attributes. This index defines the ratio of the total variation and the within-class variation. In this context, important notice is the fact that index-*b* value increases as the increase of k number. The value of β-index should be maximized.

3.2 *Quantitative Measure wVar: Within-Class Variance Measure*

Within-variance measure presents comparatively not complicated measure calculated by summing up within-variances of all clusters:

$$wVar = \frac{1}{k} \times \sum_{i=1}^{k} \sum_{j=1}^{n_i} \frac{1}{n_i} (X_{ij} - \overline{X}_i)^2. \tag{5}$$

This measure values are presented in experiments carried out on building image. Within class variance should be as low as possible and during optimalization this value should be minimized.

4 Experimental Setup and Results

In the experiments color satellite RGB image has been taken as input image, as shown in Fig. 6: buildup area image. In case of RECA, Fuzzy RECA and Fuzzy Threshold RECA 6 populations each of $Size = 30$ chromosomes have been created and the best solution presented in subsequent tables relative to β-index and within-variance measure. Number of clusters k has been set to 8 clusters. In the experiments distance threshold $\varepsilon_{\text{dist}}$ for RECA has been put to value 50, fuzzy threshold $\varepsilon_{\text{fuzz}}$ has been put to value 0.20. Fuzzifier value μ has been set to 2.5. Additionally, the best solutions from referenced k-means algorithm are also presented from 30 independent algorithm runs. Parameters $\varepsilon_{\text{dist}}$ and $\varepsilon_{\text{fuzz}}$ have been selected on experimental basis but it is possible to select thresholds that yield the maximum value of rough entropy for the given cluster centers. Experimental results for RECA algorithm, Fuzzy RECA and Fuzzy Threshold RECA results are given in Table 1.
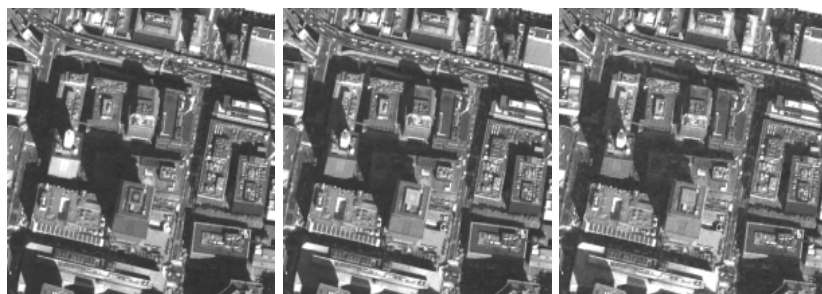
Fig. 6 Satellite buildup area image – Red, Green, and Blue channels

Table 1 Experimental results – Buildup area image – RECA, Fuzzy RECA and Fuzzy Threshold RECA in selected combinations of R, G, and B channels

Name	KMEANS		RECA		Fuzzy RECA		Fuzzy Threshold RECA	
	wVar	β-index	wVar	β-index	wVar	β-index	wVar	β-index
R-G	109.81	34.06	108.99	34.35	109.40	34.20	114.30	34.13
R-B	137.10	26.86	137.74	26.74	137.63	26.76	138.77	26.54
G-B	102.24	35.84	83.17	44.05	83.48	43.90	84.65	43.31
R-G-B	160.12	28.65	166.00	27.78	162.65	28.28	177.47	25.62

5 Conclusions

High quality of image segmentation requires incorporating reasonably as much information of an image as possible. This kind of combining diverse information in a segmentation understood as a means of improving algorithm performance has been widely recognized and acknowledged in the literature of the subject. New algorithmic schemes RECA and Fuzzy RECA extend application of available image information into rough and fuzzy sets setting that possibly takes more powerful and complete comprehension of image characteristics. In the same context, proper Fuzzy RECA parametrization contributes into deeper insight into fuzzy and rough image properties, that should improve image segmentation and further image analysis for acquired imagery. In order to make rough entropy clustering more robust, completely new approaches into parametrization of rough entropy computation have been elaborated. Proposed RECA algorithm parametrization has been described and experimental results presented together with algorithm performance comparison to reference k-means algorithms. Experimental results suggest that RECA algorithmic schemes are generally comparable to k-means solutions.

References

1. Malyszko, D., Stepaniuk, J.: Granular multilevel rough entropy thresholding in 2d domain. In: Proceedings of the 16th International Conference on Intelligent Information Systems, pp. 151–160. Zakopane, Poland (2008)
2. Malyszko, 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, vol. 5306, pp. 409–418. Springer, Heidelberg (2008)
3. Pal, S.K., Shankar, B.U., Mitra, P.: Granular computing, rough entropy and object extraction. Pattern Recognition Letters 26(16), 2509–2517 (2005)
4. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007)
5. Pedrycz, W., Skowron, A., Kreinovich, V. (eds.): Handbook of Granular Computing. John Wiley & Sons, New York (2008)
6. Stepaniuk, J.: Rough–Granular Computing in Knowledge Discovery and Data Mining. Springer, Heidelberg (2008)

Data Grouping Process in Extended SQL Language Containing Fuzzy Elements

Bożena Małysiak-Mrozek, Dariusz Mrozek, and Stanisław Kozielski

Abstract. Incorporation of fuzziness into database systems allows to expand the analysis of data during the querying process. Approximate processing of data includes not only rows that exactly meet the search criteria, but also rows that are similar with the given range of tolerance. Queries formulated in natural language usually consist of imprecise and fuzzy terms. However, the implementation of the imprecision in query languages, such as SQL, requires additional extensions in a target database system. In the paper, we devote our attention to different methods of data grouping. In the first part of the paper we concentrate on the process of fuzzy grouping of crisp data. In the second part, we focus on the grouping of fuzzy data.

Keywords: databases, SQL, fuzzy sets, fuzzy grouping, fuzzy values.

1 Introduction

Queries submitted to databases are often first formulated in a natural language. Afterwards, these queries are translated to database query languages, such as SQL [7, 21]. Imprecise and fuzzy terms are typical to appear in a natural language. This is characteristic for human's perception of things that happen in the surrounding world. However, it is very hard to implement it in database query languages. The fuzzy logic [26, 24] can be applied not only in human way of thinking, but also in databases that store data related to many domains and query languages, e.g., the SQL, operating on the data. The fundamental principle of the fuzzy logic theory is

Bożena Małysiak-Mrozek · Dariusz Mrozek · Stanisław Kozielski

Institute of Informatics, Silesian University of Technology,

Akademicka 16, 44-100 Gliwice, Poland

e-mail:{Bozena.Malysiak,Dariusz.Mrozek,
Stanislaw.Kozielski}@polsl.pl

that everything is true, but with different degree of compatibility [6, 18]. In general, the fuzzy logic can help to improve results of SQL queries. Fuzzy queries provide a simple way to retrieve data that we want to obtain without defining exact search criteria [22, 1, 12]. In huge databases this approach can be very useful and such implementation can significantly speed up the retrieval process [1, 12]. This process may consist of two stages. In the first stage, we perform a quick data search, which provides a set of data that meet search criteria at least with the given compatibility degree. Afterwards, in the next stage, the user can quickly find the most suitable data from the result set created in the first stage.

Implementation of fuzzy queries in database systems is very important for the retrieval of various types of data, such as text, image, video, audio, graphics, and animation [19]. Studies in the area have been performed for the last two decades by several research centers. E.g., in works [9] and [3], we can find proposals of querying systems (FQUERY and SQLf, respectively) that support imprecise queries. In the work [20], Tang and Chen describe fuzzy relational algebraic operators useful for designing fuzzy query languages. The usage of different membership functions in processing fuzzy SQL queries is discussed in authors' work [17].

Indeed, the SQL language is the most popular language for retrieving and modifying data stored in databases [7], [21]. Fuzzy terms can occur in all parts of the SQL SELECT statement [12, 25, 4]. In the paper, we concentrate on the GROUP BY phrase of the SELECT statement. In the following sections, we present different methods of fuzzy grouping and grouping of fuzzy data. Some of them base on already known methods and some of them are completely novel. We also show our implementation of these grouping methods in the SQL language (PostgreSQL DBMS). All issues considered in this work are illustrated by appropriate examples.

2   Data Grouping

Data grouping is often a required step in the aggregation process. In the SQL language notation, the GROUP BY phrase is responsible for the grouping process. The process groups rows with identical values in columns specified in the GROUP BY phrase, so each unique combination of values in specified columns constitutes a separate group [7, 21]. This is the classical grouping. If we allow to group similar (not identical) data in grouping columns, we can use methods of fuzzy grouping. If there is a possibility to store fuzzy data in a database, these data can be grouped with the use of classical or fuzzy grouping methods. There are different approaches to the data grouping that can be used if we incorporate the fuzziness into the database system:

- fuzzy grouping of crisp values,
- classical grouping of fuzzy values,
- fuzzy grouping of fuzzy values.

Table 1 *Measurements* table and linguistic values of the *temperature* after the assignment step

Date	Temperature	Linguistic value
02.07.2003	17	quite_warm
08.07.2003	18	warm
11.07.2003	14	quite_warm
14.07.2003	25	very_warm
21.07.2003	29	very_warm
25.07.2003	30	very_warm
31.07.2003	15	quite_warm

2.1 Fuzzy Grouping of Crisp Values

In the classical grouping of crisp values, the smallest difference between values in grouping column(s) leads to the separation of groups [7, 21]. If we want to group similar data, it is necessary to apply mechanisms of fuzzy grouping, which join similar data into the same groups [12]. In our work, we have extended the SQL language by implementing the following algorithms of fuzzy grouping:

- grouping with respect to linguistic values determined for the attribute domain,
- fuzzy grouping based on the hierarchical clustering method,
- fuzzy grouping based on the author's algorithm.

2.1.1 Grouping with Respect to Linguistic Values

This method of grouping can be applied if domains of grouped attributes can be described by linguistic values [12, 9, 2]. These values can be defined by membership functions [18, 17]. The grouping process consists of two steps: In the first step, we assign the most corresponding linguistic values (having the highest value of the compatibility degree) to existing numerical values. Afterwards, we group data according to assigned linguistic values. The number of groups is equal to the number of different linguistic values.

Example 1. Let's assume, there is the *Measurements* table in the database. The table includes the *temperature* attribute, which stores values of temperature for particular days, and temperatures are given in Celsius degrees (Table 1, left columns). For the temperature attribute we define the set of linguistic values describing possible temperatures: very cold, cold, warm, very warm, etc.. Each of these values is determined by appropriate membership function. In this case, we assign the most corresponding linguistic value to each value of the temperature column (Table 1, right column). In the next step, we run the process of classical grouping with respect to assigned linguistic values. Actually, both steps are executed in one SQL statement.

Let's consider the following query:

Display number of days with similar temperatures.

This query written in the extended SQL language can have the following form:

Table 2 Query results

Temperature	Days_No
quite_warm	3
warm	1
very_warm	3

```
SELECT temperature_to_ling(temperature) as temperature,
    COUNT(date) as Days_No
FROM Measurements
GROUP BY temperature_to_ling(temperature);
```

where the *temperature_to_ling* function converts an original, numerical value of the *temperature* column into the most corresponding linguistic value of the temperature. The result of such formulated query is presented in Table 2.

2.1.2 Modified Hierarchical Clustering Method in Data Grouping

In the previous case, the assignment to a group was strictly related to the division of the attribute domain. However, in many cases, it could be interesting to join in groups many similar values, while the division of the attribute domain is not predefined. The approach is similar to the problem of cluster separation in clustering algorithms [23, 10].

Let's make the assumption we perform clustering of N data $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N$. The purpose of the process is to obtain M clusters. The classical hierarchical clustering algorithm contains the following steps:

1. In the beginning, we assume each separate data $\underline{x}_i$ creates single cluster $K_i = \{\underline{x}_i\}$, $i = 1, \ldots, N$; the number of clusters is $l := N$;
2. We find two nearest clusters K_k and K_j using one of the distance measures defined later in the section;
3. Next, clusters K_k and K_j are joined and create the new cluster K_k; Cluster K_j is being removed; Number of clusters decreases $l := l - 1$;
4. If $l > M$, go to point 2.

Presented algorithm can be applied to group data in databases. To this purpose, we need to specify the meaning of data $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N$. In the SQL language, data grouping is implemented in the GROUP BY phrase, which can be presented in the following form:

```
GROUP BY <A1, A2, ..., Ap>,
```

where $A_1, A_2, \ldots, A_p$ are attributes (column names) of the table T, on which the grouping process is performed. A single data $\underline{x}$ is formed by a set of values of attributes $A_1, A_2, \ldots, A_p$ coming from a single row of the table T. Therefore, data

$\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N$ represent sets of attributes' values involved in the grouping, for successive rows of the table T, and the N corresponds to the number of rows being grouped.

In order to implement the presented algorithm, it is required to make the assumption that it is possible to define a function of distance between $\underline{x}_i$ and $\underline{x}_j$ (and generally – between clusters) in the multidimensional space determined by data $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N$. The distance may be calculated as [5]:

1. the minimum distance between any data of clusters A and B:

$$d_{\min}(A, B) = \min_{\underline{x}_A \in A, \underline{x}_B \in B} |\underline{x}_A - \underline{x}_B|, \tag{1}$$

2. the maximum distance between any data of clusters A and B:

$$d_{\max}(A, B) = \max_{\underline{x}_A \in A, \underline{x}_B \in B} |\underline{x}_A - \underline{x}_B|, \tag{2}$$

3. the arithmetic average of all distances between all data in clusters A and B:

$$d_{\text{avg}}(A, B) = \frac{1}{\text{card}(A)\text{card}(B)} \sum_{\underline{x}_A \in A} \sum_{\underline{x}_B \in B} |\underline{x}_A - \underline{x}_B|, \tag{3}$$

4. the distance between central point (average value) $\underline{m}_A$ of the cluster A and the central point (average value) $\underline{m}_B$ of the cluster B:

$$d_{\text{mean}}(A, B) = |\underline{m}_A - \underline{m}_B|. \tag{4}$$

In the presented algorithm, the number of groups should be defined in advance. This may be uncomfortable and unacceptable in many cases. For this reason, we decided to modify the algorithm by incorporating different stop condition. The condition should not be associated with the number of groups. Imposing restrictions on the maximal size of a group seems to be a better solution. This maximal size can be defined as a maximum distance between any two data in the group. In the modified version of the algorithm we have replaced the *cluster* concept by the *group* concept. As in previous case, we assume we want to group N data $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N$. However, in this case, the purpose of the grouping process is to obtain groups, which size does not exceed the given value *maxd*. The maximal group size *maxd* is usually defined by the domain expert.

2.1.3   The Algorithm of Hierarchical Grouping

The modified algorithm has the following steps:

1. In the beginning, we assume each data $\underline{x}_i$ creates separate group $G_i = \{\underline{x}_i\}$, $i = 1, \ldots, N$, $N > 1$. The set of all groups is represented by $G = \{G_1, G_2, \ldots, G_N\}$.
2. In the set G, we find two nearest groups G_k and G_j. Next, these two groups are joined together into G'_k group. The size d of the G'_k group is computed (maximum

distance between any two data in the group). If $d > maxd$ then cancel the G'_k group and go to 4.
3. Replace G_k group with the G'_k group. Delete group G_j from group-set G. Go to step 2.
4. Stop grouping – the G set consists of created groups.

We also considered other stop conditions. E.g., while joining groups, the distance between central points rises. For this reason, the grouping may be finished when the distance between central points of two nearest groups exceeds a given value. We have implemented the presented algorithm of hierarchical grouping in the DBMS PostgreSQL [8]. Since all data are sorted before the grouping process, we could simplify the implemented algorithm described above. In the implemented version, for sorted data, it is enough to evaluate the distance between the first and consecutive elements of the group. Each calculated distance should not exceed the specified value of the *maxd*. The implemented algorithm of hierarchical grouping is presented in Fig. 1.

With the use of the implemented algorithm of hierarchical grouping we can solve problems presented below. Let's assume there is a *Students* table in a database. The table stores the information about the height and the weight of students on particular semesters. The table has the following schema:
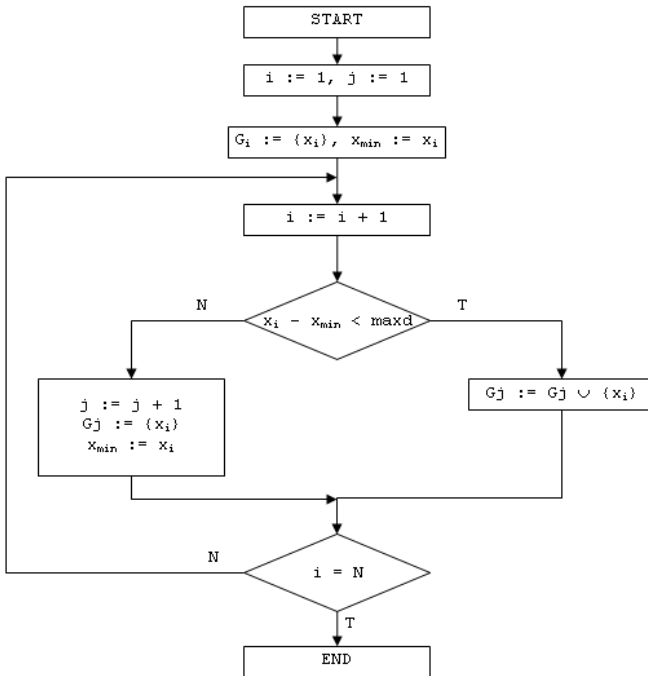
Fig. 1 The implemented algorithm of hierarchical grouping

```
Students (StdNo, StdName, semester, height, weight)
```

We will consider the following query:

Display the number of students with similar weight in particular semesters.

We assume the *similar* means the weight differs no more than 5kg in the same group. The query written in our extended SQL language has the following form:

```
SELECT semester, weight, COUNT(StdNo)
FROM Students
GROUP BY semester, surroundings(weight, 5);
```

The *surroundings* function evaluates distances between data in groups and decides where the border of a group is and which data belong to consecutive groups.

2.2   Grouping of Fuzzy Data

Grouping of fuzzy data is required only in database systems that allow to store fuzzy data. Analysing possibilities of grouping according to attributes that store fuzzy values, we can first consider the same grouping methods that operate on crisp data. In the case, we assume the grouping process is performed on LR-type fuzzy numbers [24, 18]. In the most rigorous solution, the grouping allows for all parameters describing fuzzy values. Therefore, selected fuzzy values are joined into one group, if they have the same set of parameters. More elastic solutions allow grouping of fuzzy data regarding their modal values. We can distinguish two approaches:

- grouping of LR-type fuzzy numbers – all fuzzy numbers with the same modal value form a group,
- grouping of LR-type fuzzy intervals – all fuzzy intervals with the same range of modal values become a group.

In our work, we have implemented the first approach. The example shows how it can be used. We assume there is a *Requirement* table in a database:

```
Requirement(DeptNo, Year, Paper, Toner, CDs)
```

The table stores the information about requirements of particular departments for paper, toner and CDs in particular years. The data type for *Paper*, *Toner*, and *CDs* attributes is the *fTrapezium*, which is a structure of four parameters describing trapezoidal membership function. We consider the following query:

Display the number of departments that make requirements for similar amount of paper in particular years.

The query in the extended SQL language has the following form:

```
SELECT Year, count(distinct DeptNo)
FROM Requirement
GROUP BY Paper, Year;
```

We can also apply all methods of fuzzy grouping described in Sect. 2.1 for the LR-type fuzzy data. In the situation, we deal with the process of fuzzy grouping

of fuzzy data. The process can be implemented with the use of arbitral division of a fuzzy attribute domain (linguistic values) and modified algorithm of hierarchical grouping.

Finally, we can consider specific situation that does not fit any of the given categories. The fuzziness may occur indirectly in the GROUP BY clause, e.g., if we want to group by the compatibility degree of fuzzy condition. We assume there is a *Departments* table in a database:

```
Departments(DeptNo, DeptName, NOEmployees)
```

The table stores the information about the number of employees working in particular departments. Let's consider the following query:

Display number of departments with the same compatibility degree for the fuzzy condition: the number of employees is about 25.

The query in the extended SQL language has the following form:

```
SELECT count(DeptNo), NOEmployees is about 25
FROM Departments
GROUP BY NOEmployees is about 25;
```

The query groups together rows with the same compatibility degree for the specified condition. It differs from the previous cases where values of the given attributes were grouped.

3   Concluding Remarks

Data grouping is one of the most important operations performed in all database systems. We have developed extensions to the SQL language that allow to process fuzzy data or operate on data with the use of fuzzy techniques. The main advantage of the extended GROUP BY phrase presented in the paper is its simplicity. This allows to write SQL statements with the minimum knowledge of the fuzzy sets theory. However, do not let the simplicity trip you up. The huge power of our SQL extensions including grouping process is hidden behind all functions that had to be implemented during the development process. Especially, the hierarchical grouping method required overloading of comparison operators, which is not possible in many popular DBMSs.

Our extensions to the SQL cover not only data grouping, but also fuzzy filtering in the WHERE clause, fuzzy terms in subqueries, and fuzzy aggregation in the SELECT and HAVING clauses. However, these were the subject of our previous papers, e.g., [14, 16, 13]. The grouping methods that we presented in the paper can be used for reporting purposes in conjunction with the classical or fuzzy aggregation. When we use standard grouping methods we can find data too detailed to make a reasonable analysis. Fuzzy grouping allows to generalize data in order to make better conclusions. Therefore, a conscious loss of some information results in better analysis and observations of interesting relations between data. We have recently implemented presented grouping methods in the dating agency web service [15] and in the database system registering missing and unidentified people [11]. In the last

case, presented grouping methods help to make analysis, such as which age group is mostly at risk of going missing or unidentified (e.g., old people).

Acknowledgement

References

1. Badurek, J.: Fuzzy logic in databases. Informatyka (1999)
2. Bordogna, G., Pasi, G.: A fuzzy query language with a linguistic hierarchical aggregator. ACM, New York (1994)
3. Bosc, P., Pivert, O.: SQLf: A relational database language for fuzzy querying. IEEE Transactions on Fuzzy Systems 3(1) (1995)
4. Chen, S.M., Chen, H.H.: Fuzzy query processing in the distributed relational databases environment. In: Database and Data Communication Network Systems, vol. 1. Elsevier Science, Amsterdam (2002)
5. Czogała, E., Łęski, J.: Fuzzy and neuro-fuzzy intelligent systems. Phisical-Verlag, Heidelberg (2000)
6. Dubois, D., Prade, H.: Fuzzy Sets and Systems. Academic Press, New York (1988)
7. Elmasri, R., Navathe, S.B.: Fundamentals of Database Systems. World Student. Addison-Wesley Publishing Company, Reading (2000)
8. Group, P.G.D.: PostgreSQL 7.2 Programmer's guide (2001)
9. Kacprzyk, J., Ziółkowski, A.: Database queries with fuzzy linguistic quantifiers. IEEE Transactions of Systems, Man, and Cybernetics 16(3) (1986)
10. Kolatch, E.: Clustering algorithms for spatial databases: a survey. University of Maryland (2001)
11. Małaczek, P., Małysiak, B., Mrozek, D.: Searching missing and unidentified people through the Internet. In: Kwiecień, A., et al. (eds.) Computer Networks: Applications, pp. 237–249. Wydawnictwa Komunikacji i Łączności, Warsaw (2007)
12. Małysiak, B.: Approximate retrieval methods in database systems. Ph.D. thesis, Silesian University of Technology, Gliwice, Poland (2003)
13. Małysiak, B.: Fuzzy values in nested SQL queries. Studia Informatica 25(2), 113–126 (2004)
14. Małysiak, B.: Interpretation of filtering conditions in SQL queries. Studia Informatica 25(2(58), 89–101 (2004)
15. Małysiak, B., Bieniek, S.: Internet as a medium supporting establishing relationships between people. In: Proceedings of the Computer Networks Conference. Wydawnictwa Komunikacji i Łączności, Warsaw (2005)
16. Małysiak, B., Mrozek, D.: Fuzzy aggregation in SQL queries. In: Databases: Models, Technologies, Tools, vol. 2, pp. 77–84. Wydawnictwa Komunikacji i Łączności, Warsaw (2005)
17. Małysiak, B., Mrozek, D., Kozielski, S.: Processing fuzzy SQL queries with flat, context-dependent and multidimensional membership functions. In: Proceedings of 4th IASTED International Conference on Computational Intelligence, pp. 36–41. ACTA Press, Calgary (2005)

18. Piegat, A.: Fuzzy Modeling and Control. Exit, Warsaw, Poland (1999)
19. Swain, M., Anderson, J.A., Swain, N., Korrapati, R.: Study of information retrieval using fuzzy queries. In: Proceedings of the IEEE SoutheastCon, pp. 527–533 (2005)
20. Tang, X., Chen, G.: A complete set of fuzzy relational algebraic operators in fuzzy relational databases. In: Proceedings of the IEEE International Conference on Fuzzy Systems, pp. 565–569 (2004)
21. Ullman, J.D.: Database and knowledge-base systems. Computer Science Press (1988)
22. White, D.A., Jain, R.: Algorithms and strategies for similarity retrieval. Visual Computing Laboratory, University of California (1997)
23. Wojciechowski, K.: Image processing and pattern recognition. Publishers of the Silesian University of Technology, Gliwice, Poland (1992), Script no. 1662
24. Yager, R.R., Filev, D.P.: Essentials of Fuzzy Modeling and Control. John Wiley & Sons, New York (1994)
25. Yu, C.T., Meng, W.: Principles of Database Query Processing for Advanced Applications. Morgan Kaufmann Incorporated, San Francisco (1998)
26. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)

Rough Sets in Flux: Crispings and Change

Marcin Wolski

Abstract. The paper aims to initiate the research upon a new conceptualisation of rough set theory (RST), which allows one to manage the change of the background knowledge. In the classical approach to RST, a rough set is defined as a pair $(\underline{A}, \overline{A})$ of definable (crisp) subsets (of some universe U), called the lower and the upper approximation of $A \subseteq U$, respectively. Since both approximations are idempotent, a rough set is fixed (static). By mimicking the behaviour of a topological praclosure operator, we shall generalise the above ideas and introduce a notion of a crisping concept, which will set the stage for rough sets in flux. For each crisping concept we distinguish its generalised rough set, which can be modified by means of dynamic operators such as expansion, revision or contraction (but within the scope of the crisping concept). Since this approach can cope with the change of knowledge, it may be regarded as a step towards approximating processes (i.e., sequences of changes).

Keywords: rough set, knowledge change, approximation operator.

1 Introduction

The treatment of incomplete and imprecise data has been considered essential for real world applications. As a response to this challenge, Pawlak introduced rough set theory (RST) [6, 7] and started a number of research results, which has brought effective techniques for dealing with uncertainty and vagueness in relational databases. As with other branches of computer science, RST has grown beyond the circumstances of its birth and today it has gained a great deal of attention from (both pure and applied) mathematics, biology, geography and so on.

Marcin Wolski
Department of Logic and Philosophy of Science, Maria Curie-Skłodowska University,
Marie Curie-Skłodowska Square 5, 20-031 Lublin, Poland
e-mail: marcin.wolski@umcs.lublin.pl

There are actually a number of different conceptualisations of RST. In this paper we focus upon a very elementary approach to RST, which is based on the concepts of an approximation space and approximation operators. As usual, by an approximation space we mean a non-empty set U supplied with an equivalence relation E, and sets built up from some equivalence classes of E are called definable (crisp) sets. The key idea staying behind RST is to approximate any undefinable set $A \subseteq U$ by two definable (crisp) sets: $\underline{A}$ and $\overline{A}$, called the lower and the upper approximation, respectively. A rough set is then defined as a pair $(\underline{A}, \overline{A})$, for any $A \subseteq U$. Hereafter several attempts have been made to generalise these ideas, yet all these attempts have regarded a rough set as a static object.

In order to link rough sets with the phenomenon of change, we shall mimic the behaviour of a topological praclosure operator. Specifically, a praclosure is not idempotent and thus it make sense to iterate this operator. As a consequence, for any set $A \subseteq U$ it may generate a number of its upper approximations. Our idea is to copy this (non-idempotent) behaviour and paste into an approximation space (whose operators are – of course – idempotent). In order to do so, we introduce the notion of a crisping concept defined as a family of crisp sets, which additionally is linearly ordered by the set theoretic inclusion. As we show, such defined crisping concepts bring us a very suitable stage for rough sets in flux. For each crisping concept we distinguish its generalised rough set, which can be modified by means of dynamic operators such as expansion, revision or contraction (but within the scope of the crisping concept). Our modelling of change operators is based upon the theory of belief dynamics [2, 3]; however, due to different interpretation of the underlying universe we have to reverse the order which picks up the appropriate (with respect to a given operator) sets.

2 Rough Sets and Crispings

This section recalls basic concepts from rough set theory RST [6, 7]. We start by introducing the concepts of an approximation space and approximation operators. On the basis of some generalisations of these notions we shall introduce a notion of a crisping concept, which will set the stage for rough sets in flux.

Definition 1 (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 $A = \bigcup \mathscr{B}$ for some $\mathscr{B} \subseteq U/E$, where U/E is the family of equivalence classes of E.

Definition 2 (Approximation Operators). Let (U, E) be an approximation space. For every concept $A \subseteq U$, its E-lower and E-upper *approximations*, $\underline{A}$ and $\overline{A}$, respectively, are defined as follows:

$$\underline{A}_E = \{a \in U : [a]_E \subseteq A\},$$

$$\overline{A}^E = \{a \in U : [a]_E \cap A \neq \emptyset\}.$$

The chief idea of RST is to approximate any set A by means of two definable (crisp) sets: $\underline{A}_E$ and $\overline{A}^E$. The lower approximation $\underline{A}_E$ consists of objects which necessarily belong to A, whereas the upper approximation $\overline{A}^E$ consists of objects which possibly belong to A. For any $A \subseteq U$, the pair $(\underline{A}_E, \overline{A}^E)$ is called a *rough set*. Observe that for any definable (crisp) set A, it holds that $\underline{A}_E = \overline{A}^E$. Furthermore, both approximation operators are idempotent:

$$\underline{\underline{A}_E}_E = \underline{A}_E \qquad \overline{\overline{A}^E}^E = \overline{A}^E.$$

Hereafter several attempts has been made to generalise the concept of approximation operators, e.g., [4]. Basically, instead of an equivalence E one can take any other binary relation R and define:

$$\underline{A}_R = \{a \in U : [a]_R \subseteq A\},$$

$$\overline{A}^R = \{a \in U : [a]_R \cap A \neq \emptyset\},$$

where $[a]_R = \{b \in U : (a,b) \in R\}$.

Proposition 1. *If R is a preorder (i.e., a reflexive and transitive relation) then both $\underline{A}_R$ and $\overline{A}^R$ are idempotent. More precisely, $\underline{A}_R$ is a topological closure, whereas $\overline{A}^R$ is a topological interior.*

Actually, it is the transivity condition which enforces the idempotency of the corresponding operators. So let us drop out this condition and consider $\overline{A}^R$ where R is only a reflexive relation. It turns out that a such defined operator has many interesting properties.

Definition 3 (Praclosure Operator). An abstract praclosure operator $[\ ]_p$ on a set U is a map which associates to each set $A \subseteq U$ a set $[A]_p$ such that:

1. $[\emptyset]_p = \emptyset$,
2. $A \subseteq [A]_p$ for all $A \subseteq U$,
3. $[A \cup B]_p = [A]_p \cup [B]_p$ for all $A \subseteq U$ and $B \subseteq U$.

Let us consider two simple examples. As is well known, a sequence of points $\{a_n\}_{n \in \mathbb{N}}$ of a topological space (U, τ) converges to a point a if each neighbourhood O_a of a contains all terms of the sequence after some fixed term. For any topological space (U, τ) and $A \subseteq U$ let $[A]_{seq}$ be the set of all points $a \in U$ for which there exists a sequence in A converging to a. Then a map which associates to each $A \subseteq U$ a set $[A]_{seq}$ is a praclosure operator on U. In the topological literature $[\ ]_{seq}$ is known as a sequential closure. For another simple example, let ρ be a prametric on U and $[A]_p = \{a \in U : \rho(a,A) = 0\}$, for each $A \subseteq U$. Then $[\ ]_p$ is a praclosure operator on U. For more details about praclosure operators see, e.g., [1].

Proposition 2. *Let R be a reflexive relation over U, then $\overline{A}^R$ is a praclosure operator on U.*

Since $[\ ]_p$ is not idempotent, iteration of the praclosure operator, generally speaking, may give a new set; that is, the set $[[A]_p]_p$ can differ from $[A]_p$. Thus for a set A and a reflexive relation R we can generate a sequence of its upper approximations:

$$\overline{A}^R \subseteq \overline{\overline{A}^R}^R \subseteq \overline{\overline{\overline{A}^R}^R}^R \subseteq \cdots \tag{1}$$

So, instead of two approximations, we may deal with a family of approximations ordered by the set theoretic inclusion $\subseteq$. In [5] there are considered approximations (rough sets) of different precision. Here we would like to consider an approximation (a crisping concept) with immanent different precision. In order to do so, we shall mimic the behaviour of a non-idempotent praclosure operator $[\ ]_p$. Since for any $A \subseteq U$ the behaviour of $[A]_p$ over A can described as a sequence such like (1), for R being a preorder, as a 'description' of (non-idempotent-like) behaviour of $\overline{A}^R$ (which is idempotent) over A we take a finite sequence

$$A_1 \subseteq A_2 \subseteq A_3 \subseteq \cdots \subseteq A_n,$$

where $A_i = \overline{A_i}^R$ for every i and $A \subseteq A_1$. For an approximation space we can generalise the above idea as follows.

Definition 4 (Crisp Concept). By a *crisping concept* $\mathscr{C}$ over an approximation space (U, E) we mean a finite family of definable sets $\{A_1, \ldots, A_n\}$ of (U, E), which is linearly ordered by the set inclusion: $A_1 \subseteq A_2 \subseteq \cdots \subseteq A_n$. Elements A_i of a crisping concept $\mathscr{C}$ are called *crispings*.

The least element of $\mathscr{C}$ is the least possible lower approximation whereas the greatest elements of $\mathscr{C}$ is the greatest possible upper approximation. These two sets establish the borders of all possible changes of $\mathscr{C}$. In the next section we show that crisp concepts provide a very simple stage where one can introduce the process of change into RST.

3   Crispings and Change

In the previous section we have introduced the notion of a crisping set $\mathscr{C}$. In this section we show how to 'embed' rough sets into this framework and make them changing with respect to knew information. The operations defined below are inspired by belief dynamics (the classic textbook is [2], for a nice overview see, e.g., [3]). However, due to different interpretations of the universe (in belief dynamics the universe consists of all maximally consistent theories), we usually use the opposite order and thus we obtain operators with different properties.

Definition 5. A *generalised rough set* (or simply a *rough set*) of a crisping concept $\mathscr{C} = \{B_1, \ldots B_n\}$ is a pair of its elements (A, C) such that $A, C \in \mathscr{C}$ and $A \subseteq C$. We call A the lower approximation of $\mathscr{C}$, whereas C is called the upper approximation of $\mathscr{C}$.

Fig. 1 Crisping concept 'safe car', e.g., $\mathscr{C} =_A (A,B,C)^C$, and new pieces of knowledge 1 (Swedish cars) and 2 (French cars)
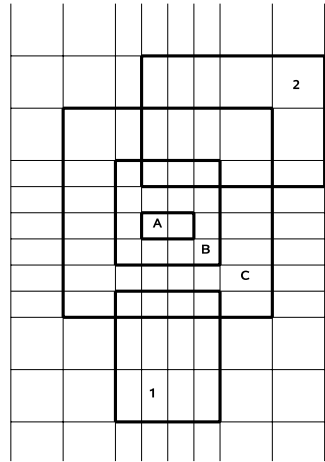
In other words, by approximations we mean a pair of crispings: the smaller one is the lower approximation and the bigger one is the upper approximation. Since the lower and the upper approximations are important, we shall represent any crisping concept $\mathscr{C} = \{B_1, \dots B_n\}$, whose the lower and the upper approximations are A and C, respectively, as:

$$_A\{B_1, \dots, B_n\}^C.$$

We say that a crisping concept $_A\{B_1, \dots, B_n\}^C$ approximates $S \subseteq U$ if $A \subseteq S \subseteq C$. If the underlying crisping concepts is fixed, we shall say that A and C approximate S. By boundary of $\mathscr{C} =_A \{B_1, \dots, B_n\}^C$ we mean $Bnd(\mathscr{C}) = C \setminus A$. A crisping concept $\mathscr{C} =_A \{B_1, \dots, B_n\}^C$ is called *exact* iff $A = C$, in other words, iff $Bnd(\mathscr{C}) = \emptyset$.

When obtaining new knowledge, we would like to change (in an appropriate way) a rough set of a given crisping concept $_A\{B_1, \dots, B_n\}^C$, that is, we would like to modify its boundary. Suppose, that we approximate a concept 'safe car' by means of two sets of cars A (the lower approximation) and B (the upper approximation), see Fig. 1. Then we are said that Swedish cars (represented by 1) have a very good opinion. How can we take this information into account? On the other hand, suppose that 'safe car' is approximated by A and C. Now, we are said that French cars (represented by 2) are safe. Should we change the original rough set?

Following proposals from belief dynamics [2, 3] we shall suggest below a few types of basic changes.

3.1 Expansion

Let $\mathscr{C} =_A (B_1, \dots, B_n)^C$ be a crisping concept and I a definable set (of an approximation space (U, E)). Now, we define the *expansion* of $_A(B_1, \dots, B_n)^C$ by I, which will be denoted by $_A(B_1, \dots, B_n)^C + I$. Let I^+ be the maximal (with respect to $\subseteq$)

$B_I \in \mathscr{C}$ such that $B_i \cap I \neq \emptyset$. If I^+ exists for a pair $\mathscr{C}$ and I then we say that I is a relevant concept for an expansion of $\mathscr{C}$, or better still, we say that I is relevant. If I is not relevant then it does not affect the crisping concept. If I is relevant then

$$
\begin{aligned}
{}_A(B_1,\ldots,B_n)^C + I &= {}_{I^+}(B_1,\ldots,B_n)^C && \text{iff } I^+ \subseteq C, \\
{}_A(B_1,\ldots,B_n)^C + I &= {}_A(B_1,\ldots,B_n)^{I^+} && \text{otherwise.}
\end{aligned}
$$

In order to keep track of the history of $\mathscr{C}$ we shall also write $A+I$ or $C+I$ instead of I^+ thus, e.g. if $I^+ \subseteq C$ then

$$
{}_A(B_1,\ldots,B_n)^C + I = {}_{I^+}(B_1,\ldots,B_n)^C = {}_{A+I}(B_1,\ldots,B_n)^C.
$$

For example, given the vague concept 'safe car', for Swedish and French cars we obtain (see Fig. 1):

$$
\begin{aligned}
{}_A(A,B,C)^B + 1 &= {}_A(A,B,C)^C = {}_A(A,B,C)^{B+1}, \\
{}_A(A,B,C)^C + 2 &= {}_C(A,B,C)^C = {}_{A+2}(A,B,C)^C.
\end{aligned}
$$

Thus, Swedish cars affect the upper approximation, whereas French cars affect the lower approximation of 'safe car'. It is worth emphasising that in belief dynamics expansion is defined by means of minimal (with respect to $\subseteq$) elements. However, such defined expansion is not very intuitive in the context of rough set theory. Observe also, that the process of expansion involves only part of new information (i.e., not all French or Swedish cars are considered after expansion but some part of them). It means, that our knowledge is considered as well justified and we take into account new information provided that it is at least partly consistent with this knowledge.

3.2 Revision

Now we define the *revision* of ${}_A(B_1,\ldots,B_n)^C$ by I, which will be denoted by ${}_A(B_1,\ldots,B_n)^C * I$. Let I^* denote the maximal B_i such that $I \subseteq B_i$. As above, if such I^* exists then I is called relevant. If I is not relevant then of course the revision brings no change. If I is relevant then

$$
{}_A(B_1,\ldots,B_n)^C * I = {}_{I^*}(B_1,\ldots,B_n)^C = {}_{A*I}(B_1,\ldots,B_n)^C
$$

iff $I^* \subseteq C$, otherwise

$$
{}_A(B_1,\ldots,B_n)^C + I = {}_A(B_1,\ldots,B_n)^{I^*} = {}_A(B_1,\ldots,B_n)^{C*1}.
$$

Thus, the revision process, in contrast to expansion, tries to take into account all pieces of knew information, of course provided that it is consistent with our previous knowledge. For example (see Fig. 1):

$$_A(A,B,C)^C * 1 = {}_A(A,B,C)^C,$$
$$_A(A,B,C,U)^C * 1 = {}_A(A,B,C,U)^U = {}_A(A,B,C)^{C*1},$$

where U, as above, is the universe of the corresponding approximation space (here, it is the set of all cars).

3.3 Contraction

For a crisping concept $\mathscr{C} =_A (B_1,\ldots,B_n)^C$ and a definable set $I \subseteq U$ of (U,E), by $_A(B_1,\ldots,B_n)^C - I$ we denote the *contraction* of $\mathscr{C}$ by I. Let I^- denote the maximal $B_i \in \mathscr{C}$ such that $B_i \cap I = \emptyset$. If such I^- does not exists then I is said to be known for sure, and then the contraction $_A(B_1,\ldots,B_n)^C - I$ brings no change. If I is not known for sure, then

$$_A(B_1,\ldots,B_n)^C - I = {}_A(B_1,\ldots,B_n)^{I^-} = {}_A(B_1,\ldots,B_n)^{C-I} \quad \text{iff } A \subseteq I^-,$$
$$_A(B_1,\ldots,B_n)^C - I = {}_{I^-}(B_1,\ldots,B_n)^C = {}_{A-I}(B_1,\ldots,B_n)^C \quad \text{otherwise.}$$

For example (see Fig. 1):

$$_B(A,B,C)^C - 2 = {}_A(A,B,C)^C = {}_{B-2}(A,B,C)^C,$$
$$_A(A,B,C)^C - 1 = {}_A(A,B,C)^B = {}_B(A,B,C)^{C-1}.$$

Thus, if possible we try to escape from I using the upper approximation, otherwise we escape by means of the lower approximation.

4 Algebraic Generalisation

For the sake of space, here we only briefly discuss an algebraic representation of crisping concepts.

Firstly, we have to make all concepts looking the same, that is, all concepts must have the same length. Then, given two crisping concepts $\mathscr{C} =_A (B_1,\ldots,B_n)^C$ and $\mathscr{D} = {}_D(F_1,\ldots,F_n)^G$ we can define the expansion (and in the same fashion also the revision and contractions) of $\mathscr{C}$ by $\mathscr{D}$, that is $_A(B_1,\ldots,B_n)^C + {}_D(F_1,\ldots,F_n)^G$, in two ways:

$$_A(B_1,\ldots,B_n)^C +_{low} {}_D(F_1,\ldots,F_n)^G = {}_A(B_1,\ldots,B_n)^C + D, \qquad (2)$$
$$_A(B_1,\ldots,B_n)^C +^{up} {}_D(F_1,\ldots,F_n)^G = {}_A(B_1,\ldots,B_n)^C + G. \qquad (3)$$

In both ways, we obtain a family $\mathscr{O}$ of crisping concepts closed under the above introduced dynamic operators. The family $\mathscr{O}_{lower}$ obtained by (2) will be called a lower dynamic ontology, whereas family $\mathscr{O}^{upper}$ obtained by (3) will be called an upper dynamic ontology. Now define:

$$A(B_1,\ldots,B_n)^C \cup {}_D(F_1,\ldots,F_n)^G = {}_{A\cup D}(B_1\cup F_1,\ldots,B_n\cup F_n)^{C\cup G},$$
$$A(B_1,\ldots,B_n)^C \cap {}_D(F_1,\ldots,F_n)^G = {}_{A\cap D}(B_1\cap F_1,\ldots,B_n\cap F_n)^{C\cap G},$$
$$\sim {}_A(B_1,\ldots,B_n)^C = {}_{\sim C}(\sim B_n,\ldots,\sim B_1)^{\sim A},$$

where $\sim$ is the set theoretic complement.

Proposition 3. *For an approximation space* (U,E), *both the lower ontology*

$$(\mathscr{O}_{\text{lower}}, \cup, \cap, \sim, {}_\emptyset(\emptyset_1,\ldots,\emptyset_n)^\emptyset, {}_U(U_1,\ldots,U_n)^U)$$

and the upper ontology

$$(\mathscr{O}^{\text{upper}}, \cup, \cap, \sim, {}_\emptyset(\emptyset_1,\ldots,\emptyset_n)^\emptyset, {}_U(U_1,\ldots,U_n)^U)$$

are de Morgan algebras.

As the open problem for future works is a question how one can characterise the full systems where also dynamic operators are included.

References

1. Arkhangel'skii, A.V., Fedorchuk, V.V.: The basic concepts and constructions of general topology. In: Arkhangel'skii, A.V., Pontryagin, L.S. (eds.) General Topology I. Encyclopedia of Mathematical Sciences, vol. 17. Springer, Heidelberg (1990)
2. Gärdefors, P.: Knowledge in Flux: Modeling the Dynamics of Epistemic States. Massetschussets Institute of Technology Press (1990)
3. Hansson, S.O.: A Textbook of Belief Dynamics. Theory of Change and Database Updating. Kluwer Academic Publishers, Dordrecht (1999)
4. Järvinen, J.: Lattice theory for 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. 400–498. Springer, Heidelberg (2007)
5. Marek, W., Truszczyński, M.: Contributions to the theory of rough sets. Fundamenta Informaticae 39(4), 389–409 (1999)
6. Pawlak, Z.: Rough sets. International Journal of Computer and Information Science 11, 341–356 (1982)
7. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)

Simplification of Neuro-Fuzzy Models

Krzysztof Simiński

Abstract. The neuro-fuzzy system presented in the paper is a system with parameterized consequences implementing hierarchical partition of the input domain. The regions are described with attributes values. In this system not all attribute values must be used to constitute the region. The attributes of minor importance may be ignored. The results of experiments show that the simplified model have less parameters and can achieve better generalisation ability.

Keywords: neuro-fuzzy system, hierarchical partition, simplification.

1 Introduction

Neuro-fuzzy systems are commonly used in data mining. The reason for the popularity of these systems is twofold: the fuzzy approach and the ability of knowledge generalisation. The advantage of the neuro-fuzzy systems is the intelligibility of model. The model is composed of rules building up the fuzzy rule base, the crucial part of fuzzy inference system. The parameters of the model are tuned to better fit the presented data and to elaborate more accurate models.

The rules in rule base are fuzzy implications. Their premises split the input domain space into regions. The parameters determining the regions are often function of all attributes values. This means that all attribute values are taken into consideration in each region as in Mamdani-Assilan [13] or Takagi-Sugeno-Kang [21, 23] fuzzy inference systems, in neuro-fuzzy systems ANFIS [7], LOLIMOT [14, 15], ANNBFIS [5, 9], ANLIR [10], HS-47 [18, 19]. In handling n-attribute objects the region is created in n-dimensional space. The idea arises to reduce the number of attributes used to constitute the region. Some attributes may be of minor importance

Krzysztof Simiński
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: Krzysztof.Siminski@polsl.pl

and can be ignored. This paper presents the system based on neuro-fuzzy system with hierarchical input domain partition [18] that elaborates the simplified models.

The paper is organized as follows. Section 2 presents neuro-fuzzy system with parameterized consequences. Section 3 describes the system producing simplified models. Section 4 presents the results of experiments and Sect. 5 discusses the results. Section 6 summaries the paper.

2 Neuro-Fuzzy System with Parameterized Consequences

The system with parameterized consequences [5, 11] is the MISO system. The rule base contains fuzzy rules as fuzzy implications

$$R^{(i)} : \underline{X} \text{ is } \underline{A}^{(i)} \Rightarrow Y \text{ is } B^{(i)}(\underline{\theta}), \tag{1}$$

where $\underline{X} = [x_1, x_2, \ldots, x_N]^{\mathrm{T}}$ and Y are linguistic variables, N – number of attributes. $\underline{A}$ and B are fuzzy linguistic terms and $\underline{\theta}$ is the consequence parameter vector. The linguistic variable A_i ($\underline{A}$ for ith attribute) is described with the Gaussian membership function:

$$\mu_{A_i}(x_i) = \exp\left(-\frac{(x_i - c_i)^2}{2\sigma_i^2}\right), \tag{2}$$

where c_i is the core location for ith attribute and σ_i is this attribute Gaussian bell deviation. The variable $\underline{A}$ represents the region in input domain.

The term B is represented by an isosceles triangle with the base width w, the altitude equal to the firing strength of the ith rule:

$$F^{(i)}(\underline{X}) = \mu_{\underline{A}^{(i)}}(\underline{X}) = \mu_{A_1^{(i)}}(x_1) \star_T \cdots \star_T \mu_{A_N^{(i)}}(x_N), \tag{3}$$

where $\star_T$ denotes the T-norm. The localisation of the core of the triangle fuzzy set is determined by linear combination of input attribute values:

$$y^{(i)} = \underline{\theta}_i^{\mathrm{T}} \cdot \left[1, \underline{X}^{\mathrm{T}}\right]^{\mathrm{T}} = \left[a_0^{(i)}, a_1^{(i)}, \ldots, a_N^{(i)}\right] \cdot [1, x_1, \ldots, x_N]^{\mathrm{T}}. \tag{4}$$

The fuzzy output of the system can be written as

$$\mu_{B'}(y, \underline{X}) = \bigoplus_{i=1}^{I} \Psi\left(\mu_{\underline{A}^{(i)}}(\underline{X}), \mu_{B^{(i)}}(y, \underline{X})\right), \tag{5}$$

where $\bigoplus$ denotes the aggregation, Ψ – the fuzzy implication and I – the number of rules.

The crisp output of the system is calculated using the MICOG method:

$$y = \frac{\sum_{i=1}^{I} g^{(i)}(\underline{X}) y^{(i)}(\underline{X})}{\sum_{i=1}^{I} g^{(i)}(\underline{X})}, \tag{6}$$

where $y^{(i)}(\underline{X})$ stands for the location of the core of the consequent fuzzy set, $F^{(i)}$ – the firing strength of the ith rule, $w^{(i)}$ – the width of the base of the isosceles triangle consequence function of the ith rule. The function g depends on the fuzzy implication, in the system the Reichenbach one is used, so for the ith rule function g is

$$g^{(i)}(\underline{X}) = \frac{w^{(i)}}{2} F^{(i)}(\underline{X}). \tag{7}$$

In neuro-fuzzy systems the parameters of the model are tuned to better fit the data. In this system the parameters of the premises (c and σ in 2) and the values of the supports w of the sets in consequences are tuned with gradient method. The linear coefficients for the calculation of the localisation of the consequence sets are optimised with LSME iterative algorithm [8].

The modern neuro-fuzzy systems possess the feature enabling automatic rule base extraction from presented data. The neuro-fuzzy systems based on system with parametrized consequences implement various methods to rules extraction: the ANNBFIS/ANLIR system [5] gets rules' premises by clustering the data in the input domain, the system proposed in [4] applies genetic algorithm, [3] proposes neuro-fuzzy modelling based on a deterministic annealing approach, the hierarchical input domain partition is proposed in [18].

The system proposed in [18] is a neuro-fuzzy system with parameterised consequences creating the fuzzy rule base by hierarchical partition of the input domain. This approach joins the advantages of grid split and clustering: no curse of dimensionality, the reduction of low membership areas and easier way to establish the number of regions. This approach is applied in systems depicted in [1, 14, 15, 18, 20].

Various methods of rules extraction are implemented in neuro-fuzzy systems. Two main classes of rules extraction methods are used. These are:

1. FPTC: elaboration of rules' premises then consequences,
2. FCTP: creation of consequences then premises for the extracted consequences [22].

In majority of systems of this kind the region that fulfils certain conditions is split into two equal subregions. In [18] the system HS-47 is presented that is able to split regions into unequal fuzzy subregions. The first step is the creation of the most general region that comprises all objects. Then iterative part of the procedure takes place: tuning the parameters of the model, error calculation (the error elaborated for each training object) and splitting the worst region (with the highest contribution to the global error of the system). The splitting procedure is based on the FCM clustering [6], what gives the fuzzy partition.

3 Elaborating Simplified Models

This paper presents the modification of the HS-47 – the system HS-87. In this systems not all attributes are used to constitute the region in the input domain. The

general scheme of algorithm is similar to HS-47. The first, most general, region contains all objects with membership equal one. Having found the worst region (the one with the highest contribution to the global error of the system). For the splitting of the found region the conditional FCM algorithm [16] is used. The condition is the membership of objects to the worst regions. Having split the worst region the most distinctive attribute is selected. The two subregions differ only in values of this attribute.

In the modification of the system with parameterized consequences not all attributes are necessarily used in determination of firing strength of the rule. So (3) becomes

$$F^{(i)}(\underline{X}) = \mu_{\underline{A}^{(i)}}(\underline{X}) = \mu_{A_1^{(i)}}(x_1) \star_T \cdots \star_T \mu_{A_{n \leqslant N}^{(i)}}(x_{n \leqslant N}), \qquad (8)$$

Figure 1 shows the modified fuzzy inference system with parameterized consequences for three-attribute objects and two fuzzy rules. For each rule the Gaussian membership values for attributes used in the rule are determined. In the first rule the first and third attribute are used, in the second rule – the second and the third one. These values are then used to calculate the firing strength of the rule for this object. The membership values for attributes are T-normed to determine the rule's firing strength. This value becomes then the height of the isosceles triangle in the rule's consequence. The values of the all object's attributes are used to calculate the localisation of the triangle fuzzy set core in the rule's consequence. In Fig. 1 these are $y^{(1)}$ for the first rule and $y^{(2)}$ for the second one. The values of the triangle set supports, $w^{(1)}$ and $w^{(2)}$, are tuned in system tuning procedure and are not further modified during calculation of the system's response. The fuzzy sets of all consequences are aggregated and the crisp output is determined with MICOG procedure (6).
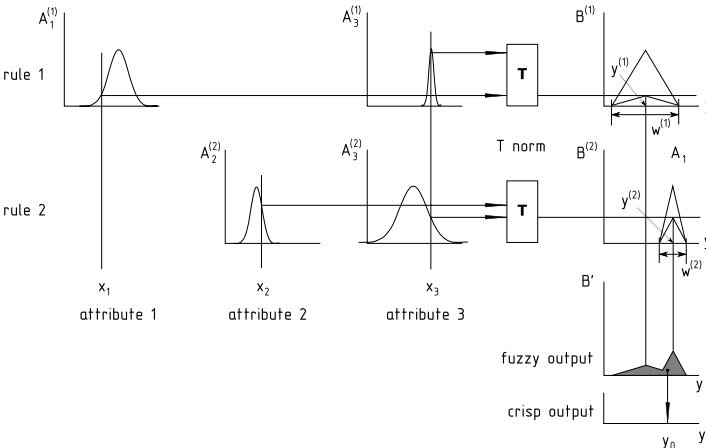
Fig. 1 The scheme of the fuzzy inference system with parametrized consequences with two rules and three-attribute objects

4   Experiments

The experiments were performed using the synthetic ('Hang', 'leukocyte concentration') and real life ('CO_2 concentration', 'wheat prices', 'trip', 'wages', 'milk') data sets. For brevity of the paper we do not provide the precise description of the datasets. The descriptions of the datasets can be found in papers: 'Hang' [17], 'leukocyte concentration' [10], 'CO_2 concentration' and 'wheat prices' [19], 'trip' [2], 'wages' [12]. The 'milk' is a real life dataset [12] describing the monthly milk production per cow over 14 years. The normalised (to the mean equal zero and standard deviation one) time series data were organised in the following manner: $[t-5, t-3, t-2, t-1, t, t+5]$. Of 159 tuples the first 90 were used as a train set and following 69 as test set. Function approximation for 'Hang' dataset was conducted as a 10-fold cross validation. The 'CO_2 concentration' dataset was divided into train and test sets. The rest of dataset represent data series and were divided into train and test sets.
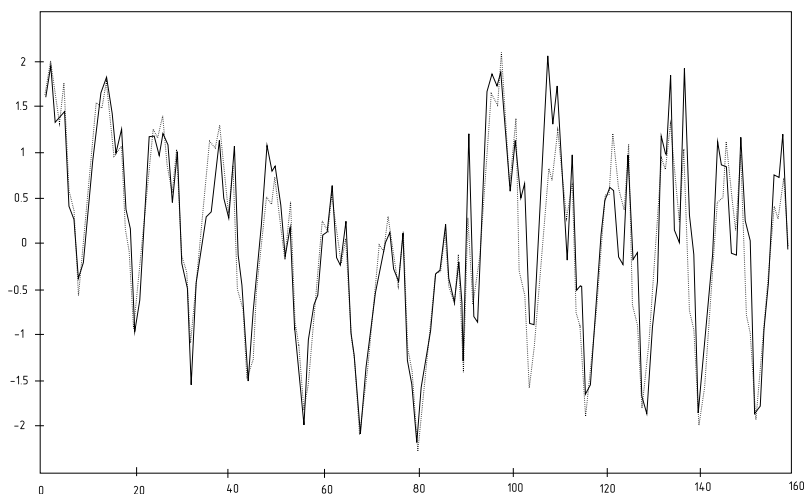
Fig. 2 The results elaborated by HS-87 system for 'milk' dataset. Abscissa represents the objects (train set: tuples 1–90, test set: tuples 91–159), ordinata – the value of the predicted attribute. The dotted line stands for expected values, the solid line – values predicted by the HS-87 system

5   Discussion

The results of the experiments executed for the above mentioned datasets are presented in Tables 1 and 2.

Table 1 presents the RMSE values for data series for ANNBFIS, HS-47 and HS-87 systems. The table also represents the number of parameters of the elaborated models. The values of RMSE are not always smaller than in HS-87 system, but

Table 1 Results elaborated in 10-fold cross validation for 'Hang' dataset. Abbreviations: 'R' – number of rules, 'iter' – number of iterations in tuning procedure, 'RMSE' – root mean square error, 'pars' – number of model's parameters

dataset	attr	ANNBFIS				HS-47				HS-87			
		R	iter	RMSE	pars	R	iter	RMSE	pars	R	iter	RMSE	pars
milk	5	8	250	0.666282	136	6	250	0.613978	102	4	250	0.522519	46
mg	4	25	100	0.001052	350	25	100	0.002586	350	25	100	0.000946	312
wheat prices	3	4	250	0.709315	44	5	250	0.716344	55	1	250	0.598857	4
wages	5	8	250	0.465313	136	4	250	1.092827	68	1	250	0.195222	7
trip	5	2	250	0.438633	34	2	250	0.462913	34	2	250	0.425966	18
CH$_4$ concen.	7	3	250	0.103795	69	2	250	0.138982	46	1	250	0.107158	9

are at least comparable. The number of parameters shows that the model created by HS-87 is simpler. The simplification of the model can lead to improving the generalisation ability of the systems.

Table 2 presents the root mean square error (RMSE) for 10-fold cross-validation of the 'Hang' dataset. The table presents the best results for the number of rules not exceeding 10. The tuning procedure was repeated for 100 iterations. The last row of the table shows the average values of the RMSE.

Figure 3 presents the models elaborated by the ANNBFIS and HS-87 systems for 'wheat prices' dataset. ANNBFIS extracted 4 rules with 44 parameters and HS-87 one rule with 4 parameters.

Table 2 The results of 10-fold cross-validation for 'Hang' dataset elaborated by ANNBFIS, HS-47 and HS-87 systems. In all system the tuning procedure with 100 iterations was applied. 'RMSE' stands for 'root mean square error'. The last row contains the average values of RMSE

ANNBFIS		HS-47		HS-87	
RMSE	rules	RMSE	rules	RMSE	rules
0.084773	9	0.017480	10	0.010745	10
0.083279	10	0.037228	10	0.023489	10
0.093957	7	0.048759	10	0.023366	10
0.090071	10	0.122333	10	0.015845	10
0.087045	7	0.041752	8	0.025104	10
0.076674	9	0.023956	8	0.020117	9
0.063281	9	0.018713	10	0.022681	9
0.061915	10	0.028574	10	0.011313	10
0.053528	9	0.027513	10	0.017966	9
0.131234	9	0.105462	10	0.090172	9
0.082576		0.047177		0.026080	

HS-87

rule 1 $\begin{cases} \text{for all objects} \\ y = 0.424935a_1 + 1.335946a_2 + 0.673199a_3 + 4.210080 \end{cases}$

ANNBFIS

rule 1 $\begin{cases} c = [0.608978, -0.111311, 0.224342] \\ s = [1.200061, 0.936951, 0.859584] \\ w = 2.015290 \\ y = 5.752873a_1 + 1.295029a_2 + 4.215323a_3 - 22.182970 \end{cases}$

rule 2 $\begin{cases} c = [0.007774, -0.260185, -0.538492] \\ s = [1.118571, 0.923723, 0.935704] \\ w = 2.005516 \\ y = 5.609943a_1 + 1.109718a_2 + 12.352983a_3 + 18.372435 \end{cases}$

rule 3 $\begin{cases} c = [-0.141376, 0.665548, 0.566459] \\ s = [1.029580, 0.992029, 0.737582] \\ w = 2.037077 \\ y = 0.049244a_1 - 2.603480a_2 + 0.114546a_3 + 4.676985 \end{cases}$

rule 4 $\begin{cases} c = [-0.962830, -1.013758, -0.954857] \\ s = [1.013760, 1.331727, 1.337511] \\ w = 1.940847 \\ y = 0.424935a_1 + 1.335946a_2 + 0.673199a_3 + 4.210080 \end{cases}$

Fig. 3 The models elaborated by the ANNBFIS and HS-87 systems for 'wheat prices' dataset. The symbol a_i stands for the ith attribute

6 Conclusions

The paper presents the simplification of the neuro-fuzzy system with parameterized consequences. The premises of the fuzzy rules split the input domain into regions. The regions are described in the n-dimensional space (where n is a number of attributes of the objects). This paper presents the system that does not necessarily uses all attribute values. Some dimension may have no greater significance for building the model and can be ignored.

The experiments on real-life and synthetic data shows that this approach can lead not only to simplification of models but also to improving the generalisation ability of the elaborated models.

The proposed system can be also used for finding attributes of minor importance. After elaboration of model some attributes may be not used in any rule, what shows that they are abundant.

References

1. Almeida, M.R.A.: Sistema híbrido neuro-fuzzy-genético para mineração automática de dados. Master's thesis, Pontifíca Universidade Católica do Rio de Janeiro (2004)
2. Chiu, S.L.: Fuzzy model identification based on cluster estimation. Journal of Intelligent and Fuzzy Systems 2, 267–278 (1994)

3. Czabański, R.: Neuro-fuzzy modelling based on a deterministic annealing approach. International Journal of Applied Mathematics and Computer Science 15(4), 561–576 (2005)

4. Czekalski, P.: Evolution-fuzzy rule based system with parameterized consequences. International Journal of Applied Mathematics and Computer Science 16(3), 373–385 (2006)

5. Czogała, E., Łęski, J.: Fuzzy and neuro-fuzzy intelligent systems. Studies in Fuzziness and Soft Computing (2000)

6. Dunn, J.C.: A fuzzy relative of the ISODATA process and its use in detecting compact, well separated clusters. Journal Cybernetics 3(3), 32–57 (1973)

7. Jang, J.S.R.: ANFIS: Adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man, and Cybernetics 23, 665–684 (1993)

8. Larminat, P., Thomas, Y.: Automatyka – układy liniowe. Wydawnictwa Naukowo-Techniczne (1983)

9. Łęski, J., Czogała, E.: A new artificial neural network based fuzzy inference system with moving consequents in if-then rules and selected applications. BUSEFAL 71, 72–81 (1997)

10. Łęski, J.: Systemy neuronowo-rozmyte. Wydawnictwa Naukowo-Techniczne. Warsaw, Poland (2008)

11. Łęski, J., Czogała, E.: A new artificial neural network based fuzzy inference system with moving consequents in if-then rules and selected applications. Fuzzy Sets and Systems 108(3), 289–297 (1999)

12. Makridakis, S.G., Wheelwright, S.C., Hyndman, R.J.: Forecasting: Methods and Applications, 3rd edn. John Wiley & Sons, Chichester (1998)

13. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies 7(1), 1–13 (1975)

14. Nelles, O., Fink, A., Babuška, R., Setnes, M.: Comparison of two construction algorithms for Takagi-Sugeno fuzzy models. International Journal of Applied Mathematics and Computer Science 10(4), 835–855 (2000)

15. Nelles, O., Isermann, R.: Basis function networks for interpolation of local linear models. In: Proceedings of the 35th IEEE Conference on Decision and Control, vol. 1, pp. 470–475 (1996)

16. Pedrycz, W.: Conditional fuzzy clustering in the design of radial basis function neural networks. IEEE Transactions on Neural Networks 9(4), 601–612 (1998)

17. Rutkowski, L., Cpałka, K.: Flexible neuro-fuzzy systems. IEEE Transactions on Neural Networks 14(3), 554–574 (2003)

18. Simiński, K.: Neuro-fuzzy system with hierarchical partition of input domain. Studia Informatica 29(4A (80)), 43–53 (2008)

19. Simiński, K.: Two ways of domain partition in fuzzy inference system with parametrized consequences: Clustering and hierarchical split. In: Proceedings of the 10th International PhD Workshop, pp. 103–108 (2008)

20. de Souza, F.J., Vellasco, M.M.R., Pacheco, M.A.C.: Hierarchical neuro-fuzzy quadtree models. Fuzzy Sets and Systems 130(2), 189–205 (2002)

21. Sugeno, M., Kang, G.T.: Structure identification of fuzzy model. Fuzzy Sets and Systems 28(1), 15–33 (1988)

22. Sugeno, M., Yasukawa, T.: A fuzzy-logic-based approach to qualitative modeling. IEEE Transactions on Fuzzy Systems 1(1), 7–31 (1993)

23. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modeling and control. IEEE Transactions on Systems, Man and Cybernetics 15(1), 116–132 (1985)

Fuzzy Weighted Averaging
Using Criterion Function Minimization

Alina Momot and Michał Momot

Abstract. In this paper there is presented the computational study of fuzzy weighted averaging of data in the presence of non-stationary noise. There is proposed a new method, which is an extension of Weighted Averaging using Criterion Function Minimization (WACFM). In the method the weighting coefficients are fuzzy numbers instead of classical real numbers. The determining of these coefficients requires the extending of WACFM method for certain types of fuzzy numbers. In the presented case there is made an assumption of the triangle membership function for fuzzy coefficients. The performance of presented method is experimentally evaluated and compared with the traditional arithmetic averaging as well as Weighted Averaging using Criterion Function Minimization (WACFM) for the ECG signal.

Keywords: weighted averaging, fuzzy coefficients, criterion function minimization, biomedical signals.

1 Introduction

Traditional or weighted averaging, being an example of information fusion, may be used when several sources of data (for example sensors) are employed in a system in order to reduce uncertainty and resolve the ambiguity often present in the information from a single source. Consequently, the addition of extra sources, providing

Alina Momot
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: `alina.momot@polsl.pl`

Michał Momot
Institute of Medical Technology and Equipment,
Roosevelt 118, 41-800 Zabrze, Poland
e-mail: `michal.momot@itam.zabrze.pl`

redundant and complementary information, offers the capability of resolving most complex situations and leads to a richer description of the world. Several approaches for fuzzy averaging have been proposed in the literature: fuzzy arithmetic averaging in [6] as well as fuzzy weighted averaging in [7], and in [3] there is presented comparison of several discrete algorithms for fuzzy weighted average.

Similarly the procedure of averaging can be applied to the set of data containing repetitive patterns which situation arises in biomedical signals such as ECG or EEG signals. Usually recording the electrodiagnostic signals is performed in the presence of a noise and one of the commonly used techniques to extract a useful signal distorted by a noise is weighted averaging, since the nature of the signals is quasi-cyclic with the level of noise power varying from cycle to cycle.

This paper presents a new method, which is an extension of Weighted Averaging using Criterion Function Minimization (WACFM) [9]. In the method the weighting coefficients are fuzzy numbers instead of classical real numbers. The determining of these coefficients requires the extending of WACFM method for certain types of fuzzy numbers. In the presented case there is made an assumption of the triangle membership function for fuzzy coefficients. The performance of presented method will be experimentally evaluated and compared with the traditional arithmetic averaging as well as Weighted Averaging using Criterion Function Minimization (WACFM) for the ECG signal from CTS database [4].

2 Signal Averaging Methods

Let us assume that in each signal cycle $y_i(j)$ is the sum of a deterministic (useful) signal $x(j)$, which is the same in all cycles, and a random noise $n_i(j)$ with zero mean and variance for the ith cycle equal to σ_i^2 . Thus,

$$y_i(j) = x(j) + n_i(j), \tag{1}$$

where i is the cycle index $i \in \{1, 2, \ldots, M\}$, and the j is the sample index in the single cycle $j \in \{1, 2, \ldots, N\}$ (all cycles have the same length N). The weighted average is given by

$$v(j) = \sum_{i=1}^{M} w_i y_i(j), \tag{2}$$

where w_i is a weight for ith signal cycle ($i \in \{1, 2, \ldots, M\}$) and $\mathbf{v} = [v(1), v(2), \ldots, v(N)]$ is the averaged signal.

The traditional ensemble averaging with arithmetic mean as the aggregation operation gives all the weights w_i equal to M^{-1} . If the noise variance is constant for all cycles, then these weights are optimal in the sense of minimizing the mean square error between $\mathbf{v}$ and $\mathbf{x}$, assuming Gaussian distribution of noise. When the noise has a non-Gaussian distribution, the estimate (2) is not optimal, but it is still the best of all linear estimators of $\mathbf{x}$ [8].

2.1 Weighted Averaging Method WACFM

In [9] it is presented algorithm WACFM (Weighted Averaging method based on Criterion Function Minimization). The idea of the algorithm is based on the fact that for $\mathbf{y}_i = [y_i(1), y_i(2), \ldots, y_i(N)]^T$, $\mathbf{w} = [w_1, w_2, \ldots, w_M]^T$ and $\mathbf{v} = [v(1), v(2), \ldots, v(N)]$ minimization of the following scalar criterion function

$$I_m(\mathbf{w}, \mathbf{v}) = \sum_{i=1}^{M} (w_i)^m \rho (\mathbf{y}_i - \mathbf{v}), \tag{3}$$

where $\rho(\cdot)$ is a measure of dissimilarity for vector argument and $m \in (1, \infty)$ is a weighting exponent parameter, with respect to the weights vector yields

$$w_i = \left(\rho (\mathbf{y}_i - \mathbf{v})^{\frac{1}{1-m}} \right) \Big/ \left(\sum_{k=1}^{M} \rho (\mathbf{y}_k - \mathbf{v})^{\frac{1}{1-m}} \right), \tag{4}$$

for $i \in \{1, 2, \ldots, M\}$. When the quadratic function $\rho(\cdot) = \| \cdot \|_2^2$ is used, the averaged signal can be obtained as

$$\mathbf{v} = \left(\sum_{i=1}^{M} (w_i)^m \mathbf{y}_i \right) \Big/ \left(\sum_{i=1}^{M} (w_i)^m \right), \tag{5}$$

for the weights vector given by (4) with the quadratic function. The optimal solution for minimization (3) with respect to $\mathbf{w}$ and $\mathbf{v}$ is a fixed point of (4) and (5) and it is obtained from the Picard iteration.

If parameter m tends to one, then the trivial solution is obtained where only one weight, corresponding to the signal cycle with the smallest dissimilarity to averaged signal, is equal to one. If m tends to infinity, then weights tend to M^{-1} for all i. Generally, a larger m results in a smaller influence of dissimilarity measures. The most common value of m is 2 which results in greater decrease of medium weights [9].

2.2 Fuzzy Weighted Averaging Method FWACFM

In the method the weighting coefficients (4) are triangular fuzzy numbers instead of classical real numbers. Consequently, the weighted average (2) is vector containing triangular fuzzy numbers and there is necessary to compute distance between the input signal (vector of real numbers) and the averaged signal. As the desired distance it is taken the distance between the real number and the α-cut of the corresponding fuzzy number which will be describe explicitly below.

Fuzzy number on the real line $\mathscr{R}$ may be represented by the L–R representation [5], e.g., $A = (l_A, m_A, u_A)_{L-R}$ where l_A and u_A denoted the lower and upper bounds, m_A the mode, and L and R the left and right membership (reference or shape) functions of A, respectively. The membership function of A which defines the degree of belongingness of elements $x \in \mathscr{R}$ to A, is denoted as $\mu_A(x) \in [0, 1]$ and

is defined by the $L(x)$ and $R(x)$. The α-cut, $\alpha \in [0,1]$, of a fuzzy number is defined as the ordinary subset $\{x \in \mathcal{R} : \mu_A(x) \geq \alpha\}$ and written as $(A)_\alpha = [a,b]$, where a and b denote the left and right endpoints of $(A)_\alpha$, respectively. Thus, the triangular fuzzy number may be specially denoted as $(l_A, m_A, u_A)_T$ with

$$\mu_A(x) = \begin{cases} L(x) = (x-l_A)/(m_A-l_A) & \text{for } x \in [l_A, m_A] \\ R(x) = (u_A-x)/(u_A-m_A) & \text{for } x \in [m_A, u_A] \\ 0 & \text{for } x \in \mathcal{R} \setminus [l_A, u_A] \end{cases} \tag{6}$$

and $(A)_\alpha = [a,b] = [(m_A-l_A)\alpha + l_A, u_A - (u_A-m_A)\alpha]$.

In this approach there will be used symmetrical triangular fuzzy number, where $l_A = m_A - r_A$ and $u_A = m_A + r_A$. Then

$$\mu_A(x) = \max\left\{1 - |x-m_A| r_A^{-1}, 0\right\} \tag{7}$$

and

$$(A)_\alpha = [m_A - r_A(1-\alpha), m_A + r_A(1-\alpha)]. \tag{8}$$

It is worth noting that in this case each triangular fuzzy number can be uniquely described by two parameters, namely m_A which is its center and r_A which is its radius. Thus, the distance between the real number x and the α-cut of the symmetrical triangular fuzzy number A is given by $\rho_\alpha(x,A) = \rho(x,(A)_\alpha)$, which leads to

$$\rho_\alpha(x,A) = \inf\{|x-t| : t \in [m_A - r_A(1-\alpha), m_A + r_A(1-\alpha)]\} \tag{9}$$

and the explicit formula of which is given by

$$\rho_\alpha(x,A) = \max\left\{|x-m_A| - r_A, 0\right\}. \tag{10}$$

When the arguments of function $\rho_\alpha(\cdot,\cdot)$ are K-dimensional vectors, the formula is

$$\rho_\alpha(\mathbf{x},\mathbf{A}) = \sum_{k=1}^{K} (\rho_\alpha(x_k, A_k))^2. \tag{11}$$

Adopting the notations from above, the proposed new weighted averaging algorithm FWACFM (Fuzzy WACFM) can be described as follows, where ε is a preset parameter:

1. Determine parameters r which is radius of symmetrical triangular fuzzy numbers and α which is the level of its cutting. These parameters remain constant during iterations. Initialize fuzzy weights $\mathbf{w}^{(0)}$. Set the iteration index $k = 1$.
2. Calculate vector of centers of fuzzy weights $\mathbf{w}^{(k)}$ using

$$w_i^{(k)} = \left(\rho_\alpha(\mathbf{y}_i, \mathbf{v})^{\frac{1}{1-m}}\right) \Big/ \left(\sum_{k=1}^{M} \rho_\alpha(\mathbf{y}_k, \mathbf{v})^{\frac{1}{1-m}}\right), \tag{12}$$

for $i \in \{1, 2, \ldots, M\}$.

3. Calculate the averaged signal as

$$\mathbf{v}^{(k)} = \left(\sum_{i=1}^{M} (W_i^{(k)})^m \mathbf{y}_i \right) \Big/ \left(\sum_{i=1}^{M} (W_i^{(k)})^m \right), \tag{13}$$

where $W_i^{(k)}$ is a symmetrical triangular fuzzy number with center given by (12) and radius r.

4. If $\|\mathbf{w}^{(k-1)} - \mathbf{w}^{(k)}\| > \varepsilon$ then $k \leftarrow k+1$ and go to 2.
5. Calculate final averaged signal as

$$\mathbf{v}^{(k)} = \left(\sum_{i=1}^{M} (w_i^{(k)})^m \mathbf{y}_i \right) \Big/ \left(\sum_{i=1}^{M} (w_i^{(k)})^m \right). \tag{14}$$

It is worth noting that this is generalization of WACFM method because for radius equal 0 both methods are equivalent.

3   Numerical Experiments

The procedure of averaging often is used to attenuate the noise in sets of data containing repetitive patterns such as ECG signals, and even in some medical applications the averaging is the only method taken into account [1]. The CTS database [4], which contains files with ECG signals, was proposed by the International Electrotechnical Commission (IEC) within the European project *Common Standards for Quantitative Electrocardiography* in order to test an accuracy of signal processing methods. Below, the numerical experiments present application of FWACFM method to attenuate the noise in ECG signals. As the deterministic component was taken ANE20000 signal from CTS database, where the signal amplitude is expressed in Volts, and to this signal independent realizations of random noise were added.

Performance of the new method FWACFM was experimentally compared with the traditional averaging by using arithmetic mean and weighted averaging method WACFM. In all experiments, using weighted averaging, calculations were initialized as the means of disturbed signal cycles, parameter m was equal to 2 and the parameter ε was equal to 10^{-12}. In FWACFM parameter α was equal to 0.5. For the computed averaged signal the performance of tested methods was evaluated by the maximal absolute difference between the deterministic component and the averaged signal (MAX). The root mean-square error between the deterministic component and the averaged signal (RMSE) was also computed. All experiments were run on the PC platform using implementation in the C++ language.

First, as the input data a series of 100 ECG cycles was generated with the same deterministic component (ANE20000 signal) and zero-mean white Gaussian noise with different standard deviations with constant amplitude of noise during each cycle. For the first, second, third and fourth 25 cycles, the noise standard deviations
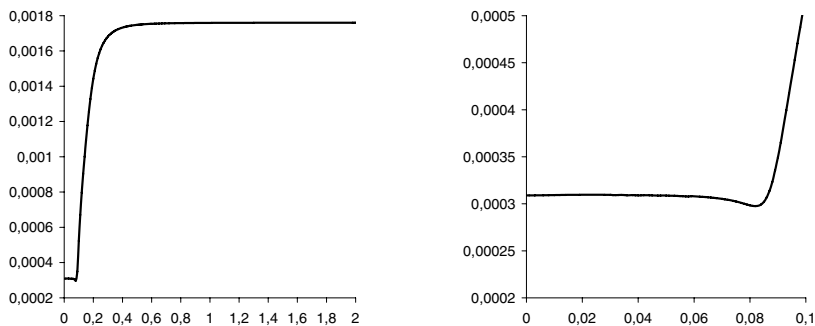
Fig. 1 RMSE in case of Gaussian noise (the enlarged selected interval on the right)
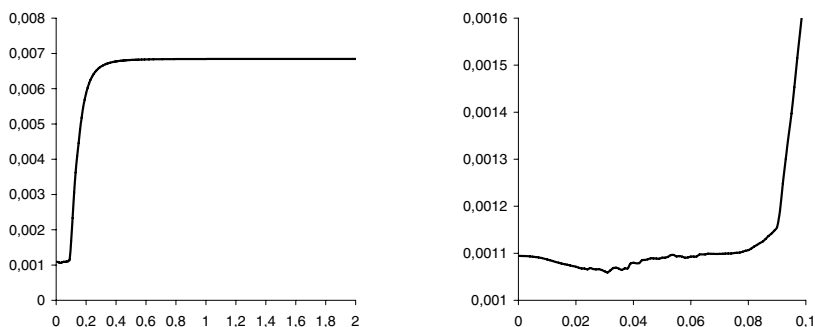
Fig. 2 MAX in case of Gaussian noise (the enlarged selected interval on the right)

were respectively $0.01s$, $0.05s$, $0.1s$, $0.2s$, where s is the sample standard deviation of the deterministic component. For FWACFM there were performed experiments with radius varying from 0 to 4 with step 0.01, the best root mean-square error was obtained for radius equal 0.08 (see figure 1 on the left) and the best maximal absolute difference between the deterministic component and the averaged signal was obtained for radius equal 0.03 (see Fig. 2 on the left). The results for radius above 2 were almost constant, therefore Figs. 1 and 2 present only results for radius varying from 0 to 2.

Table 1 presents the RMSE and the absolute maximal value (MAX) of residual noise for the traditional Arithmetic Averaging (AA), WACFM method and the best results for FWACFM.

Additionally there were performed experiments with radius varying from 0 to 0.1 with step 0.001 (see Figure 1 and Figure 2 on the right), which was motivated by local minimum of both RMSE and MAX appearing there. The additional experiments showed that minimum of RMSE was equal to 0.000297654 for radius $r = 0.082$ and minimum of MAX was equal to 0.00105918 for radius $r = 0.031$.

The similar series of experiments was performed where the input data was a set of 100 ECG cycles generated with the same deterministic component (ANE20000

Table 1 Results for Gaussian noise

Method	RMSE	MAX
AA	0.00176037	0.00684356
WACFM	0.000308965	0.00109442
FWACFM	0.000298404	0.00106167
	$(r = 0.08)$	$(r = 0.03)$
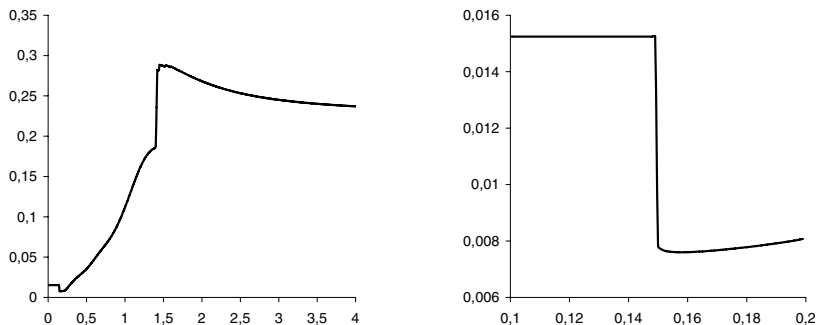
Fig. 3 RMSE in case of Cauchy noise (the enlarged selected interval on the right)

signal) with added Cauchy noise [2], which was treated as simulated impulse noise, with location parameter equal 0 and scale parameter constant during each cycle. For the first, second, third and fourth 25 cycles, the scale parameters were respectively $0.01s$, $0.05s$, $0.1s$, $0.2s$, where s is the sample standard deviation of the deterministic component. For FWACFM there were performed experiments with radius varying from 0 to 4 with step 0.01. For radius equal 0.16 there was obtained both the best RMSE (see Figure 3 on the left) and the best MAX (see figure 4 on the left).
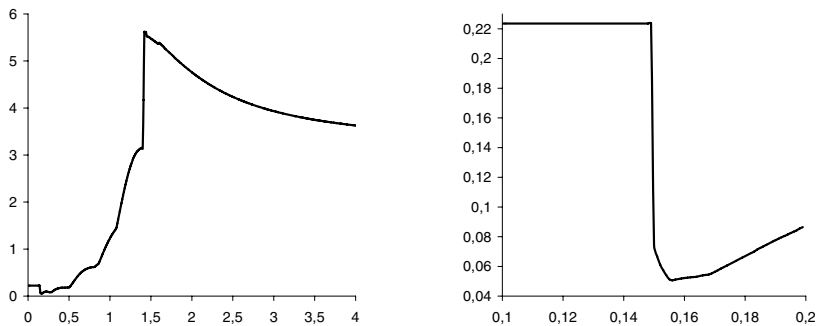
Fig. 4 MAX in case of Cauchy noise (the enlarged selected interval on the right)

Table 2 Results for Cauchy noise

Method	RMSE	MAX
AA	0.227349	3.2445
WACFM	0.015242	0.224
FWACFM	0.0076	0.052

Table 2 presents the RMSE and the absolute maximal value (MAX) of residual noise for the traditional Arithmetic Averaging (AA), WACFM method and the best results for FWACFM ($r = 0.16$).

There were also performed experiments with radius varying from 0.1 to 0.2 with step 0.001 (see Fig. 3 and Fig. 4 on the right), which showed that minimum of RMSE was equal to 0.00760095 for radius $r = 0.158$ and minimum of MAX was equal to 0.0506516 for radius $r = 0.156$.

All performed experiments show that the FWACFM method can be applied to attenuate the noise in ECG signal since such weighted averaging results in lower errors than arithmetic averaging which is traditionally used method. For some positive values of radius parameter such generalization of WACFM method (for radius equal 0 both methods are equivalent) outperforms the WACFM (even greatly in the case of Cauchy noise), which motivates further research on methods for automatic determination of this parameter.

References

1. Bailon, R., Olmos, S., Serrano, P., Garcia, J., Laguna, P.: Robust measure of ST/HR hysteresis in stress test ECG recordings. Computers in Cardiology 29, 329–332 (2002)
2. Brandt, S.: Statistical and Computational Methods in Data Analysis. Springer, Heidelberg (1997)
3. Chang, P.T., Hung, K.C., Lin, K.P., Chang, C.H.: A comparison of discrete algorithms for fuzzy weighted average. IEEE Transaction on Fuzzy Systems 14(5), 663–675 (2006)
4. Commission, I.E.: Standard 60601-3-2 (1999)
5. Dubois, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, New York (1980)
6. Gao, L.S.: The fuzzy arithmetic mean. Fuzzy Sets and Systems 107, 335–348 (1999)
7. Lee, D.H., Park, D.: An efficient algorithm for fuzzy weighted average. Fuzzy Sets and Systems 87, 39–45 (1997)
8. Łęski, J.: Application of time domain signal averaging and Kalman filtering for ECG noise reduction. Ph.D. thesis, Silesian University of Technology, Gliwice, Poland (1989)
9. Łęski, J.: Robust weighted averaging. IEEE Transactions on Biomedical Engineering 49(8), 796–804 (2002)

Approximate String Matching
by Fuzzy Automata

Václav Snášel, Aleš Keprt, Ajith Abraham, and Aboul Ella Hassanien

Abstract. We explain new ways of constructing search algorithms using fuzzy sets and fuzzy automata. This technique can be used to search or match strings in special cases when some pairs of symbols are more similar to each other than the others. This kind of similarity cannot be handled by usual searching algorithms. We present sample situations, which would use this kind of searching. Then we define a fuzzy automaton, and some basic constructions we need for our purposes. We continue with definition of our fuzzy automaton based approximate string matching algorithm, and add some notes to fuzzy–trellis construction which can be used for approximate searching.

Václav Snášel
Department of Computer Science, Faculty of Electrical Engineering and Computer Science,
VŠB – Technical University of Ostrava
17. listopadu 15, 708 33 Ostrava–Poruba, Czech Republic
e-mail: `Vaclav.Snasel@vsb.cz`

Aleš Keprt
Department of Computer Science, Faculty of Sciences, Palacký University,
Tomkova 40, 779 00 Olomouc, Czech Republic
e-mail: `Ales.Keprt@upol.cz`

Ajith Abraham
Center of Excellence for Quantifiable Quality of Service, Norwegian University of Science and Technology,
Norway
e-mail: `ajith.abraham@ieee.org`

Aboul Ella Hassanien
Faculty of Computer and Information, Cairo University,
Giza, Egypt
e-mail: `aboitcairo@gmail.com`

1 Introduction

When constructing search algorithms, we often need to solve the problem of approximate searching. These constructions can also be extended by a weight function, as described by Muthukrishnan [7].

Approximate string matching and searching is not a new problem, it has been faced and solved many times. It is usually based on Aho-Corasick automata and trellis constructions, and is often used when working with text documents or databases, or antivirus software.

In this paper, we present a technique which can be used to search or match strings in special cases when some pairs of symbols are more similar to each other than other pairs. This kind of similarity cannot be handled by usual searching algorithms. (Also note that even so called 'fuzzy string matching' does not distinguish between more or less similar symbols, so it's not related to fuzzy math at all.) We start our paper with some motivational examples, showing that there are several situations, which would use this kind of searching. Then we define a fuzzy automaton, and some basic constructions we need for our purposes. We continue with definition of our fuzzy automata based approximate matching algorithm, and add some notes to fuzzy-trellis construction which can be used for approximate searching.

2 Motivational Examples

Let us start with some motivational examples.

2.1 DNA Strings

We can understand DNA as a string in alphabet $\Sigma = \{A, C, G, T\}$. Bases A and G are called purine, and bases C and T are called pyrimidine.

Kurtz [4] writes: 'The transversion/transition weight function reflect the biological fact that a purine $\rightarrow$ purine and pyrimidine $\rightarrow$ pyrimidine replacement is much more likely to occur than a purine $\rightleftarrows$ pyrimidine replacement. Moreover, it takes into account that a deletion or insertion of a base occurs more seldom.'

In the other words, we have to take into account that the level of similarity or difference of two particular DNA strings cannot be simply expressed as the number of different symbols in them. We need to look at the particular symbol pairs. Obviously, the classic algorithm of approximate string searching does not cover this situation.

2.2 Spell Checker

A spell checker based on a dictionary of correct words and abbreviations is a common way of doing basic check of a text document. We go through document and search each of its words in our dictionary. The words not found in there are highlighted and a correction is suggested. The suggested words are those ones which

are present in the dictionary and are the most similar to the unknown one is sense of addition, deletion and replacement of symbols.

This common model is simple to implement, but it does not cover the fact that some pairs of symbols are more similar than others. This is also very language-specific. For example in Latin alphabet 'a' and 'e' or 'i' and 'y' are somewhat related hence more similar than for example 'w' and 'b'. In many European languages we can found some letters of extended Latin alphabet, whose similarity solely depends on the nature of a national language, e.g. in some languages 'ä' is similar or even identical to 'ae' so their exchange should be favored over other string operations. The primary problem here is that it cannot be simply implemented by standard string search models.

2.3 Summary

A fuzzy automaton allows us to define individual levels of similarity for particular pairs of symbols or sequences of symbols, so it can be used as a base for a better string search in the sense of presented examples.

There are extensive research materials discussing fuzzy automata – you can find a list of other works in Asveld's survey [1].

3 Approximate String Matching by Fuzzy Automata

3.1 Fuzzy Set

For completeness, we start with a very short definition of a fuzzy set. We define set L as the interval $L = [0,1]$, $\bigvee L = \sup L = 1$, $\bigwedge L = \inf L = 0$. Let B is a finite set. Then function $A : B \rightarrow L$ is called *fuzzy set* A of B.

Whenever $A \subseteq B$, we can also take A as a fuzzy set $A : B \rightarrow L$.

$$\forall b \in B : A(b) = \begin{cases} \bigvee L \text{ if } b \in A, \\ \bigwedge L \text{ if } b \notin A. \end{cases}$$

Note: Definition of L and related stuff can also be more generalized, see Nguyen & Walker [8] or Bělohlávek [2] for more details. Also, an infinite B could possibly be a base of an infinite fuzzy set, but we do not need this kind of generalization here.

3.2 Fuzzy Automaton

Fuzzy automata are generalization of nondeterministic finite automata (see Gruska [3]) in that they can be in each of its states with a degree in range L.

Fuzzy automaton is a system

$$M = (\Sigma, Q, \delta, S, F),$$

where:

- Σ is a finite input alphabet,
- Q is a finite set of states,
- $S \subseteq Q$ is the set of start states,
- $F \subseteq Q$ is the set of final (accepting) states,
- $\delta = \{\delta_a : a \in \Sigma\}$ is a fuzzy transition function,
- $\delta_a : Q \times Q \to L$ is a fuzzy transition matrix of order $|Q|$, i.e., a fuzzy relation.

Note: Fuzzy Automaton recognizes (accepts) a fuzzy language, i.e., language to which words belong in membership/truth degrees not necessarily equal to 0 or 1. Instead, each state of fuzzy automaton is a vector of values in range $[0,1]$ (i.e., each state maps $Q \to L$).

3.3 The Transition Function

Fuzzy transition function δ is actually the set of fuzzy relation matrices mentioned above, i.e., a fuzzy set of $Q \times \Sigma \times Q$:

$$\delta : Q \times \Sigma \times Q \to L.$$

For a given $s, t \in Q$ and $a \in \Sigma$, value of $\delta(s, a, t) = \delta_a(s, t)$ is the degree of transition from state s to state t for input symbol a.

Every fuzzy set A of Q is called a *fuzzy state* of automaton M. If an input $a \in \Sigma$ is accepted by M, the present fuzzy state A will be changed to the state $B = A \circ \delta_a$, where $\circ$ is a composition rule of fuzzy relations (e.g., minimax product).

Note: This definition is very similar to the one of the probabilistic finite automaton, including the set of transition matrices (see [3]). We can see that the notation is similar, but we must be aware that the principles of fuzzy automata are quite different and more generic compared to the quite old-fashioned probabilistic automata.

3.4 Minimax Product

Minimax product is defined as follows. Let $P = [p_{ij}]$, $Q = [q_{jk}]$ and $R = [r_{ik}]$ be matrix representations of fuzzy relations for which $P \circ Q = R$. Then, by using matrix notation, we can write $[p_{ij}] \circ [q_{jk}] = [r_{ik}]$, where

$$r_{ik} = \bigvee_{\forall j}(p_{ij} \wedge q_{jk}).$$

Note: This is equivalent to the classic matrix multiplication with operators $\vee$ (join) and $\wedge$ (meet) used as a substitute for classic operators $+$ (plus) and $\cdot$ (times) respectively. We express this analogy since it can be useful when implementing the fuzzy automata on a computer.

3.5   Extension to Words

The fuzzy transition function δ can be extended to the word-based *extended fuzzy transition function* δ^*:

$$\delta^* : Q \times \Sigma^* \times Q \rightarrow L.$$

For $w = a_1 a_2 \ldots a_n \in \Sigma^*$ the fuzzy transition matrix is defined as a composition of fuzzy relations: $\delta^*(w) = \delta_{a_1} \circ \delta_{a_2} \circ \cdots \circ \delta_{a_n}$ (from left to right).

For empty word ε we define

$$\delta^*(q_1, \varepsilon, q_2) = \begin{cases} \bigvee L & \text{if } q_1 = q_2, \\ \bigwedge L & \text{if } q_1 \neq q_2. \end{cases}$$

Note that if $L = [0,1]$, then $\bigvee L = 1$ and $\bigwedge L = 0$.

3.6   Final (Accepting) States

Function f_M is the *membership degree of word* $w = a_1 \ldots a_n$ to the fuzzy set F of final states.

$$f_M : \Sigma^* \rightarrow L,$$

$$f_M(w) = f_M(a_1, \ldots, a_n) = S \circ \delta_{a_1} \circ \cdots \circ \delta_{a_n} \circ F.$$

Note that f_M is a fuzzy set of Σ^*, but we do not use this terminology here. Instead, we use f_M to determine membership degree of a particular word w.

3.7   Epsilon Transitions

In Sect. 3.5 we defined ε-transitions for extended fuzzy transition function. We can generalize that definition to generic ε-transitions, i.e., we define a fuzzy relation δ_ε:

$$\delta_\varepsilon : Q \times Q \rightarrow L,$$

$$\delta_\varepsilon(q_1, q_2) = \delta^*(q_1, \varepsilon, q_2).$$

4   Minimization of Fuzzy Automata

4.1   The Minimization of an Automaton

One of the most important problems is the minimization of a given fuzzy automaton, i.e., how to decrease the number of states without the loss of the automaton functionality.

For a given $\lambda \in L$, let us have a partition (factor set) $Q_\lambda = \{\bar{q}_1, \ldots, \bar{q}_n\}$ of set Q, such that $\forall \, \bar{q}_i \in Q_\lambda$, $q_r, q_t \in \bar{q}_i$, $q \in Q$, and $a \in \Sigma$ holds

$$|\delta_a(q_r,q) - \delta_a(q_t,q)| < \lambda,$$
$$|\delta_a(q,q_r) - \delta_a(q,q_t)| < \lambda, \qquad (1)$$
$$|S(q_r) - S(q_t)| \quad < \lambda,$$

$$\bar{q}_i \subseteq F \text{ or } \bar{q}_i \cap F = \emptyset. \qquad (2)$$

We construct fuzzy automaton $M_\lambda = (\Sigma, Q_\lambda, \delta_\lambda, S_\lambda, F_\lambda)$, where

$$\delta_\lambda(\bar{q}, u, \bar{r}) = \delta_{\lambda u}(\bar{q}, \bar{r}) = \frac{\displaystyle\sum_{q_i \in \bar{q}} \sum_{r_j \in \bar{r}} \delta_u(q_i, r_j)}{|\bar{q}| \cdot |\bar{r}|},$$

$$S_\lambda(q) = \frac{\displaystyle\sum_{q_j \in \bar{q}} S(q_j)}{|\bar{q}|},$$

$$F_\lambda = \{\bar{q} : \bar{q} \subseteq F\},$$

and $\bar{q}, \bar{r} \in Q_\lambda$.

Theorem 1. *Let $w = a_1 a_2 \dots a_m$. Then $|f_M(w) - f_{M_\lambda}(w)| < \lambda(m+2)$.*

Proof. See Močkoř [6].

Let us describe how to use these equations: We must define the maximum word length m_0, and the maximum acceptable difference λ_0 for the words of this maximum size. Then, we can compute λ this way:

$$\lambda = \frac{\lambda_0}{m_0 + 2}. \qquad (3)$$

Having the λ value, we can perform desired automaton minimization.

4.2 An Example

Let us have fuzzy automaton $M = (\Sigma, Q, \delta, S, F)$, such that:

$$\Sigma = \{0, 1\}, \qquad\qquad Q = \{q_1, q_2, q_3, q_4, q_5\},$$

$$\delta_0 = \begin{pmatrix} 0.45 & 0.50 & 0.80 & 0.31 & 0.35 \\ 0.47 & 0.46 & 0.78 & 0.34 & 0.30 \\ 0.10 & 0.15 & 0.51 & 0.83 & 0.78 \\ 0.70 & 0.67 & 0.42 & 1.00 & 0.94 \\ 0.71 & 0.68 & 0.37 & 0.95 & 1.00 \end{pmatrix}, \qquad \delta_1 = \begin{pmatrix} 0.78 & 0.74 & 1.00 & 1.00 & 0.96 \\ 0.73 & 0.77 & 0.96 & 0.96 & 0.96 \\ 1.00 & 0.96 & 0.00 & 0.00 & 0.05 \\ 0.10 & 0.12 & 0.80 & 1.00 & 0.97 \\ 0.14 & 0.12 & 0.76 & 0.99 & 0.95 \end{pmatrix},$$

$$S = \begin{pmatrix} 1.00 & 0.15 & 1.00 & 0.85 & 0.90 \end{pmatrix}, \qquad F = \{q_3\}.$$

We want to minimize this fuzzy automaton in such way that the outgoing transition function will differ by less than 0.25 for words long 2 symbols at most.

According to (3), $\lambda_0 = 0.25$, $m_0 = 2$, so we compute $\lambda = \frac{0.25}{2+2} = 0.0625$.

Now we make the fuzzy automaton M_λ from this analysis according to (1) and (2):

$$Q_\lambda = \{\bar{q}_1, \bar{q}_2, \bar{q}_3\},$$

$$\bar{q}_1 = \{q_1, q_2\}, \qquad \bar{q}_2 = \{q_3\}, \qquad \bar{q}_3 = \{q_4, q_5\},$$

$$S_\lambda = \begin{pmatrix} 0.125 & 1.000 & 0.875 \end{pmatrix}.$$

Then, for example, we get

$$\delta_{\lambda 0}(\bar{q}_1, \bar{q}_1) = \delta_\lambda(\bar{q}_1, 0, \bar{q}_1) = \frac{1}{4}(0.45 + 0.50 + 0.47 + 0.46) = 0.47,$$

$$f_M(01) = S \circ \delta_0 \circ \delta_1 \circ F = 0.8,$$

$$f_{M_\lambda}(01) = S_\lambda \circ \delta_{\lambda 0} \circ \delta_{\lambda 1} \circ F_\lambda = 0.8 .$$

As can be seen in the example, we reduced the number of states from 5 to 3, and still the $f_M(01) = f_{M_\lambda}(01)$. Generally, according to the above formulas, $|f_M(w) - f_{M_\lambda}(w)| < 0.25$.

5 Approximate String Matching Based on Distance Functions

5.1 Hamming and Levenshtein Distance

In this section we present constructions $R(M, k)$ and $DIR(M, k)$, originally introduced by Melichar and Holub [5].

These constructions correspond to the creation of a nondeterministic automaton M' from the automaton M accepting string $P = p_1 p_2 \dots p_n$. Automaton M' accepts those strings P', whose value of P to P' distance is equivalent or lower than the given k while using distance functions R or DIR.

Distance $R(P, P')$ is called *Hamming distance* and is defined as the minimum number of symbol replacement operations in string P required for the conversion of string P into string P' (or vice versa).

Distance $DIR(P, P')$ is called *Levenshtein distance* and is defined as the minimum number of operations of symbol deletion (D), insertion (I) or replacement (R) in P required for the conversion of string P into string P' (or vice versa).

Automaton M' is called R-trellis or DIR-trellis as the case may be. The construction of these automata is described, for example, in the paper [5]. The trellis construction is a crucial part in the approximate string matching, so we need to generalize the trellis construction to our fuzzy automata. We will do in the following paragraphs.

5.2 Construction of Fuzzy Trellis from a Non-fuzzy One

Let us have *similarity function s*, which defines similarity level between each pair of input symbols:

$$s : \Sigma \times \Sigma \to L \tag{4}$$

$$s(a_i, a_j) = \begin{cases} \bigwedge L \text{ if } a_i \text{ and } a_j \text{ are fully different,} \\ \bigvee L \text{ if } a_i \text{ and } a_j \text{ are fully equal,} \\ \text{other value} \in L \text{ if } a_i \text{ and } a_j \text{ are similar.} \end{cases} \tag{5}$$

We usually also define $s(a_i, a_j) = s(a_j, a_i)$, but it is generally not required. If we do not obey this rule, we get an asymmetric relation, which is less common in real applications, but still mathematically correct.

Now we can define fuzzy transition function δ' as

$$\delta'_a(q_i, q_j) = \bigvee_{b \in \Sigma} (s(a, b) \cdot \delta_b(q_i, q_j)) \qquad \forall q_i, q_j \in Q, a \in \Sigma. \tag{6}$$

We use (6) to construct a fuzzy trellis $M' = (\Sigma, Q, \delta', S, F)$ from the given similarity function s and a given non-fuzzy trellis $M' = (\Sigma, Q, \delta, S, F)$. This construction is generic and not limited to just R-trellis and DIR-trellis presented above.

5.3 An Example

Let us show an example of fuzzy trellis construction for Hamming distance.

Let us have $\Sigma = \{\text{'a'}, \text{'b'}, \text{'c'}\}$, and automaton $M' = (\Sigma, Q, \delta, S, F)$ is a R-trellis based on word $w_0 = \{\text{'ab'}\}$.

$$\delta_a = \begin{pmatrix} 1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \end{pmatrix} \quad \delta_b = \begin{pmatrix} 1\,0\,0\,1\,0 \\ 0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0 \end{pmatrix} \quad \delta_c = \begin{pmatrix} 1\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \end{pmatrix}$$

Now we are going to construct fuzzy R-trellis of this R-trellis. Let us say symbols 'a' and 'c' are a bit similar, while the other ones are pairwise different.

$$s = \begin{pmatrix} 1\,0\,* \\ 0\,1\,0 \\ *\,0\,1 \end{pmatrix}, * \in L$$

We construct fuzzy transition function δ'.

$$\delta_a = \begin{pmatrix} 1\,1\,0\,*\,0 \\ 0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \end{pmatrix} \quad \delta_b = \begin{pmatrix} 1\,0\,0\,1\,0 \\ 0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0 \end{pmatrix} \quad \delta_c = \begin{pmatrix} 1\,*\,0\,1\,0 \\ 0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \end{pmatrix}$$

Now we can compare words $w_1 = \{\text{'bb'}\}$ and $w_1 = \{\text{'cb'}\}$ to the original word w_0.

If we use the original Hamming function distance R, we get $R(w_0, w_1) = 1$ and $R(w_0, w_2) = 1$. In this case the words w_1 and w_2 have got the same level of similarity to the w_0.

If we say that symbols 'a' and 'c' are a bit similar, we can define for example $* = s(\text{'a'},\text{'c'}) = 0.3$. Now $f_{M'}(w_1) = 0$, while $f_{M'}(w_2) = 0.3$, and we can see that w_0 is more similar to w_2 than w_1. This is how a fuzzy trellis works.

5.4   Another Variants of Similarity Function

It may be better to use a different approach to similarity function than the one shown in formula 5. Formally, we stay at the same definition (see formula 4), but use a non-zero values for fully different symbols. This time, we define a minimum value s_{min} which is assigned to $s(a_i, a_j)$ whenever i and j and fully different, and use greater values than s_{min} when i and j are similar. This modified approach performed better in our experiments.

6   Conclusion

We described the construction of a fuzzy automaton for string matching and searching for patterns in strings. Our approach allows defining different similarity levels for particular pairs of symbols, which can be very useful in several applications. Specifically, we expect immediate applicability in the field of DNA string matching. We also did some research in the field of fuzzy Aho-Corasick automata (ACA), proving that the classic ACA can be generalized to fuzzy ACA. This generalization brings our fuzzy based approximate string matching to another set of classic pattern searching applications.

In the future we want to focus on the minimization of fuzzy automata and compare fuzzy automata with other related techniques, e.g. constructions with weight function as described by Muthukrishnan [7].

References

1. Asveld, P.R.J.: A bibliography on fuzzy automata, grammars and language. Bulletin of the European Association for Theoretical Computer Science 58, 187–196 (1996)
2. Bělohlávek, R.: Fuzzy Relational Systems: Foundations and Principles. Kluwer Academic Publishers, Dordrecht (2002)
3. Gruska, J.: Foundations of Computing. International Thomson Computer Press (1997)
4. Kurtz, S.: Approximate string searching under weighted edit distance. In: Proceedings of the 3rd South American Workshop oh String Processing, pp. 156–170. Carton University Press (1996)
5. Melichar, B., Holub, J.: 6D classification of pattern matching problem. In: Proceedings of the Prague Stringology Club Workshop, pp. 24–32 (1997)

6. Močkoř, J.: Minimization of fuzzy automata. RSV FMPE (1982) (in Czech)
7. Muthukrishnan, S.: New results and open problems related to non-standard stringology. In: Proceedings of the 6th Combinatorial Pattern Matching Conference. LNCS, vol. 937, pp. 298–317. Springer, Espoo (1995)
8. Nguyen, H.T., Walker, E.A.: A First Course in Fuzzy Logic, 2nd edn. Chapman & Hall/CRC, Boca Raton (2000)

Remark on Membership Functions in Neuro-Fuzzy Systems

Krzysztof Simiński

Abstract. The sigmoidal membership function applied in the neuro-fuzzy systems with hierarchical input domain partition and gradient tuning method may deteriorate the tuning process. The function's high value plateau with very low derivative's value stops the gradient based tuning procedure. This leads to less adequate models and poorer results elaborated by the system. In such systems the membership function should satisfy the condition that the low values of derivatives in respect of the function parameters should be followed by the low values of membership function itself. The points of the domain that do not fulfil this condition may only be isolated point of the domain. The function should have no high membership plateaux. The functions suitable for systems with hierarchical input domain partition are bell-like functions as Gaussian, generalised bell function, (a)symmetric π function.

Keywords: neuro-fuzzy systems, membership function, sigmoidal, Gauss.

1 Introduction

Neuro-fuzzy systems were introduced for their ability to model and generalise knowledge. In comparison with the artificial neural networks in neuro-fuzzy systems the knowledge extracted from the presented examples is stored in form of rules. This enables interpretation of elaborated knowledge by humans.

The crucial part of the fuzzy inference system (cf. Fig. 1) is the fuzzy rule base [3, 8]. The rule base is the physical representation of knowledge extracted by the neuro-fuzzy systems from presented data.

Krzysztof Simiński
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: `Krzysztof.Siminski@polsl.pl`

Fig. 1 Fuzzy inference
system [3]
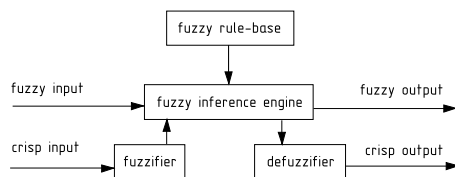
The fuzzy rule is a fuzzy implication:

$$R : X \text{ is } A \Rightarrow Y \text{ is } B, \tag{1}$$

where X and Y are linguistic variables, A and B are fuzzy linguistic terms. The fuzzy inference system aggregates the consequences of the rules and elaborates the output of the system.

The fuzzy rule base is created either on the knowledge from the expert or by automatic extraction from the presented data. The methods of rules extraction for neuro-fuzzy systems can be divided into two main classes: (1) FPTC – first premises then consequences and (2) FCTP – first consequences then premises. The first method is used in ANNBFIS [3], ANLIR [4], LOLIMOT [7], the latter in [14]. The FPTC approach requires partition of the input domain. Three kinds of methods can be enumerated:

1. grid/scatter partition,
2. clustering, and
3. hierarchical split.

In grid/scatter partition and clustering the attribute values are divided into bounded intervals so it is natural and common to apply bell membership function, e.g., Gaussian function:

$$g(x) = \exp\left(-\frac{(x-c)^2}{2s^2}\right). \tag{2}$$

In hierarchical (tree-like) domain partition it seems reasonable to use half-bounded sigmoidal functions. The are many sigmoidal functions as logistic function

$$f(x) = \frac{1}{1 + e^{-b(x-c)}}, \tag{3}$$

arc-tangent, hyperbolic tangent, Gompertz function or some algebraic functions as for instance $y = x/\sqrt{1+x^2}$. In further deliberation we will, without loss of generality, confine ourselves to the Gaussian and logistic function defined above.

2 Tuning

In neuro-fuzzy systems the parameters of the extracted rules can be tuned. This enables elaboration of more precise models better fitting the presented data. The

parameters of the rules' consequences depend on the applied neuro-fuzzy systems (in Takagi-Sugeno-Kang system the consequences are constituted by moving fuzzy singletons [13, 12], in Mamdami-Assilan system – the isosceles triangle fuzzy sets [6], in ANNBFIS [3], ANLIR [4] – moving isosceles triangle fuzzy sets). The parameters of the rules' premises are the parameters of the membership functions.

Both global optimizing techniques as evolutionary algorithms [2], simulated annealing [1] and local optimizing techniques are used in neuro-fuzzy systems. In systems using gradient optimizing method for tuning model's parameters the fuzzy set membership functions have to be differentiable. It is *conditio sine qua non*, the undifferentiable functions (e.g., triangle, trapezoid) cannot be used in such systems. The sigmoidal functions seems the most natural candidate for modeling the half-bounded interval. In tuning the parameter θ its value is modified according to the formula

$$\theta \leftarrow \theta - \eta \frac{\partial E}{\partial \theta}, \tag{4}$$

where E is a optimized criterion (for example the squared error value). In tuning of the parameter p of membership function μ the above formula becomes

$$p \leftarrow p - \eta \frac{\partial E}{\partial p} = p - \eta \frac{\partial E}{\partial \mu} \frac{\partial \mu}{\partial p}. \tag{5}$$

The partial derivatives for the Gaussian g (cf. (2)) and the logistic f (cf. (3)) functions are:

$$\frac{\partial g}{\partial c}(x) = 2g(x)\frac{x-c}{s^2}, \tag{6}$$

$$\frac{\partial g}{\partial s}(x) = 2g(x)\frac{(x-c)^2}{s^3}, \tag{7}$$

$$\frac{\partial f}{\partial b}(x) = (x-c)f^2(x)e^{-b(x-c)}, \tag{8}$$

$$\frac{\partial f}{\partial c}(x) = f^2(x)be^{-b(x-c)}. \tag{9}$$

The graphs in Figs. 2 and 3 show respectively the Gaussian function g (2) and the logistic function f (3) and their derivatives with respect to the functions' parameters. When using sigmoidal functions (cf. Fig. 3) three situation should be analysed:

1. The object has low membership so the derivatives represented by (8) and (9) are also small (e.g., for $x < -4$ in Fig. 3), so the parameter is not supposed to be changed. This is an expected behaviour.
2. The object with a certain attribute value has a high membership and the value of derivative is also high (e.g., for $0.5 < x < 1.5$ in Fig. 3). If the error is high the factor $\partial E/\partial \mu$ (cf. (5)) is high and the change of p is significant. This is also an expected behaviour.
3. The object has high membership but the value of the derivative is low (e.g., for $x > 4$ in Fig. 3). If the error is significant the parameter value should be corrected,
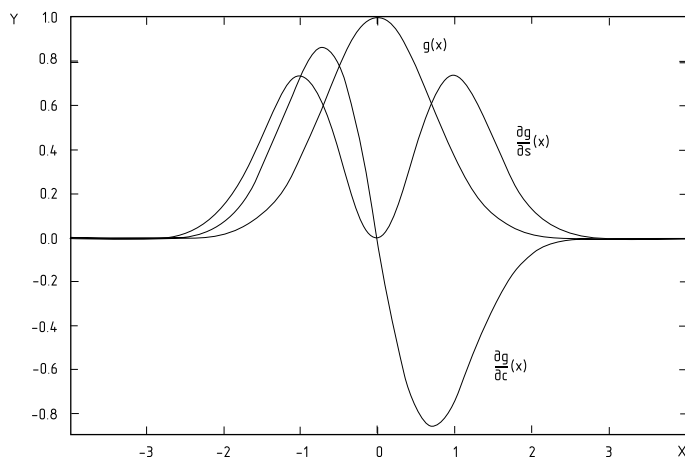
Fig. 2 Gauss function g (cf. (2)) with $c = 0$ and $s = 1$ and its derivatives $\partial g / \partial c$ and $\partial g / \partial s$
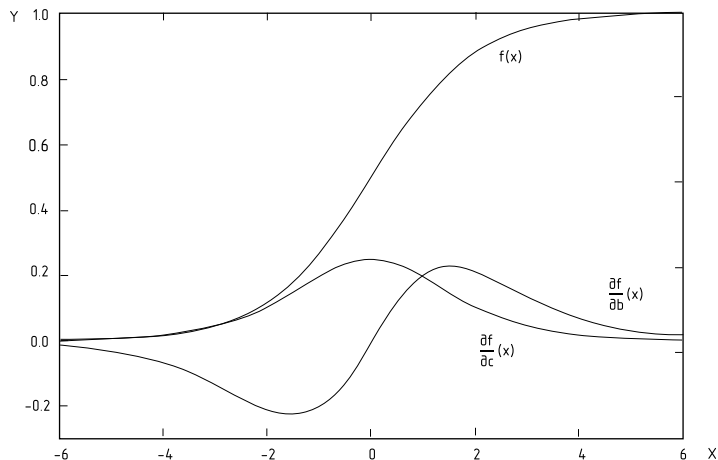
Fig. 3 Logistic function f (cf. (3)) with $c = 0$ and $b = 1$ and its derivatives $\partial f / \partial c$ and $\partial f / \partial b$

because the rule elaborates poor result, but will not be corrected for the absolute value of the derivative is very low.

The situation described in point 3 may lead to the deterioration or even halt of tuning process on the function's high membership plateaus. The experiments reveal that the values of derivatives are sometimes very small or even zero whereas the error is relatively high. This makes the tuning process extremely inefficient. This short reasoning leads to the following conclusions:

1. The membership function has to be differentiable. This condition is required by the gradient method.

2. When the absolute value of the first derivative (with respect to parameter p) of the function is low, the value of the function should also be low. Condition W:

$$\underset{0<\varepsilon_1\ll1}{\forall} \quad \underset{0<\varepsilon_2\ll1}{\exists} \quad \left|\frac{\partial f}{\partial p}(x)\right| < \varepsilon_1 \Rightarrow f(x) < \varepsilon_2 \tag{10}$$

3. It is permitted that the condition W expressed in conclusion 2 is not satisfied only for isolated points of the domain.

The functions satisfying the above expressed conditions are Gaussian, generalised bell, (a)symmetric π or function.

3 Experiments

The experiments verifying the theoretical consideration above were conducted using the commonly known data set depicting the concentration x of leukocytes described by Mackey-Glass equation [5]:

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = \frac{ax(t-\tau)}{1+(x(t-\tau))^{10}} - bx(t), \tag{11}$$

where $a = 0.2$, $b = 0.1$, and $\tau = 17$ are constants. The tuples basing on this data series were created according to the template

$$[x(t), x(t-6), x(t-12), x(t-18), x(t+6)]. \tag{12}$$

The predicted value is the six-step prediction of leukocytes concentration. The prepared tuples are split into two series: training data (tuples 1–500) and testing data (501–1000).

The second data set used in experiments is the synthetic 'NewSinCos' data set. The tuples of this data set are $\langle x, y, z \rangle$, where x, y are randomly selected from interval $[0,1]$ and $z = 5 + x + y + \sin 5x - \cos 5y$ [10]. The dataset is split into train (250 tuples) and test (250 tuples) sets.

The systems used in experiments are neuro-fuzzy systems using Gaussian membership function: ANNBFIS [3], HS-47 [9], HS-65 [11] and the T-7 system. The T-7 system is similar to the HS-47 system, the main difference being the kind of membership function used in premises of the fuzzy rules. The T-7 system employs the logistic function (cf. (3)) whereas the other systems – the Gaussian one. The gradient method is applied in tuning procedure of the membership functions in all above mentioned systems.

Table 1 shows exemplary values of the derivatives of membership functions with the respect to the function's parameters. The derivative values in system with logistic functions are significantly lower in comparison with systems implementing Gaussian functions. The tuning process in the T-7 system is practically stopped because of low derivatives values what causes poorer results of root mean square error for knowledge generalisation. The experiments show that in systems with

Table 1 The values of derivatives of membership functions with respect to the functions' parameters in neuro-fuzzy systems with gaussian membership fuctions (ANNBFIS, HS-47, HS-65) and logistic functions (T-7). The value of root mean square error (RMSE) for knowledge generalisation is put in the separated line below derivatives values

function:	Gaussian			logistic
system:	ANNBFIS	HS-47	HS-65	T-7
dataset:	*leukocyte concentration dataset*			
derivatives:	-3.6909×10^{-2}	1.6313×10^{-1}	1.5199×10^{-2}	8.8217×10^{-19}
	-4.1427×10^{-4}	-1.3577×10^{-1}	-4.9980×10^{-3}	-1.3732×10^{-19}
	-3.3226×10^{-3}	-1.8563×10^{-1}	-1.6573×10^{-2}	-6.5512×10^{-20}
	1.6992×10^{-1}	1.7296×10^{-1}	-1.7367×10^{-2}	4.3291×10^{-17}
	-6.9436×10^{-2}	-1.8323×10^{-3}	-5.7337×10^{-3}	-2.0075×10^{-17}
	-1.1557×10^{-1}	-9.7728×10^{-3}	1.0052×10^{-2}	-7.0704×10^{-19}
	-6.0356×10^{-2}	-6.4101×10^{-4}	1.3020×10^{-2}	1.3139×10^{-19}
	5.2470×10^{-2}	-2.3489×10^{-2}	-3.0672×10^{-3}	5.3000×10^{-20}
	-1.2751×10^{-1}	2.0228×10^{-2}	-1.4404×10^{-2}	5.3000×10^{-20}
	-1.9950×10^{-2}	2.0423×10^{-1}	-4.8991×10^{-4}	-4.4359×10^{-17}
RMSE =	0.004333	0.004673	0.004629	0.018849
dataset:	*sincos synthetic dataset*			
derivatives:	4.9284×10^{-2}	-2.1934×10^{-2}	2.4487×10^{-3}	6.7303×10^{-39}
	4.7441×10^{-2}	-3.0035×10^{-2}	-8.9443×10^{-3}	-3.6502×10^{-38}
	-1.5809×10^{-2}	1.8216×10^{-2}	5.4377×10^{-2}	-2.9444×10^{-39}
	-3.4771×10^{-2}	-1.3012×10^{-1}	-1.0389×10^{-1}	-4.6127×10^{-42}
	-3.9681×10^{-2}	-3.1163×10^{-3}	4.6218×10^{-2}	3.6765×10^{-50}
	-5.2811×10^{-2}	4.6634×10^{-3}	-5.3872×10^{-2}	-5.4380×10^{-10}
	4.2667×10^{-2}	5.0169×10^{-2}	1.3843×10^{-1}	-5.8848×10^{-13}
	-3.7130×10^{-3}	-1.0487×10^{-2}	8.0062×10^{-2}	4.5805×10^{-20}
	-3.6810×10^{-2}	-6.0904×10^{-2}	2.3279×10^{-2}	7.9412×10^{-27}
	1.7258×10^{-2}	-2.0508×10^{-2}	-1.4664×10^{-1}	-3.9934×10^{-13}
RMSE =	0.009775	0.018106	0.005987	0.277190

hierarchical input domain partition it is better to use the Gaussian than sigmoidal functions. Using of sigmoidal functions stops the tuning procedure because of low and very low derivative values.

4 Conclusions

The neuro-fuzzy systems use the membership functions in premises of the fuzzy rules. Two main classes of function can be used: bell-like for bounded interval values of attributes and sigmoidal for half-bounded attributes. In neuro-fuzzy systems with hierarchical input domain partition using gradient method for parameter tuning

the application of sigmoidal functions may cause deterioration of tuning of membership function's parameters. This leads to less adequate models and poorer results elaborated by the system.

In such systems the membership function should satisfy the condition that the low values of derivatives in respect of the function parameters should be accompanied by the low values of membership function itself. The points of the domain that do not fulfil this condition may only be isolated point of the domain. The function should have no high membership plateaux.

The functions suitable for systems with hierarchical input domain partition are bell-like functions as Gaussian, generalised bell function, (a)symmetric π function.

References

1. Czabański, R.: Extraction of fuzzy rules using deterministic annealing integrated with ε-insensitive learning. International Journal of Applied Mathematics and Computer Science 16(3), 357–372 (2006)
2. Czekalski, P.: Evolution-fuzzy rule based system with parameterized consequences. International Journal of Applied Mathematics and Computer Science 16(3), 373–385 (2006)
3. Czogała, E., Łęski, J.: Fuzzy and Neuro-Fuzzy Intelligent Systems. In: Studies in Fuzziness and Soft Computing. Physica-Verlag, Heidelberg (2000)
4. Łęski, J.: Systemy neuronowo-rozmyte. Wydawnictwa Naukowo-Techniczne, Warsaw (2008)
5. Mackey, M.C., Glass, L.: Oscillation and chaos in physiological control systems. Science 197(4300), 287–289 (1977)
6. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies 7(1), 1–13 (1975)
7. Nelles, O., Fink, A., Babuška, R., Setnes, M.: Comparison of two construction algorithms for Takagi-Sugeno fuzzy models. International Journal of Applied Mathematics and Computer Science 10(4), 835–855 (2000)
8. Rutkowski, L., Cpałka, K.: A general approach to neuro-fuzzy systems. In: Proceedings of the IEEE International Conference on Fuzzy Systems, pp. 1428–1431 (2001)
9. Simiński, K.: Neuro-fuzzy system with hierarchical domain partition. In: Proceedings of the International Conference on Computational Intelligence for Modeling, Control and Automation, Vienna, Austria (2008)
10. Simiński, K.: Data noise reduction in neuro-fuzzy systems. In: Kurzyński, M., Woźniak, M. (eds.) Computer Recognition Systems. Advances in Intelligent and Soft Computing, vol. 3, pp. 203–210. Springer, Heidelberg (2009)
11. Simiński, K.: Patchwork neuro-fuzzy system with hierarchical domain partition. In: Kurzyński, M., Woźniak, M. (eds.) Computer Recognition Systems. Advances in Intelligent and Soft Computing, vol. 3, pp. 13–20. Springer, Heidelberg (2009)
12. Sugeno, M., Kang, G.T.: Structure identification of fuzzy model. Fuzzy Sets and Systems 28(1), 15–33 (1988)
13. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modeling and control. IEEE Transactions on Systems, Man and Cybernetics 15(1), 116–132 (1985)
14. Tsukamoto, Y.: An approach to fuzzy reasoning method. In: Gupta, M.M., Ragade, R.K., Yager, R.R. (eds.) Advances in Fuzzy Set Theory and Applications, pp. 137–149 (1979)

Capacity-Based Definite Rough Integral and Its Application

Puntip Pattaraintakorn, James F. Peters, and Sheela Ramanna

Abstract. This paper introduces an extension of the original capacity-based rough integral defined over a specific interval. The approach hearkens back to the pioneering work on capacities and the generalization of the Lebesgue integral by Gustav Choquet during the 1950s. Variations in the definition of the capacity function has led to various forms of the discrete Choquet integral. In particular, it is the rough capacity function (also called a rough membership function) introduced by Zdzisław Pawlak and Andrzej Skowron during the 1990s that led to the introduction of a capacity-based rough integral. By extension of the original work on the rough integral introduced in 2000, a discrete form of a capacity-based definite rough integral is introduced in this paper. This new form of the rough integral provides a means of measuring the relevance of functions representing features useful in the classification of sets of sample objects.

Keywords: capacity, Choquet integral, feature selection, rough set theory, rough membership function, definite rough integral.

1 Introduction

During the early 1990s, Zdzisław Pawlak introduced a discrete form of the definite Riemann integral of a continuous, real function defined over intervals representing

Puntip Pattaraintakorn
Department of Mathematics and Computer Science, Faculty of Science,
King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand 10520
e-mail: kppuntip@kmitl.ac.th

James F. Peters
Department of Electrical and Computer Engineering,
University of Manitoba Winnipeg, Manitoba R3T 5V6 Canada
e-mail: jfpeters@ee.umanitoba.ca

Sheela Ramanna
Department of Applied Computer Science, University of Winnipeg,
Winnipeg, Manitoba R3B 2E9 Canada
e-mail: s.ramanna@uwinnipeg.ca

equivalence classes [6]. Since that time, work on various forms of rough integrals has continued fairly steadily (see, e.g., [5, 8, 9, 10]). The original integral was called a *rough integral* because it was defined over intervals represented equivalence classes in a partition of sequences of reals defined by an equivalence relation.

Based on work on the Choquet integral $(C) \int f \, d\mu$ considered in the context of rough sets [12], a new form of discrete rough integral $\int f \, d\mu_x^B$ was introduced [8, 9] and elaborated in [10]. The Choquet integral is a generalization of the Lebesgue integral defined relative to a capacity μ [1, 2]. A *capacity* is a function μ that assigns a non-negative real number to every subset of a finite set X and satisfies $f(\emptyset) = 0$ [4]. When the discrete form of the Choquet integral $(C) \int f \, d\mu$ is defined relative to a finite universe, the Lebesgue integral reduces to a (convex) linear combination, where each individual integrand function value is weighted with a capacity function value [2].

This article introduces a new form of the Choquet integral called a *capacity-based definite rough integral* because its capacity is a function defined relative to equivalence classes. The extension of the capacity-based rough integral has a number applications. In particular, we show in this article how this integral can be used in feature selection with the DRI tool implemented in MATLAB®.

This article is organized as follows. A capacity-based rough integral is defined in Sect. 3. A discrete form of the definite rough integral (DRI) is defined in Sect. 4. An illustration of the application of the DRI is presented in Sect. 5.

2   Rough Capacity Function

Rough capacity functions were introduced during the mid-90s [11]. A rough capacity function returns the degree of overlap between a fixed set containing objects of interest and a set of sample objects.

Definition 1 (Rough Capacity Function). Let $S = (\mathcal{O}, \mathcal{F})$ denote an information system. Assume $X \subseteq \wp(\mathcal{O})$, $B \subseteq \mathcal{F}$, $x \in X$ and $[x]_B \subseteq X/ \sim_B$. The capacity $\mu_x^B : \wp(\mathcal{O}) \longrightarrow [0,1]$ is defined:

$$\mu_x^B(X) = \begin{cases} \frac{|X \cap [x]_B|}{|[x]_B|}, & \text{if } X \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The capacity μ_x^B is an example of a set function, i.e., a function whose domain is a family of sets [3]. This set function measures the degree of overlap between the set X and the class $[x]_B$, i.e., the extent that X is covered by $[x]_B$. Recall that $[x]_B$ is a set of objects having matching descriptions. This is important in evaluating the set X, since $\mu_x^B(X)$ is a measure of the extent that the objects in X are part of the classification represented by $[x]_B$. In the context of capacity-based integrals, the function μ_x^B is a source of weights, i.e., degree of importance of each set X in a weighted sum for the discrete form of the integral.

It can also be observed that μ_x^B is a real-valued set function that is additive. That is, for X, X' in $\wp(\mathcal{O})$, it can be shown that

$$\mu_x^B(X \cup X') = \mu_x^B(X) + \mu_x^B(X').$$

3 Capacity-Based Discrete Integrals

This section gives a brief introduction to the Choquet integral, which eventually led to the introduction of a capacity-based rough integral.

3.1 Discrete Choquet Integral

Recall that the Choquet integral $(C) \int f \, d\mu$ is a generalization of the Lebesgue integral defined with respect to a non-classical measure μ called a capacity. Also recall that a capacity is a real-valued set function

$$\mu : \wp(X) \longrightarrow \Re^+,$$

such that $\mu(\emptyset) = 0$ and $X' \subset X'' \subset \wp(X)$ implies $\mu(X') \leq \mu(X'')$ (monotonicity). When the Choquet integral is defined relative to a finite sets, then the Choquet integral reduces to a weighted sum that has a variety of applications, especially in multi-criteria decision-making (see, e.g., [1, 2]).

Definition 2 (Discrete Choquet Integral [2]). Let μ be a capacity defined on a finite set X. The discrete Choquet integral of a function $f : X \longrightarrow \Re^+$ with respect to capacity μ is defined by

$$(C) \int f \, d\mu = \sum_{i=1}^n \left[f\left(x_{(i)}\right) - f\left(x_{(i-1)}\right) \right] \cdot \mu(X_{(i)}),$$

where $\cdot_{(i)}$ denotes a permuted index so that $0 \leq f\left(x_{(1)}\right) \leq f\left(x_{(2)}\right) \leq \cdots \leq f\left(x_{(i)}\right) \leq \cdots \leq f\left(x_{(n)}\right) \leq 1$. Also, $X_{(i)} = \left\{x_{(i)}, \ldots, x_{(n)}\right\}$, and $f(x_{(0)}) = 0$.

3.2 Discrete Rough Integral

The introduction of the rough capacity function μ_x^B paved the way for a discrete rough integral $(P) \int f \, d\mu_x^B$ named after Zdzisław Pawlak. This rough integral is a variation of the discrete Choquet integral [1, 2, 7, 9, 10].

Definition 3 (Discrete Rough Integral). Let μ_x^B be a rough capacity function defined on a finite set X. The discrete rough integral of a function $f : X \to \Re^+$ with respect to capacity μ_x^B is defined by

$$(P) \int f d\mu_x^B = \sum_{i=1}^{n} \left[f\left(x_{(i)}\right) - f\left(x_{(i-1)}\right) \right] \cdot \mu_x^B(X_{(i)}),$$

where $\cdot_{(i)}$ denotes a permuted index so that $0 \le f\left(x_{(1)}\right) \le f\left(x_{(2)}\right) \le \cdots \le f\left(x_{(i)}\right) \le \cdots \le f\left(x_{(n)}\right) \le 1$. Also, $X_{(i)} = \{x_{(i)}, \ldots, x_{(n)}\}$, and $f(x_{(0)}) = 0$.

If f is non-negative, then $(P) \int f d\mu_x^B$ represents the lower approximation of the area under the graph of f.

Proposition 1. *If $Min\mu \le \mu_x^B(X_{(i)}) \le Max\mu$, $1 \le i \le n$, then $0 \le (P) \int f \, d\mu_x^B \le Max\mu$.*

Proof

$$
\begin{aligned}
(P) \int f \, d\mu_x^B &= \sum_{i=1}^{n} \left[f\left(x_{(i)}\right) - f\left(x_{(i-1)}\right) \right] \cdot \mu_x^B(X_{(i)}) \\
&\le \sum_{i=1}^{n} \left[f\left(x_{(i)}\right) - f\left(x_{(i-1)}\right) \right] \cdot Max\mu \\
&= Max\mu \cdot \left[\left(f\left(x_{(1)}\right) - f\left(x_{(0)}\right) \right) + \left(f\left(x_{(2)}\right) - f\left(x_{(1)}\right) \right) + \right. \\
&\quad \left. \cdots + \left(f\left(x_{(n)}\right) - f\left(x_{(n-1)}\right) \right) \right] \\
&= Max\mu \cdot \left[-f\left(x_{(0)}\right) + f\left(x_{(n)}\right) \right] \\
&\le Max\mu \qquad \text{(because } f(x_{(0)}) = 0 \text{ and } f\left(x_{(n)}\right) \le 1).
\end{aligned}
$$

It can be proven in a straightforward way that $(P) \int f \, d\mu_x^B \ge 0$. $\qquad\square$

Consider a specialized capacity $\mu_x^{\{\phi\}}$ defined in terms of a single function $\phi \in B$, where B is a set of functions representing features of objects in a finite set X.

Proposition 2. *[9] Let $0 < s \le r$ and $\phi \in B$. If $f(x) \in [s,r]$ for all $x \in X$, then $(P) \int f \, d\mu_x^{\{\phi\}} \in (0, r]$.*

4 Definite Discrete Rough Integral

In this section, we introduce a discrete form of definite rough integral (DRI) of a function f denoted by $\int_a^b f \, d\mu_x^B$, where $\cdot_{(i)}$ is a permuted index and a, b such that $x_{(1)} \le a \le b \le x_{(n)}$ are the lower and upper integral limits, respectively. The limits on the rough integral specify the interval over which a function f is integrated and it is assumed that f is continuous over $[a,b]$. This integral is defined in terms of an upper integral $\overline{\int_a^b} f \, d\mu_x^B$ and a lower integral $\underline{\int_a^b} f \, d\mu_x^B$.

$$\int_a^b f \, d\mu_x^B = \overline{\int_a^b} f \, d\mu_x^B - \underline{\int_a^b} f \, d\mu_x^B.$$

The discrete forms of the lower and upper integral are defined w.r.t. $[x_1]_B$ in (2) and (3), respectively.

$$\underline{\int_a^b} f \, d\mu_{x_1}^B = \sum_{i=a}^b \left[f(x_{(i-1)}) \right] \Delta_{(i)} \mu_x^B, \tag{2}$$

$$\overline{\int_a^b} f \, d\mu_{x_1}^B = \sum_{i=a}^b \left[f(x_{(i)}) \right] \Delta_{(i)} \mu_x^B, \tag{3}$$

where $\Delta_{(i)} \mu_x^B$ is defined in (4).

$$\Delta_{(i)} \mu_x^B = \left| \mu_x^B(X_{(i)}) - \mu_x^B(X_{(i-1)}) \right|. \tag{4}$$

Definition 4 (Definite Discrete Rough Integral). Let μ_x^B denote a rough capacity function with domain $X_{(i)}$ that is the set

$$X_{(i)} = \left\{ x_{(i)}, x_{(i+1)}, \ldots, x_{(n)} \right\},$$

where $\cdot_{(i)}$ is a permuted index, $X_{(0)} = \emptyset$, and a, b such that $x_{(1)} \le a \le b \le x_{(n)}$ are the lower and upper integral limits, respectively. It is assumed that $f\left(x_{(0)}\right) = 0$. The difference $\Delta_{(i)} \mu_x^B$ is defined in (4) relative to the set $X_{(i)}$. The discrete definite rough integral of $f : X \to \Re^+$ is defined by

$$\int_a^b f \, d\mu_x^B = \overline{\int_a^b} f \, d\mu_x^B - \underline{\int_a^b} f \, d\mu_x^B,$$

where the lower and upper integrals are defined in (2) and (3), respectively.

4.1 Interpretation

Observe that the capacity function μ_x^B is defined in terms of a set of functions B representing the features of sample objects of interest. The set B provides a basis for the relation $\sim_B$ that defines a partition of the sample objects X (source of integral limits). Then a function f is integrated with respect to μ_x^B. In the discrete form of the DRI, μ_x^B provides a weight on each summand $X_{\{i\}}$, a set of sample objects. The capacity μ_x^B computes the degree of overlap between a set X and a class $[x]_B$ representing objects that have been classified relative to the features represented by B (see Sect. 2). In effect, B is a source of criteria for grouping together objects matching the criteria represented by B. Hence, the definite rough integral indicates the importance and relevance of a function integrated with respect to μ_x^B. Hence, if a function ϕ representing an object feature is integrated with respect to μ_x^B, the DRI provides an effective means of selecting features that can be used to discriminate objects. In effect, the definite rough integral is useful for feature selection within the prescribed limits of the integral. The novelty here is that the limits on the DRI can be varied to measure varying importance of an individual feature represented by a function ϕ.

5 Feature Selection

In this section, we briefly illustrate an application of the discrete definite rough integral (DRI). For simplicity, we assume that each vector of function values used to describe a sample object is evaluated, e.g., acceptable ($d = 1$) vs. unacceptable ($d = 0$). Put $B = \{d\}$, where $d \in \{0,1\}$ in defining the capacity μ_x^B. Then any set of objects X can be partitioned using the set B.

5.1 Illustration

Next, consider, for example, a set of objects described with two functions, namely, ϕ_1, ϕ_2 representing features of objects in a sample X. For the purposes of illustration, we treat ϕ_1, ϕ_2 abstractly, i.e., without considering specific functions. Here is a partial, sample description table with the evaluation (decision) column d included. Note that $x_{(i)}$ indicates a permuted object with values in ascending order.

Table 1 Sample data

X	ϕ_1	ϕ_2	d	X	ϕ_1	ϕ_2	d
$x_{(1)} = x_4$	0.79224	29.988	0	$x_{(11)} = x_{11}$	0.92282	13.787	1
$x_{(2)} = x_3$	0.79467	30.114	0	$x_{(12)} = x_{12}$	0.93357	11.387	1
$x_{(3)} = x_2$	0.80596	27.402	0	$x_{(13)} = x_{13}$	0.94553	9.7302	1
$x_{(4)} = x_5$	0.81286	27.633	0	$x_{(14)} = x_{14}$	0.90996	13.979	1
$x_{(5)} = x_1$	0.85808	21.754	0	$x_{(15)} = x_{15}$	0.95608	7.1776	1
$x_{(6)} = x_6$	0.86020	22.866	0	$x_{(16)} = x_{16}$	0.94722	11.387	1
$x_{(7)} = x_7$	0.84569	24.43	0	$x_{(17)} = x_{17}$	0.94467	9.0222	1
$x_{(8)} = x_8$	0.87886	20.16	0	$x_{(18)} = x_{18}$	0.96424	6.3259	1
$x_{(9)} = x_9$	0.88235	19.945	0	$x_{(19)} = x_{19}$	0.92804	12.398	1
$x_{(10)} = x_{10}$	0.88094	19.227	0	$x_{(20)} = x_{20}$	0.93925	9.9671	1
				$x_{(21)} = x_{21}$	0.99435	2.3081	1

We now compute the DRI relative to $\left[x_{(1)}\right]_{\{d=0\}}$ starting with integration of ϕ_1 where $B = \{d = 0\}$. The lower integral $\underline{\int_{x_{(1)}}^{x_{(10)}} \phi_1 \, d\mu_{x_1}^{\{d=0\}}}$ is

$$\underline{\int_{x_{(1)}}^{x_{(10)}}} \phi_1 d\mu_{x_1}^{\{d=0\}} = \phi_1\left(x_{(0)}\right) \cdot \Delta_{(1)}\mu_{x_1}^{\{d=0\}} + \phi_1\left(x_{(1)}\right) \cdot \Delta_{(2)}\mu_{x_1}^{\{d=0\}} + \cdots = 0.7,$$

and the upper integral $\overline{\int_{x_{(1)}}^{x_{(10)}} \phi_1 \, d\mu_{x_1}^{\{d=0\}}}$ is

$$\overline{\int_{x_{(1)}}^{x_{(10)}}} \phi_1 \, d\mu_{x_1}^{\{d=0\}} = \phi_1\left(x_{(1)}\right) \cdot \Delta_{(1)}\mu_{x_1}^{\{d\}} + \phi_1\left(x_{(2)}\right) \cdot \Delta_{(2)}\mu_{x_1}^{\{d=0\}} + \cdots = 1.45.$$

Using these results, the definite integral $\int_{x_{(1)}}^{x_{(10)}} \phi_1 \, d\mu_{x_1}^{\{d=0\}}$ is

$$\int_{x_{(1)}}^{x_{(10)}} \phi_1 \ d\mu_{x_1}^{\{d=0\}} = \overline{\int_{x_{(1)}}^{x_{(10)}} \phi_1 \ d\mu_{x_1}^{\{d=0\}}} - \underline{\int_{x_{(1)}}^{x_{(10)}} \phi_1 \ d\mu_{x_1}^{\{d=0\}}} = 0.75.$$

Next, integrate ϕ_2 with respect to $\mu_{x_1}^{\{d=0\}}$ using class $\left[x_{(1)}\right]_{\{d=0\}}$, and obtain

$$\int_{x_{(1)}}^{x_{(10)}} \phi_2 \ d\mu_{x_1}^{\{d=0\}} = \overline{\int_{x_{(1)}}^{x_{(10)}} \phi_2 \ d\mu_{x_1}^{\{d=0\}}} - \underline{\int_{x_{(1)}}^{x_{(10)}} \phi_2 \ d\mu_{x_1}^{\{d=0\}}} = 2.3.$$

This means that for class $\left[x_{(1)}\right]_{\{d=0\}}$, the feature represented by ϕ_2 is more important than the feature represented by ϕ_1. The computations have been performed using the DRI tool (see Fig. 1) implemented in MATLAB ®. From the plots in Fig. 2, notice that there is less dispersion of the values for the plot for $\int_{x_{(1)}}^{x_{(10)}} \phi_2 \ d\mu_{x_1}^{\{d=0\}}$, i.e., the values for the successive lower and upper values are tightly groups around the $\int_{x_{(1)}}^{x_{(10)}} \phi_1 \ d\mu_{x_1}^{\{d=0\}}$ values compared with the $\int_{x_{(1)}}^{x_{(10)}} \phi_2 \ d\mu_{x_1}^{\{d=0\}}$ values.
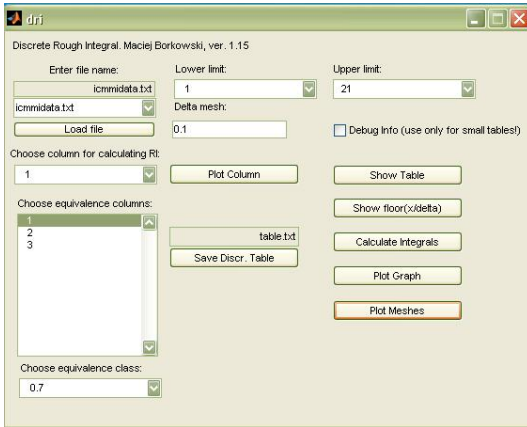
Fig. 1 DRI Tool interface

Similarly, consider the case for class $\left[x_{(1)}\right]_{\{d=1\}}$ and use the DRI to discover which function has greater importance, i.e., carries more weight. Then the DRI computed relative to ϕ_1 is

$$\int_{x_{(11)}}^{x_{(21)}} \phi_1 \ d\mu_{x_1}^{\{d=1\}} = \overline{\int_{x_{(11)}}^{x_{(21)}} \phi_1 \ d\mu_{x_1}^{\{d=1\}}} - \underline{\int_{x_{(11)}}^{x_{(21)}} \phi_1 \ d\mu_{x_1}^{\{d=1\}}} = 0.$$

Next, integrate ϕ_2 for the same class, and obtain

$$\int_{x_{(11)}}^{x_{(21)}} \phi_2 \ d\mu_{x_1}^{\{d=1\}} = \overline{\int_{x_{(11)}}^{x_{(21)}} \phi_2 \ d\mu_{x_1}^{\{d=1\}}} - \underline{\int_{x_{(11)}}^{x_{(21)}} \phi_2 \ d\mu_{x_1}^{\{d=1\}}} = 0.1.$$
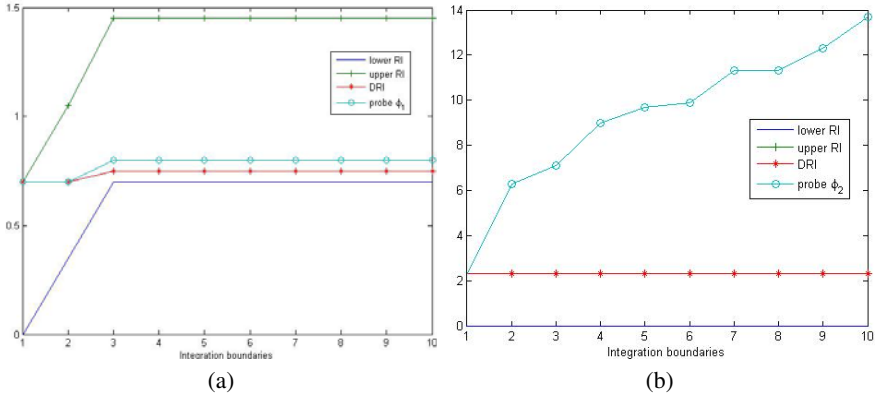
Fig. 2 DRI for ϕ_1, ϕ_2 for class $\left[x_{(1)}\right]_{\{d=0\}}$ **a** DRI $\phi_1, d = 0$. **b** DRI $\phi_2, d = 0$
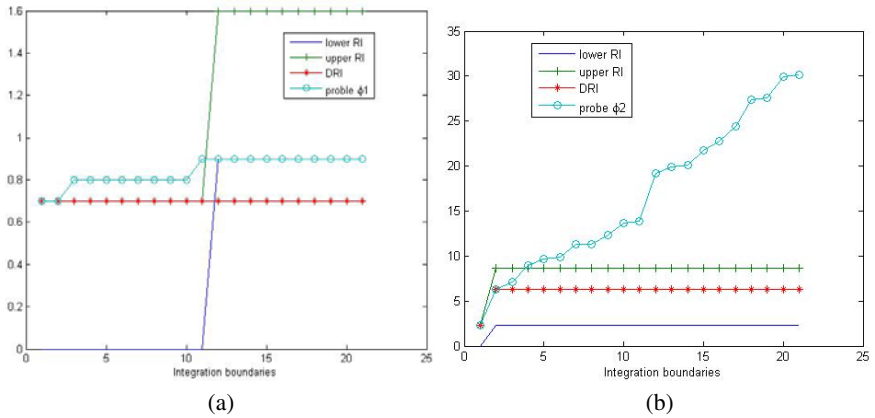
Fig. 3 DRI for ϕ_1, ϕ_2 for class $\left[x_{(2)}\right]_{\{d=1\}}$ **a** DRI $\phi_1, d = 1$. **b** DRI $\phi_2, d = 1$

We now calculate the DRI over the entire data set, for both classes $\left[x_{(2)}\right]_{\{d=0\}}$ and $\left[x_{(2)}\right]_{\{d=1\}}$ for the two features ϕ_1 and ϕ_2 respectively:

$$\int_{x_{(1)}}^{x_{(21)}} \phi_1 \, d\mu_{x_2}^{\{d=0\}} = \overline{\int_{x_{(1)}}^{x_{(21)}}} \phi_1 \, d\mu_{x_2}^{\{d=0\}} - \underline{\int_{x_{(1)}}^{x_{(21)}}} \phi_1 \, d\mu_{x_2}^{\{d=0\}} = 0.7.$$

$$\int_{x_{(1)}}^{x_{(21)}} \phi_2 \, d\mu_{x_2}^{\{d=0\}} = \overline{\int_{x_{(1)}}^{x_{(21)}}} \phi_2 \, d\mu_{x_2}^{\{d=0\}} - \underline{\int_{x_{(1)}}^{x_{(21)}}} \phi_2 \, d\mu_{x_2}^{\{d=0\}} = 6.3.$$

From this, we can conclude that the feature represented by ϕ_2 is more important than the feature represented by ϕ_1 with respect to both classes and for different equivalence classes. The plots in Figs. 2 and 3 reveal an interesting feature of the sample objects that are a source of limits on the DRI, i.e., at $x_{(12)}$ there is a sharp

change in the difference between ϕ_2 and DRI values (this is especially evident in the plot). This suggests a place to begin experimenting with different limits on the integral. This break in the DRI values also indicates a change in the evaluation of the functions representing object features.

6 Conclusion

This paper introduces the definite rough integral defined relative to a rough capacity function. It is the capacity μ_x^B that distinguishes the rough integral from the original capacity-based integral introduced by Choquet during the 1950s. In addition to the geometric interpretation of the definite rough integral already mentioned, this integral is highly significant because it offers a measure of the importance and relevance of features of sample objects. This measurement of the relevance of a feature is a byproduct of the rough capacity function that provides the basis for the rough integral. Now, with the introduction of rough integral limits, it is possible to measure the relevance of features relative to selected ranges of objects. In effect, the definite rough integral provides a new basis for feature selection in the classification of objects. Future work includes a study of the properties of the definite rough integral and its applications.

References

1. Grabisch, M.: Modelling data by the Choquet integral. In: Torra, V. (ed.) Information Fusion in Data Mining, pp. 135–148. Physica Verlag, Heidelberg (2001)
2. Grabisch, M., Murofushi, T., Sugeno, M., Kacprzyk, J.: Fuzzy Measures and Integrals. Theory and Applications. Physica Verlag, Heidelberg (2000)
3. Halmos, P.R.: Measure Theory. Van Nostrand and Co., London (1950)
4. Lehrer, E.: A new integral for capacities. Tech. rep., School of Mathematical Sciences, Tel Aviv University (2005)
5. Nguyen, H.S., Jankowski, A., Peters, J.F., Skowron, A., Szczuka, M., Stepaniuk, J.: Discovery of process models from data and domain knowledge: A rough-granular approach. In: Yao, J.T. (ed.) Novel Developements in Granular Computing: Applications of Advanced Human Reasoning and Soft Computation. Information Science Reference, Hersey (2009) (in press)
6. Pawlak, Z.: On rough derivatives, rough integrals and rough differential equations. Tech. Rep. 41/95, Institute of Computer Science, Warsaw Institute of Technology (1995)

7. Pawlak, Z., Peters, J.F., Skowron, A.: Approximating functions using rough sets. In: Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society, vol. 2, pp. 785–790 (2004)

8. Pawlak, Z., Peters, J.F., Skowron, A., Suraj, Z., Ramanna, S., Borkowski, M.: Rough measures, rough integrals and sensor fusion. In: Inuiguchi, M., Tsumoto, S., Hirano, S. (eds.) Rough Set Theory and Granular Computing, pp. 263–272. Springer, Heidelberg (2000)

9. Pawlak, Z., Peters, J.F., Skowron, A., Suraj, Z., Ramanna, S., Borkowski, M.: Rough measures: theory and applications. Tech. Rep. 1/2, Bulletin of the International Rough Set Society (2001)

10. Pawlak, Z., Peters, J.F., Skowron, A., Suraj, Z., Ramanna, S., Borkowski, M.: Rough measures and integrals: a brief introduction. In: Terano, T., Nishida, T., Namatame, A., Tsumoto, S., Ohsawa, Y., Washio, T. (eds.) JSAI-WS 2001. LNCS (LNAI), vol. 2253, pp. 375–379. Springer, Heidelberg (2001)

11. Pawlak, Z., Skowron, A.: Rough membership functions. In: Yager, R., Fedrizzi, M., Kacprzyk, J. (eds.) Advances in Dempster-Shafer Theory of Evidence, pp. 251–271. John Wiley & Sons, New York (1994)

12. Peters, J.F., Han, L., Ramanna, S.: The Choquet integral in a rough software cost decision system. In: Grabisch, M., Murofushi, T., Sugeno, M., Kacprzyk, J. (eds.) Fuzzy Measures and Integrals. Theory and Applications, pp. 392–414. Physica Verlag, Heidelberg (2000)

Part VI
Advances in Classification Methods

Classifier Models in Intelligent CAPP Systems

Izabela Rojek

Abstract. The paper presents classifier models in intelligent computer aided process planning (CAPP) systems. The models of simple classifiers and models of multiple classifiers are compared in order to obtain optimal classification. The models are tested on real data from an enterprise. Based on the classification models, the intelligent support system allows to create scenarios for selection of tools to manufacturing operations. Therefore the embedded models improve this selection. We present decision trees as models for classifcation in intelligent CAPP systems. The research was done for selected manufacturing operations: turning, milling and grinding. Models for milling operation were presented in detail.

Keywords: simple classifier, multiple classifier, classification model.

1 Introduction

At present, any knowledge is treated as the basic and the most valuable enterprise resource and a factor of enterprise development. It is a specific type of resource – in contrast to other resources it grows in proportion to its use. Currently the most difficult problem in creation of systems of knowledge acquisition is the withdrawal of knowledge and experience from enterprise employees, as they are reluctant to share their knowledge with others. That is why data mining is essential nowadays [6, 7]. Data Mining is based on three principles: data, techniques and modelling. These

Izabela Rojek

Institute of Environmental Mechanics and Applied Computer Science,

Kazimierz Wielki University in Bydgoszcz,

Chodkiewicza 30, 85-064 Bydgoszcz, Poland

e-mail: `izarojek@ukw.edu.pl`

processes cannot be automated, they need to be operated by a user. Data mining techniques – decision trees, neural networks, and regression are implemented onto different tools called Enterprise Miner. Data Mining focuses on making a client aware of the complexity of data they have at their disposal, and the possibilities of using this data effectively. For the purposes of data mining, data warehouses are constructed, collecting data from different, sometimes global sources.

The paper presents and compares classification models in the form of different decision trees. After comparison, we can say which type of decision tree models is the best for classification and selection of tools to manufacturing operations. On the basis of the decision tree model, decision rules were created. Next, these rules were used in expert system for selection of tools.

2 Simple Classifier – Decision Tree

The first researched model was a simple classifier in the form of a decision tree. The method of decision trees induction allows approximation of classification functions of discrete output values relating to certain terms or decision classes [3, 9, 8]. The model learns decision making on the basis of the examples chosen from the database by experts. In the method of decision trees induction a decision tree is created. This tree allows classification of the whole learning examples file into homogeneous classes. Decision trees generated from learning examples are often used for classifying new objects to a decision class. In this application, the trees fulfill the function of classifiers. Evaluation of tree usability to classify new objects is achieved by evaluation of classification correctness relating to the file of learning examples. And even during generating even the best trees classifiers can meet difficulties, especially if the real data included conflicting or incomplete description of learning examples. In the learning algorithms, particular branches of decision trees are developed more deeply until examples in the site are classified to a single decision class.

3 Multiple Classifiers

In recent years, we have observed a growing scientific interest in creating multiple classifiers. The idea involves integration of many simple learning algorithms in a one multiple classifier. The aim of the integration is the achievement of better classification accuracy from the one obtained as a result of separately used simple classifiers included in a multiple classifier [1, 4]. The nineties have witnessed many works concerning multiple classifiers. Experimental results confirm the height of classification accuracy for the offered systems. Integration of homogeneous and heterogeneous classifiers (or learning algorithms) could be conducted in various ways. One of two main categories of complex classifiers is the category of multiple classifiers. The multiple classifier relates to a collection of simple classifiers, the reactions of which are aggregated to one reaction of the whole system. Component classifiers

could be homogeneous or heterogeneous. In the first case classifiers are learned on diverse data. The main question concerns the ways of creating multiple classifiers that allow to achieve higher classification accuracy than in the component classifiers working separately. According to some researchers, classifiers should work with a certain degree of disagreement, i.e., the classification errors committed should not be correlated. Aggregation of solutions from separate classifiers into a global solution is done in groups or by specialization. In the first case, all classifiers partake in creating the final solution for a new object. In the second case these classifiers the specialization of which contains characteristics of classifiable entity are determined. The aggregation itself is often realized through simple voting (voice of every classifier has equal priority) or weighted voting. There are many ways of creating complex classifiers. A part of them relies on modification of learning data for separate homogenous classifiers, others assume diversity of classifier models themselves. Very few review works concerning different propositions have been written so far. Multiple classifiers are divided into classifiers founded on:

- diverse classifiers:

  - *Stacking method* – It is a model of a hierarchical, multi-layer complex classifier proposed by Wolpert [13]. Component classifiers of the first layer learn on original data and their answers are the input data to the next layer. Separate classifier giving the final decision is placed in last layer. Higher layer classifiers attempt to minimize errors committed by classifiers from lower layers.

- homogenous classifiers:

  - *Bagging method* – Model introduced by Breiman uses the idea of duplicating the learning file with the use of multiple sampling 'bootstrapping' method [2]. Every duplicated file has the same size as the original file, but as a result of sampling some learning examples are copied while others cannot multiply. Each of these sets generates a classifier using the same learning algorithm. The answers of classifiers are aggregated by simple voting.

  - *Boosting method* – The method was introduced by Freund and Schapire [5, 12]. The core of this method is the association of any learning entity with a weight, the value of which expresses its significance. Learning algorithm is run by iteration on a modified learning file. Modifications concerning changes of weights in the subsequent duplicates of learning file which focuses on the activity of learning algorithm around the so-called difficult examples, for which the classifiers produced errors in previous iterations. The final answer of classifier is based on weighted voting (here weights of voting classifiers are dependent on errors committed by the given component classifiers). Experimental comparison of complex classifiers created via bagging and boosting method was presented in [10]. According to the author of these works, the methods are effective if we use the so-called unstable learning algorithms, i.e. such which create different classifiers in an answer to slight changes in a learning file.

4 Decision Tree for Classification of Tools for Manufacturing Operations

Experiments were conducted for selected manufacturing operations: turning, milling and grinding. For each operation, the models of simple classifiers and models of multiple classifiers were drawn. For each type of a classifier, a research examining which decision tree produced best results was done. Analysis and comparison of classification models for selection of tools was carried out. The paper was illustrated with selected models. The aim of this task was the selection of a tool to given manufacturing operation on the basis of the given input parameters. It was intended to carry out historical data mining by means of decision tree methods, to find knowledge useful for description relationships between applying certain input parameter values of manufacturing operations and selecting tool for manufacturing operation, as well as to aid process engineers in selecting this tool. Evaluation produced effectiveness of tool selection with certain group of tools possible to select. The bigger the number of tools correctly classified the better. Experience was represented by database gathered in an enterprise. We suppose that representative number of cases of tools selection is located in database. Representation of learning cases occurs most often in case of notation of attribute-values. Therefore we suppose that the file of learning examples is described by means of the established attribute file which pose certain features characterizing properties of objects described in examples file. In practice, examples are represented in a form of a table where the rows of table represent examples, and columns – attributes. Additionally aside from the file of learning examples the file of test examples can be created. The format of both files is the same.

4.1 Data Preparation

The correctness of the acquired knowledge greatly depends on data examples on the basis of which methods of knowledge acquisition obtained this knowledge. The initial data processing plays an important role during learning and testing decision tree induction. At this stage, we must solve such problems as proper selection of features or right examples. Certain additional problems limiting choice of optimal feature set, such as the problem of dimensionality, data correlation, or data interrelationship, combine with the right selection of features. Learning and testing files were prepared with the aim of knowledge acquisition, aiding tool selection for manufacturing operation. Examples of tool selection divided thematically into learning and testing files, separately for turning, milling and grinding. For milling, in first stage learning files include 111 instances, whereas testing files about 70. In next stage learning files include 555 instances, whereas testing files about 70. Input attributes are nominal, ordinal and numeric type. In files, data is presented in table-oriented form, where table rows correspond to examples and columns correspond to attributes. The files are decision tables, in which the last column in table is a decision attribute (tool symbol). 'Data purification' process was carried out. An accurate data

profiling, data parsing, data verification (on level of a field, a row, a table) and data standardization was carried out. Data duplication was removed, too. The examples included in files are real tool selections carried out during design of manufacturing processes in an enterprise.

In the case of *turning*, input data includes: the kind of turning (e.g. roughing), turning cut (e.g. straight), stock symbol (e.g., St3W), type of stock (e.g., soft), turning tool structure (e.g., mono-lithic), kind of turning tool (e.g., left-cut tool). Output data is turning tool symbol (e.g., 23 600).

In the case of *grinding*, input data includes: the kind of grinding (e.g., roughing), grinding cut (e.g., surface), stock symbol (e.g., 1 053), demanded surface roughness (e.g., 40), external dimension of grinding wheel (e.g., 80), internal dimension of grinding wheel (e.g., 30), grain size (e.g., 540), structure of grinding wheel (e.g., compact). Output data is grinding wheel symbol (e.g., 56 350).

In the case of *milling*, input data includes: the kind of milling (e.g., roughing), type of machining surface (e.g., surface), stock symbol (e.g., 1 053), demanded surface roughness (e.g., 40), milling tool structure (e.g., inserted-tooth cutter), kind of milling tool clamping (e.g., arbor), dimension (e.g., 160), tooth number (e.g., 10), total length of milling tool (e.g., 300). Output data is milling tool symbol (e.g., hR 257 .1-160) [11]. The learning and testing file are in ARFF format.

4.2   Description of Classification Models

The experiment presents research carried out for simple and multiple classifiers. We used simple classifiers: Decision Stump, J48, Random Tree, REPTree, ZeroR and multiple classifiers: *LogitBoost*, *FilteredClassifier*, *AttributeSelectedClassifier*, *OrdinalClassClassifier*, *RandomCommittee*, *Bagging*, and *Stacking*. *LogitBoost* multiple classifier uses simple *DecisionStump* classifiers. *OrdinalClassClassifier*, *AttributeSelectedClassifier*, and *FilteredClassifier* use simple *J48* classifiers. *Bagging* multiple classifier uses simple *REPTree* classifiers. *Stacking* multiple classifier uses simple *ZeroR* classifiers. *RandomCommittee* multiple classifier uses simple *RandomTree* classifiers.

We compare correctly and incorrectly classified instances, mean absolute error, root mean squared error, relative absolute error and root relative squared error. Figure 1 presents parameters of classifiers for learning set with 111 instances. In the first case we used 10-fold cross-validation to evaluate models accuracy. Next we confirmed accuracy of the classifiers on the independent test data set (78 instances). Figure 2 presents correctly classified instances in percent from activity of selected classifiers. For example, the correctly classified instances obtained 82.88% for J48 and Attribute Selected classifiers, 66.67% for RandomCommittee model, 68.47% for Random Tree model and 85.59% for LogitBoost model. We confirmed accuracy of the classifiers on the independent test data set. For example, the correctly classified instances obtained 88.46% for J48 and Attribute Selected classifiers, 87.18% for RandomCommittee model, 85.90% for Random Tree and LogitBoost model.

Parameter of classifiers	Classifier type											
	Decision Stump	Logi Boost	J48 pruned tree	Attribute Selected Classifier	Filtered Classifier	Ordinal Class Classifier	REP Tree	Bagging	ZeroR	Stacking	Random Tree	Random Committee
Small learning set (111 instances) with 10-fold cross-validation												
Correctly Classified Instances	32.00	95.00	92.00	92.00	88.00	88.00	82.00	83.00	21.00	21.00	76.00	74.00
Incorrectly Classified Instances	79.00	16.00	19.00	19.00	23.00	23.00	29.00	28.00	90.00	90.00	35.00	37.00
Correctly Classified Instances %	28.83	85.59	82.88	82.88	79.28	79.28	73.87	74.77	18.92	18.92	68.47	66.67
Incorrectly Classified Instances %	71.17	14.41	17.12	17.12	20.72	20.72	26.13	25.23	81.08	81.08	31.53	33.33
Mean absolute error	0.07	0.01	0.02	0.02	0.02	0.03	0.03	0.04	0.08	0.08	0.03	0.03
Root mean squared error	0.19	0 .10	0.11	0.12	0.13	0.13	0.14	0.13	0.20	0.20	0.17	0.15
Small learning set (111 instances) with user supplied test set (78 instances)												
Correctly Classified Instances	20.00	67.00	69.00	69.00	60.00	65.00	51.00	59.00	11.00	11.00	67.00	68.00
Incorrectly Classified Instances	58.00	11.00	9.00	9.00	18.00	13.00	27.00	19.00	67.00	67.00	11.00	10.00
Correctly Classified Instances %	25.64	85.90	88.46	88.46	76.92	83.33	65.38	75.64	14.10	14.10	85.90	87.18
Incorrectly Classified Instances %	74.36	14.10	11.54	11.54	23.08	16.67	34.62	24.36	85.90	85.90	14.10	12.82
Mean absolute error	0.08	0.01	0.01	0.02	0.03	0.03	0.04	0.04	0.08	0.08	0.01	0.01
Root mean squared error	0.20	0.10	0.09	0.10	0.14	0.11	0.15	0.13	0.21	0.21	0.11	0.08

Fig. 1 Parameters of classifiers
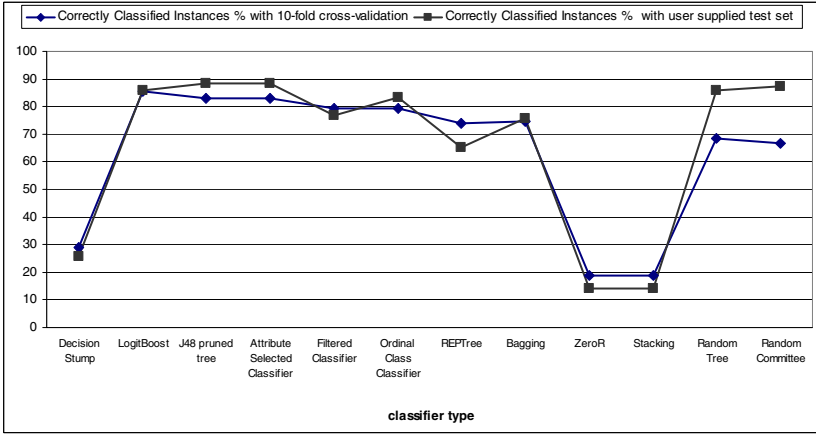
Fig. 2 Correctly classified instances for learning set with 111 instances
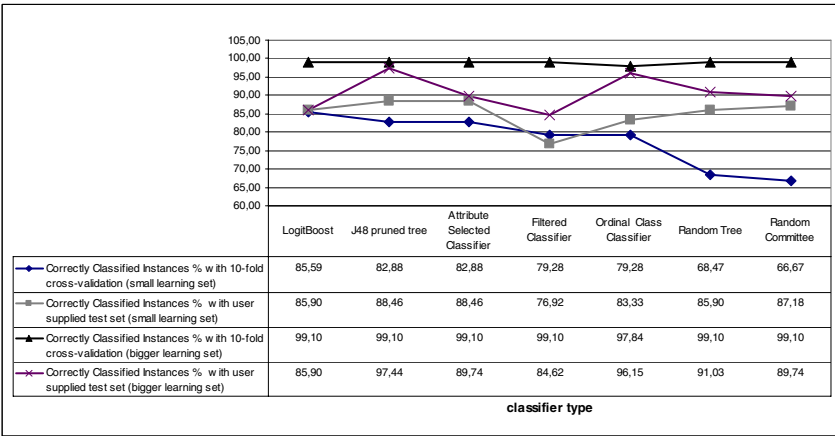
Fig. 3 Correctly classified instances for selected models

In next stage we changed learning set. We used bigger learning set with 555 instances. In the first case we used 10-fold cross-validation to evaluate models accuracy. For example, the correctly classified instances obtained 99.10% as many as six models: LogitBoost, J48, Attribute Selected classifier, Filtered Classifier, Random Tree classifier and RandomCommittee. We confirmed accuracy of the classifiers on the independent test data set. For example, the best correctly classified instances obtained 97.44% for J48 classifier and 96.15% for Ordinal Class Classifier.

Figure 3 presents correctly classified instances in percent from activity of selected classifiers for all cases: small learning set (111 instances) with 10-fold cross-validation, small learning set (111 instances) with independent test set (78 instances), bigger learning set (555 instances) with 10-fold cross-validation, bigger

learning set (555 instances) with independent test set (78 instances). Having analyzed various classification models, it was the J48 pruned tree model which proved the most effective, and then the OrdinalClassClassifier model should be listed.

5   Conclusion

The general principle of science states that if there is a possibility to choose between a complex and a basic model, the basic model should always be given priority – unless of course the more complex one is significantly better than the basic one for the particular data. This principle should also be applied to decision trees. Having analyzed various classification models, it was the J48 pruned tree model which proved the most effective, and then the OrdinalClassClassifier model should be listed. The J48 pruned tree model was reacting correctly in the case of three manufacturing operations: turning, milling and grinding. The model was tested on data from testing file and the model activity was tested for new input values.

Using decision trees as classification models, application having aid process engineer in selection of tools for manufacturing operation was implemented. The application of tool selection in dialog form queries process engineer about input attributes and answers in the form of tool symbol. The system described, enables the acquisition of data, knowledge and experience of a process engineer in its entirety. The comprehensive method of knowledge acquisition is a considerable achievement when compared to traditional expert systems based on human expert-derived knowledge. The method is substantial especially when such knowledge is inaccessible, difficult to formalize or unreliable. When acquiring new data, these models should be learnt in any moment in time. Further research into when and how to learn these models should be done.

Further research concerning the creation of classification models should be directed towards constructing hybrid classification models integrating different models in learning phase.

References

1. Bauer, E., Kohavi, R.: An empirical comparision of voting classification algorithms: bagging, boosting and variants. Machine Learning 36, 105–139 (1999)
2. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
3. Cios, K.J., Pedrycz, W., Swinarski, R.W.: Data mining methods for knowledge discovery. Kluwer, Dordrecht (1999)
4. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
5. Freund, Y.: Boosting a weak learning algorithms by majority. Information and Computation 121(2), 256–285 (1995)
6. Han, J., Kamber, M.: Data Mining: Concepts and techniques. Morgan Kaufmann, San Francisco (2000)

7. Klosgen, W., Zytkow, J.M.: Handbook of Data Mining and Knowledge Discovery. Oxford Press (2002)

8. Langley, P., Simon, H.A.: Fielded applications of machine learning. In: Michalski, R.S., Bratko, I., Kubat, M. (eds.) Machine learning and Data Mining. John Wiley & Sons, Chichester (1998)

9. Michalski, R.S., Bratko, I., Kubat, M.: Machine learning and Data Mining. John Wiley & Sons, Chichester (1998)

10. Quinlan, J.R.: Bagging, boosting and C4.5. In: Proceedings of the 13th National Conference on Artificial Intelligence, pp. 725–730 (1996)

11. Rojek, I.: Knowledge discovery for computer aided process planning systems. Polish Journal of Environmental Studies 16(4A), 268–270 (2007); Short communication

12. Schapire, R.E.: The strength of weak learnability. Machine Learning 5, 197–226 (1990)

13. Wolpert, D.: Stacked generalisation. Neural Networks 5, 241–259 (1992)

Classification Algorithms Based on Template's Decision Rules

Barbara Marszał-Paszek, Piotr Paszek, and Alicja Wakulicz-Deja

Abstract. In the paper, classification algorithms are presented. These algorithms are based on nondeterministic decision rules that are called *template's decision rules*. The conditional part of these rules is a template and the decision part is satisfactorily small set of decisions. Only rules with suficiently large support are used. The proposed classification algorithms were tested on the group of decision tables from the UCI Machine Learning Repository. Results of experiments show that the classification algorithms based on template's decision rules are often better than the algorithms based on deterministic decision rules.

Keywords: rough sets, Dempster-Shafer theory, nondeterministic decision rules, template's decision rules.

1 Introduction

The Dempster-Shafer theory [3, 9] is based on belief functions, and plausible reasoning is used to combine separate pieces of information (evidence) to calculate the probability of an event. This theory is called a mathematical theory of evidence. The rough set theory was proposed by Pawlak [8] as a mathematical tool for describing the uncertain knowledge.

The basic functions of the evidence theory were defined based on the notation from rough set theory in [4, 10]. These dependencies were used by us to find a template's decision rules in a given decision table [5].

2 Main Notions

Let $\mathbb{A} = (U, A, d)$ be a *decision table*, where $U = \{u_1, \ldots, u_n\}$ is a finite nonempty set of *objects*, $A = \{a_1, \ldots, a_m\}$ is a finite nonempty set of *conditional attributes*, and

Barbara Marszał-Paszek · Piotr Paszek · Alicja Wakulicz-Deja
Institute of Computer Science, University of Silesia,
Będzińska 39, 41-200 Sosnowiec, Poland
e-mail:{bpaszek,paszek,wakulicz}@us.edu.pl

d is the *decision attribute*. The decision d creates a partition of the universe U into decision classes $X_1, \ldots, X_{r(d)}$, where $r(d) = |\{k : \exists_{x \in U} : d(x) = k\}|$ is the number of different values of the decision attribute.

Let $\Theta_A = \{1, 2, \ldots, r(d)\}$ be the frame of discernment defined by the decision d in the decision table $\mathbb{A}$. For any $\theta \in \Theta_A$ the following equalities hold [10]:

$$Bel_A(\theta) = \frac{\left| \underline{A} \bigcup\limits_{i \in \theta} X_i \right|}{|U|}, \qquad Pl_A(\theta) = \frac{\left| \overline{A} \bigcup\limits_{i \in \theta} X_i \right|}{|U|}.$$

A *template T* [7] in a decision table is any sequence $v_1, \ldots, v_m$, where

$$v_i \in V_{a_i} \cup \{*\}.$$

The symbol '$*$' appearing in a given template means that the value of the marked attribute is not restricted by the template. Alternatively, a template can be defined as the conjunction of a certain number of descriptors. A given object matches a given template if $a_i(x) = v_i$, for each i such that $v_i \neq *$.

The following minimal templates problem (MTP) was considered [5]:

Problem 1 (Minimal Templates Problem)

- Input:

    - A decision table $\mathbb{A}$; thresholds $\varepsilon_1, \varepsilon_2 \in (0, 1)$, a natural number $1 \leq k < r(d)$.

- Output:

    - The set of minimal template's decision rules $\{T \Rightarrow \theta\}$, where T is a template, $\theta \subseteq \Theta_{\mathbb{A}_T}$, $|\theta| \leq k$, satisfying the following conditions:

$$|Pl_{\mathbb{A}_T}(\theta) - Bel_{\mathbb{A}_T}(\theta)| < \varepsilon_1 \quad \text{for} \quad \varepsilon_1 \in (0, 1), \quad \theta \subseteq \Theta_{A_T}, \qquad (1)$$

$$|Pl_{\mathbb{A}_T}(\theta)| > 1 - \varepsilon_1 \quad \text{for} \quad \varepsilon_1 \in (0, 1), \quad \theta \subseteq \Theta_{A_T}, \qquad (2)$$

$$\frac{|U_T|}{|U|} > \varepsilon_2 \quad \text{for} \quad \varepsilon_2 \in (0, 1). \qquad (3)$$

Based on (1)–(3) the following rule is obtained:

$$T \Rightarrow \theta,$$

where the conditional part of the rule is a template T, and the decision part is a set θ. If $|\theta| \geq 1$ then template's decision rule is nondeterministic rule.

In [11] it was shown that there exist information systems $S = (U, A)$ such that the set U can not be described by deterministic rules. In [6] it was shown that for any information system $S = (U, A)$ the set U can be described by nondeterministic rules. It means that the nondeterministic rules can express essentially more information encoded in information systems than the deterministic rules.

3   Classifications

In order to use the template's decision rules, the following classification methods were proposed: voting on templates, the excluding method, and the expanding method.

Voting on templates is defined in the following way: having the set of rules in the shape of $\{T \Rightarrow \theta\}$ and the object x, that we want to classify, we decide to choose:

- these rules, in that T fits to the conditional part of the object x,
- then, we choose rule $T \Rightarrow \theta$ with the largest support.

The support of the rule is the number of objects from training table matching the conditional part of the rule. The support of the rule is stored with every rule. In the classification process, in the excluding method and expanding method, we use rules that are generated in RSES [2] (minimal, covering, genetic rules) and template's decision rules. On rules from RSES, we start standard voting (each rule has as many votes as supporting objects), and on rules based on templates, we start `voting on templates`. In the classification process the information obtained from the rules and template's rules is used.

Let Θ_A be the set of possible decisions for the given decision table. The following process is started for the proposed methods:

- *Excluding classification method*:

    1. When we receive from the rules (standard voting) $\{v_i\}$ and from the templates (voting on templates) θ, and $\{v_i\} \subset \theta$ (*no conflict*), then we assign the single decision $\{v_i\}$.
    2. When we receive from the rules (standard voting) $\{v_i\}$ and from the templates (voting on templates) θ, and $v_i \notin \theta$ (*conflict*), then we assign the decision $\{v_i\}$, if the rule support is larger than the template rule support. In the opposite case, we assign the single decision from the set θ, with the largest support.
    3. When we receive from the rules (standard voting) $\{v_i\}$ and from the template's rules (voting on templates) $\emptyset$ (empty set – does not match to any templates), then we assign the decision $\{v_i\}$.
    4. When we receive from the rules (standard voting) $\emptyset$ and from the template's rules (voting on templates) θ, then we assign the single decision from θ, with the largest support.
    5. In the remaining cases the object is not classified.

- *Expanding classification method*:

    1. When we receive from the rules (standard voting) decision $\{v_i\}$ and from the templates (voting on templates) θ, and $\{v_i\} \subset \theta$ then we assign the decision θ. That is, in the process of the classification, if the new object has the decision $v_j \notin \theta$, it is a classification error.

2. When we receive from the rules (standard voting) $\{v_i\}$ and from the templates (voting on templates) θ, and $v_i \notin \theta$ (*conflict*), then we assign the decision $\{v_i\} \cup \theta$.
3. When we receive from the rules (standard voting) $\{v_i\}$ and from the template's rules (voting on templates) $\emptyset$ (empty set – does not match to any templates), then we assign the decision $\{v_i\}$.
4. When we receive from the rules (standard voting) $\emptyset$ and from the template's rules (voting on templates) θ, then we assign the decision θ.
5. In the remaining cases the object is not classified.

4　Results

The classification methods proposed in the previous section were tested on the group of decision tables from the UCI Machine Learning Repository [1]. During the tests, four classification schemes were used:

- standard voting on the rules from RSES (only rules),
- voting on templates (only templates),
- excluding method (rules and templates),
- expanding method (rules and templates).

Table 1 contains basic information about tested datasets. It contains the information about the success rates (accuracy × coverage) for all classification schemes too. For the datasets presented in Table 1, the 5-fold cross-validation method was used several times. Then, the average success rates with standard deviation were calculated.

　　Our work demonstrated that the use of the classification on templates themselves (voting on templates) improved the quality of the classifications in comparison to the rules themselves for datasets Derma, Ecoli, Glass, and Vehicle. The number of template's decision rules for the given decision table is small. Despite this, the classification error is comparable from minimal rules error rate.

　　The excluding classification method (classification on rules and templates) gave better results for the Derma and Ecoli datasets. The expanding classification method always improved the quality of the classification (reduced the error rate).

Table 1 General information on the dataset and the success rate (with standard deviation) for the 5-fold cross-validation method

Data	General Info			5-fold cross-validation – success rate			
	attr.	examples	class	rules	templates	exclude	expand
Derma	34	366	6	61.6±3.1	95.6±2.6	97.2±2.8	99.4±1.1
Ecoli	7	336	8	53.4±7.5	97.6±1.4	59.4±7.3	97.7±1.4
Glass	9	214	3	69.2±9.0	91.6±6.1	66.3±8.4	95.2±4.6
Vehicle	18	846	4	62.4±3.6	82.8±4.8	61.9±3.3	91.3±2.4

5   Conclusions

In this paper, we have demonstrated that the relationships between rough-set theory and evidence theory can be used to find nondeterministic decision rules that we called template's decision rules. The conditional part of these rules is a template and the decision part is satisfactorily small set of decisions. Such rules are used to support classification process. In the article, it was shown that template's decision rules used in the process of classification increased classification accuracy.

References

 1. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007)
 2. Bazan, J.G., Szczuka, M.S., Wojna, A., Wojnarski, M.: On the evolution of rough set exploration system. In: Tsumoto, S., Słowiński, R.W., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 592–601. Springer, Heidelberg (2004)
 3. Dempster, A.P.: Upper and lower probabilities induced from a multivalued mapping. Annals of Mathematical Statistics 38, 325–339 (1967)
 4. Grzymała-Busse, J.W.: Rough-set and Dempster-Shafer approaches to knowledge acqusition uder uncertainty - a comparison. Tech. rep., Department of Computer Science, University of Kansas (1987)
 5. Marszał-Paszek, B., Paszek, P.: Minimal templates and knowledge discovery. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS, vol. 4585, pp. 411–416. Springer, Heidelberg (2007)
 6. Moshkov, M., Skowron, A., Suraj, Z.: On maximal consistent extensions of information systems. In: Wakulicz-Deja, A. (ed.) Decision Support Systems, pp. 199–206. University of Silesia Academic Press, Katowice (2007)
 7. Nguyen, S.H., Skowron, A., Synak, P., Wróblewski, J.: Knowledge discovery in databases: Rough set approach. In: Mares, M., Meisar, R., Novak, V., Ramik, J. (eds.) Proceedings of the 7th International Congress on Fuzzy Systems Association World, vol. 2, pp. 204–209. Prague, Czech Republic (2007)
 8. Pawlak, Z.: Rough sets. International Journal of Computer and Information Science 11, 341–356 (1982)
 9. Shafer, G.: A mathematical theory of evidence. Princeton University Press, Princeton (1976)
10. Skowron, A., Grzymała-Busse, J.: From rough set theory to evidence theory. In: Yager, R.R., Kacprzyk, J., Fedrizzi, M. (eds.) Advances in the Dempster-Shafer theory of evidence, pp. 193–236. John Wiley & Sons, New York (1994)
11. Skowron, A., Suraj, Z.: Rough sets and concurrency. Bulletin of the Polish Academy of Sciences 41(3), 237–254 (1993)

Fast Orthogonal Neural Network for Adaptive Fourier Amplitude Spectrum Computation in Classification Problems

Bartłomiej Stasiak and Mykhaylo Yatsymirskyy

Abstract. Fourier amplitude spectrum is often applied in pattern recognition problems due to its shift invariance property. The phase information, which is frequently rejected, may however be also important from the classification point of view. In this paper, fast orthogonal neural network (FONN) is used to compute amplitude spectrum in an adaptable way, enabling to extract more class-relevant information from input data. The complete architecture of the neural classifier system is presented. The proposed solution is compared to standard multilayer perceptron on an artificially generated dataset, proving its superiority in terms of computational complexity and generalization properties.

Keywords: fast orthogonal neural network, amplitude spectrum, classification.

1 Introduction

Fourier amplitude spectrum is often treated as a natural feature selector in various classification tasks, i.a. due to its shift invariance property [1, 4]. In sound analysis, for example, the perception of timbre is mostly dependent on the relative amplitudes, not the phase shifts, of the harmonics. In image recognition, the change of the object/observer location, usually irrelevant from the classification point of view, is reflected by phase shifting in the frequency domain. On the other hand, the phase spectrum may also contain important information [5], which may be shown e.g. by image reconstruction from phase spectrum only. It would be therefore convenient to have an adaptable Fourier amplitude spectrum extractor, which would be able to benefit also from the phase information.

Bartłomiej Stasiak · Mykhaylo Yatsymirskyy
Institute of Computer Science, Technical University of Łódź,
Wólczańska 215, 93-005 Łódź, Poland
e-mail:{basta,jacym}@ics.p.lodz.pl

The idea of a neural network constructed on a base of a fast algorithm of an orthogonal transform has been proposed in [3] and the method of reducing the number of neural weights by application of basic operation orthogonal neurons (BOONs) has been presented in [9]. The resulting fast orthogonal neural network (FONN) has been based on fast cosine transform algorithm. In this paper we modify the FONN to enable computation of Fourier transform and we introduce special elements enabling Fourier amplitude spectrum computation. In this way we obtain a tool which can compute the Fourier amplitude spectrum in an adaptable way.

In order to verify the usefulness of the proposed network for classification purposes, a special dataset has been constructed in such a way that only half of it can be properly classified on the basis of amplitude spectrum analysis. Using the phase information is necessary to obtain better classification results. The dataset is used to compare the classification potential of the proposed fast orthogonal network and a standard multilayer perceptron [6, 8, 11].

2 The Structure of the Neural Network

The proposed neural network is based on the fast, two-stage, homogeneous algorithm of cosine transform, type II [10], given as:

$$L_N^{II}(k) = \mathrm{DCT}_N^{II}\{x(n)\} = \sum_{n=0}^{N-1} x(n) C_{4N}^{(2n+1)k},$$ (1)

where $n, k = 0, 1, \ldots, N-1; C_K^r = \cos(2\pi r/K); S_K^r = \sin(2\pi r/K)$ [7].

The algorithm may be presented in a form of a flow graph (Fig. 1), which defines the structure of the neural connections directly. Note that the input sequence $x(n)$ must be given in bit-reversed order.

The basic operations are implemented as BOONs [9]. Each BOON represented by a filled triangle contains two independent weights which may be subjected to adaptation. The trivial basic operations (white triangles) in the first stage are usually not adapted, as they do not involve any multiplication coefficients. The indices shown in Fig. 1 may be used to pre-initialize the weights of the BOONs with values: C_{4N}^k, S_{4N}^k according to Fig. 2. If pre-initialized, the network computes the cosine transform without any training (the weights denoted in Fig. 1 as s must be set to $\sqrt{(2)}/2$ in this case).

2.1 Neural Realization of Fast Fourier Transform

Two modifications of the presented neural architecture are needed in order to compute Fourier transform [10]. Firstly, special input permutation t_N must be added (independently of the bit-reversal):

$$t_N(2n) = n, \quad t_N(2n+1) = N-1-n,$$ (2)
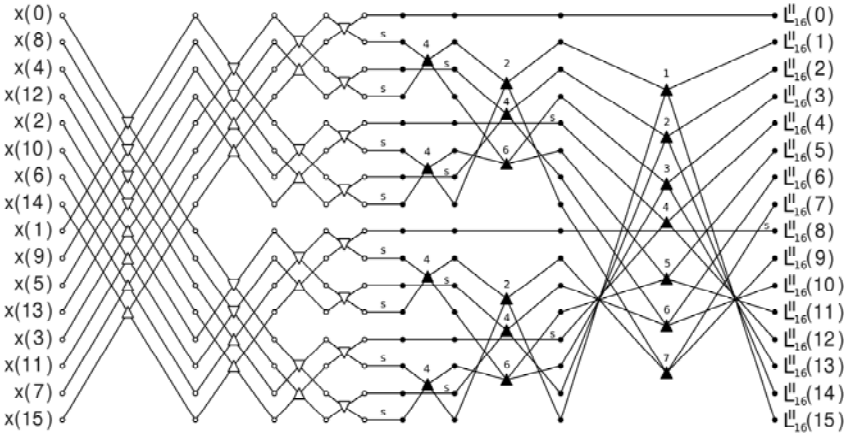
Fig. 1 The structure of the network for cosine transform computation ($N = 16$)
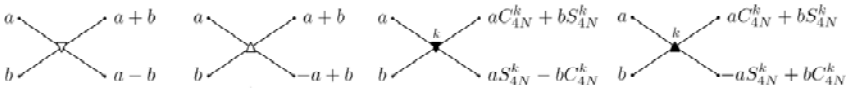
Fig. 2 The basic operations of the network
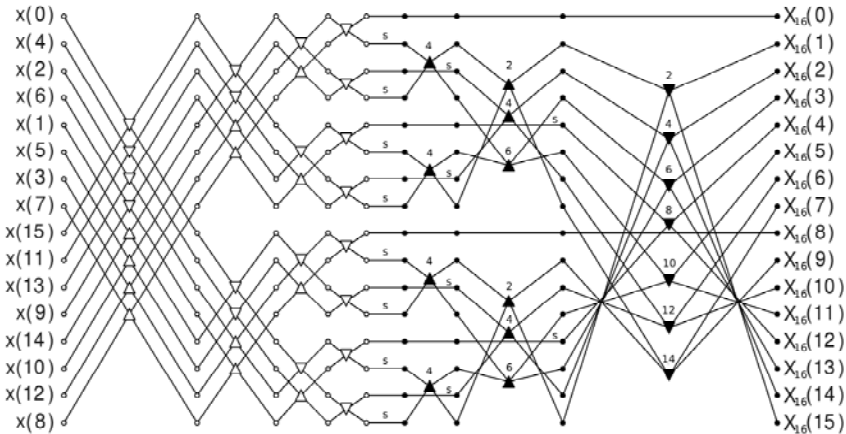
Fig. 3 The structure of the network for Fourier transform computation ($N = 16$)

where $n = 0, 1, \ldots, N/2 - 1$. Secondly, the type of the basic operations in the last layer must be changed (Figs. 2 and 3).

The relation between the outputs of the network $X_N(n)$ and the complex values of Fourier transform $F_N(n) = DFT_N\{x(k)\}$ is given as:

$$X_N(n) = \text{Re}\{F_N(n)\}, \tag{3}$$
$$X_N(N/2+n) = \text{Im}\{F_N(N/2-n)\},$$

where $n = 1, 2, \ldots, N/2 - 1$, and:

$$X_N(0) = \text{Re}\{F_N(0)\}, \tag{4}$$
$$X_N(N/2) = \text{Re}\{F_N(N/2)\}.$$

2.2   The Algorithm with Tangent Multipliers

The fast cosine transform, type II may be computed by the algorithm with tangent multipliers [2] which enables to reduce the number of coefficients (i.e., the number of neural weights) from two to one per each basic operation of the second stage. In this case the structure in Fig. 1 remains the same but the operations in its second stage should be replaced by those shown in Fig. 4a.
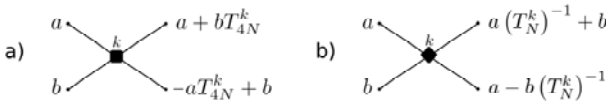
Fig. 4 The basic operations for the algorithm with tangent multipliers

Also, the coefficients s are not used. Instead, every output of the network is multiplied by a coefficient U_N^k, where k is the number of the output, defined recursively as:

$$U_N^0 = 1, \; U_4^1 = \sqrt{2}/2 \times C_{16}^1, \; U_K^k = U_{K/2}^k C_{4K}^k,$$
$$U_K^{K/2-k} = U_{K/2}^k C_{4K}^{K/2-k}, \; U_N^{N/2} = \sqrt{2}/2,$$
$$U_K^{K/4} = \sqrt{2}/2 \times C_{4K}^{K/4}, \; K = 8, 16, \ldots, N, \tag{5}$$
$$U_N^{N/2+k} = U_N^{N/2-k}, \; U_N^{N-k} = U_N^k,$$
$$k = 1, 2, \ldots, K/4 - 1.$$

The neural network for Fourier transform computation based on the algorithm with tangent multipliers is derived as in Sect. 2.1, i.e., the input permutation (2) is used and the types of basic operations in the last layer are changed (Fig. 5).

The coefficients in the output layer, denoted in Fig. 5 as $\hat{U}_N^k$ need additional attention. As the operation type in the last BOON-based layer is changed, the last recursion level in (5) should be also modified: the coefficients C_{4N}^k must be replaced by S_{2N}^k. Taking into account the identity:

$$\sin\left(\frac{2\pi k}{2N}\right) \bigg/ \cos\left(\frac{2\pi k}{4N}\right) = 2\sin\left(\frac{2\pi k}{4N}\right), \tag{6}$$
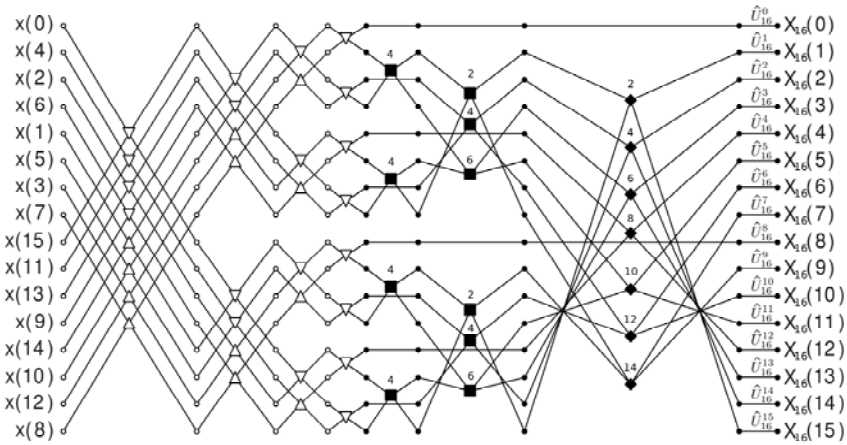
Fig. 5 Network based on Fourier transform with tangent multipliers ($N = 16$)

we find that $\hat{U}_N^k = U_N^k \times V_N^k$, where U_N^k are defined by (5), and the coefficients V_N^k for $k = 1, 2, \ldots, N/2 - 1$ are defined as:

$$V_N^0 = 1, \; V_N^{N/2} = \sqrt{2}, \; V_N^k = V_N^{N-k} = 2 \times \sin\left(\frac{2\pi k}{4N}\right).$$

$$(7)$$

2.3 Fourier Amplitude Spectrum

In order to make the network able to learn the amplitude spectrum, a special output layer, containing elements computing absolute values of complex numbers, should be attached to either of the graphs in Figs. 3, 5.
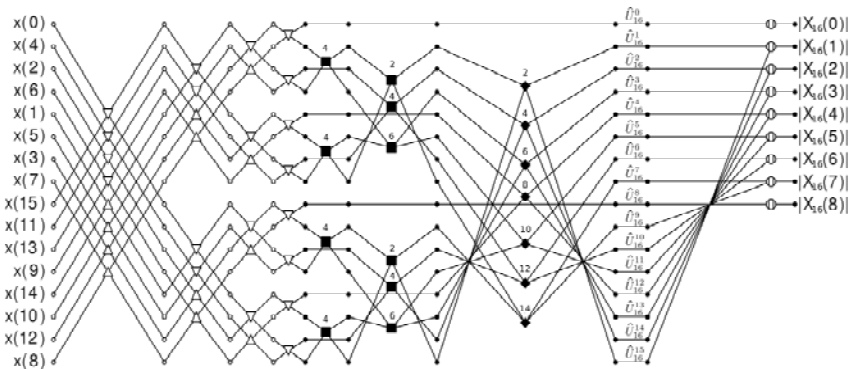
Fig. 6 Neural network for Fourier amplitude spectrum computation
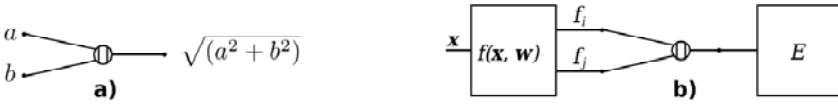
Fig. 7 (**a**) Element computing absolute value. (**b**) Computational scheme

The example based on the algorithm with tangent multipliers is presented in Fig. 6 and the additional elements are described in Fig. 7a.

Training of the neural network containing elements from Fig. 7a is based on the following observations: the elements of this type do not have weights, therefore the size of the gradient vector remains the same; the value of error signal passing through this element during backpropagation must be adequately modified. In order to compute the modified value of error signal let us consider the scheme presented in Fig. 7b. It contains two basic computational blocks: the block corresponding to our neural network and the block representing the error function. Both blocks are connected via the element computing the absolute value. Taking into account the following:

$$u(w) = \sqrt{\left(f_i^2(w) + f_j^2(w)\right)},$$ (8)

the value of error signal with respect to the weights vector is given as:

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial u}\frac{\partial u}{\partial w} = \frac{\partial E}{\partial u}\frac{1}{2u}\left(2f_i\frac{\partial f_i}{\partial w} + 2f_j\frac{\partial f_j}{\partial w}\right) = \frac{\partial E}{\partial u}\frac{f_i}{u}\frac{\partial f_i}{\partial w} + \frac{\partial E}{\partial u}\frac{f_j}{u}\frac{\partial f_j}{\partial w}.$$ (9)

Hence, the target modification of error values during backpropagation amounts to multiplying them by the quotient input/output of the elements under consideration.

3 Classification Experiments

The neural networks presented in Figs. 3, 5 were implemented and their ability to learn Fourier transform was successfully verified. It was also confirmed that the structure in Fig. 6 may easily learn Fourier amplitude spectrum.

For the classification purposes, the FONN network in Fig. 6 was chosen and one additional output neural layer was added to it. Assuming that we have K different classes, this additional layer contains K neurons, each corresponding to a single class. Every output neuron is biased, it has unipolar sigmoidal activation function and $N/2+1$ inputs, where N is the size of input vectors to classify. For a comparison, a standard multilayer perceptron (MLP: N-H-K) with one hidden layer of size H is used [6, 8, 11]. All hidden and output neurons are biased, and they have unipolar, sigmoidal activation function.

The dataset consists of random vectors of size $N = 256$ divided into $K = 8$ classes. The amplitude and phase spectra of each vector are randomized separately in order to guarantee that: all the vectors from a single class have the same amplitude spectrum;
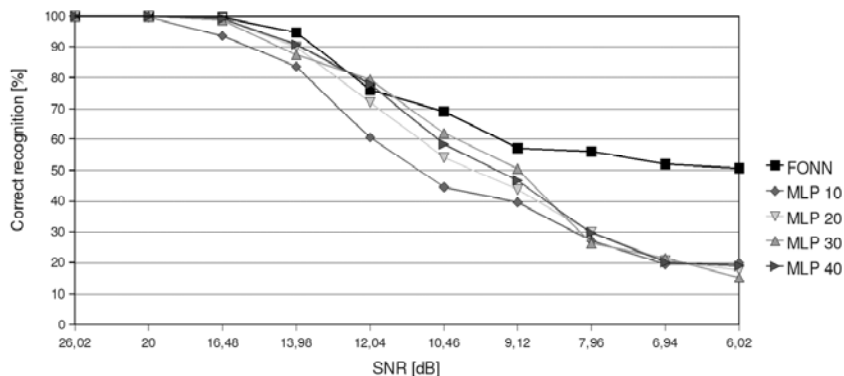
Fig. 8 Results of the FONN and the MLP with $H = 10, \ldots, 40$ hidden neurons

all the vectors having the same amplitude spectrum belong to either of two classes, i.e. there are only 4 different amplitude spectra in the whole dataset, each specific to a pair of classes. In this way, any classification method based on amplitude spectrum analysis alone would lead to confusing the classes from the same pair, yielding the best result of 50%. Considering also the phase spectrum, which was different in each class, is therefore necessary for successful classification.

Every class in the training dataset contains 10 vectors varied by their phase spectra and a noise component added to their (otherwise identical) amplitude spectra. Every class in the testing dataset contains 40 vectors with noise component added to both phase and amplitude spectra. Several values of signal-to-noise ratio (SNR) were used for the testing set construction. For each SNR value the training of both FONN and MLP networks was repeated 20 times and the averaged classification results have been presented in Fig. 8. Classification on a separate validation dataset was used to determine the optimal network performance in all the cases. In case of the FONN classifier only the output layer weights were randomized; the FONN part was always initially fixed (although later also adapted) to compute Fourier transform. It should be stressed that for every SNR value the whole dataset was identical for all the tested networks. Also the gradient optimization method and all other simulation parameters were the same.

It may be easily seen that the FONN-based classifier performs better in the presence of noise, proving its good generalization properties. The relatively weak results of the MLP may be a consequence of the noised phase spectrum in the testing set. The amplitude spectrum-based FONN seems to be more immune to phase distortions while, on the other hand, it can extract enough relevant phase information to distinguish between the classes having the same amplitude spectrum. It is also worth noting that the presented FONN classifier has 2319 weights to adapt (the FONN part: 1279, the output layer: 1040), while the considered MLPs contain from 2658 (MLP 10) to 10 608 (MLP 40) weights. Obviously, 10 hidden neurons seems suboptimal for this dataset.

4 Conclusion

Fast orthogonal neural network capable of learning the Fourier transform and its amplitude spectrum has been presented. The network has been used for a classifier construction, which proved its usefulness in terms of noise resistance and generalization properties in comparison to a multilayer perceptron. The computational complexity expressed in the number of weights subjected to adaptation is also much lower, which results from the sparse connection scheme of the proposed neural network.

References

1. Derrode, S., Ghorbel, F.: Robust and efficient Fourier and Mellin transform approximations for gray-level image reconstruction and complete invariant description. Computer Vision and Image Understanding 83, 57–78 (2001)
2. Jacymirski, M.: Fast homogeneous algorithms of cosine transforms, type II and III with tangent multipliers. Automatics 7, 727–741 (2003) (in Polish)
3. Jacymirski, M., Szczepaniak, P.S.: Neural realization of fast linear filters. In: Proceedings of the 4th EURASIP - IEEE Region 8 International Symposium on Video/Image Processing and Multimedia Communications, pp. 153–157 (2002)
4. Milanese, R., Cherbuliez, M.: A rotation-, translation-, and scale-invariant approach to content-based image retrieval. Journal of Vision Communication and Image Representation 10, 186–196 (1999)
5. Oppenheim, A.V., Hayes, M.H., Lim, J.S.: Iterative procedures for signal reconstruction from Fourier transform phase. Optical Engineering 21, 122–127 (1982)
6. Osowski, S.: Neural networks for information processing. Oficyna Wydawnicza Politechniki Warszawskiej, Warsaw (2000) (in Polish)
7. Rao, K.R., Yip, P.: Discrete cosine transform. Academic Press, San Diego (1990)
8. Rutkowski, L.: Methods and Techniques of Artificial Intelligence. Państwowe Wydawnictwa Naukowe, Warsaw (2005) (in Polish)
9. Stasiak, B., Yatsymirskyy, M.: Fast orthogonal neural networks. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 142–149. Springer, Heidelberg (2006)
10. Szczepaniak, P.S.: Intelligent computations, fast transforms and classifiers. EXIT Academic Publishing House, Warsaw (2004) (in Polish)
11. Tadeusiewicz, R., Gąciarz, T., Borowik, B., Leper, B.: Discovering neural network proprieties by means of C# programs. Polska Akademia Umiejętności, Cracow (2007) (in Polish)

Relative Reduct-Based Selection of Features for ANN Classifier

Urszula Stańczyk

Abstract. Artificial neural networks hold the established position of efficient classifiers used in decision support systems, yet to be efficient an ANN-based classifier requires careful selection of features. The excessive number of conditional attributes is not a guarantee of high classification accuracy, it means gathering and storing more data, and increasing the size of the network. Also the implementation of the trained network can become complex and the classification process takes more time. This line of reasoning leads to conclusion that the number of features should be reduced as far as possible without diminishing the power of the classifier. The paper presents investigations on attribute reduction process performed by exploiting the concept of reducts from the rough set theory and employed within stylometric analysis of literary texts that belongs with automatic categorisation tasks.

Keywords: ANN, rough sets, classifier, feature selection, stylometry.

1 Introduction

The process of decision making requires the support of an efficient classifier. The classifier can be an example of a connectionist approach with distributed processing and representation such as an artificial neural network, or a rule-based solution such as a decision algorithm constructed from rules obtained with rough set methodology. Both attitudes are used with success also in cases when accessible knowledge is incomplete or uncertain, yet to be efficient any classifier needs input data corresponding to significant features of objects to be classified and the process of selecting conditional attributes is not trivial as they have to be representative for classes of objects.

Urszula Stańczyk
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice
e-mail: urszula.stanczyk@polsl.pl

It may seem that when in doubt it is always better to operate on higher rather than on lower number of features. The former causes the information to be repetitive or excessive which may slow the processing down but beside that is rather harmless, while the latter may result in data insufficient for correct classification. Furthermore, both connectionist and rule-based approaches possess inherent mechanisms of finding the key elements within the input data by establishing significance of features. In artificial neural networks this is obtained by the training procedure that computes weights associated with interconnections, while in rough set methodology there are determined relative reducts which are such subsets of conditional attributes that preserve the classification properties of the decision table that describes objects.

However, the high number of features is not a guarantee of high classification accuracy. More attributes cause the necessity of gathering potentially large amounts of data even if some cumbersome computations are involved, and then storing and accessing this data. With many inputs the classifier is usually more complex thus its implementation is more expensive and, even with putting aside the time needed for construction, the classification processing time can cease to be negligible for intended application. All these arguments bring the conclusion that the number of descriptive features should be reduced as far as possible without undermining the power of the classifier.

The paper describes the process of constructing an ANN-based classifier applied in textual analysis with selection of features incorporating elements of rough set methodology [10]. The proposed hybrid approach enables obtaining the classifier with at least the same average classification accuracy than the one employing the original set of textual descriptors working as conditional attributes even when the reduction of attributes reaches almost 50%.

The presented area of application for the designed classifier of textual analysis enables to characterise, compare and recognise authors of texts. This can be exploited within automatic categorisation of texts, which in turn belongs with the area of information retrieval.

2   Stylometric Analysis

Automatic categorisation of texts can be performed basing on the content (recognition of certain key words or phrases) but also on the source (recognition of the author). In the latter case the analysis can employ stylometric tasks of author characterisation, comparison and attribution, historically used for proving or disproving the authenticity of documents or settling the questions of dubious authorship.

The birth of modern stylometry is usually associated with the late XIXth century and works of Mendenhall who was the first to propose using quantitative as opposed to qualitative features of texts. Such numerical approach has been difficult to apply before the development of contemporary computers, which by their high computational power make possible exploiting statistical-oriented analysis and techniques from the artificial intelligence domain.

Contrary to historic textual analysis that could base only on striking features of texts for their characterisation and comparison, contemporary techniques exploit even common parts of speech which, since used rather subconsciously, are less likely to be imitated and thus allow to recognise individual writing styles. Textual descriptors enabling to settle the question of authorship form so-called author invariant.

There is no consensus as to which features of a text constitute authorial invariant. Some analysts propose lexical, while others vote for syntactic, structural or content specific descriptors. Lexical characteristics base on such statistics as average number of characters per word or sentence, or words per sentence, usage frequency for single characters or words, especially function words. Syntactic descriptors reflect the structure of created sentences as given by punctuation marks. Structural features define organisation of text into constructing elements such as headings, paragraphs, signatures, and content-specific markers find words of special meaning in the given context [7].

The choice of textual descriptors is one of crucial issues within stylometric analysis while the other is the selection of the processing technique applied to the task. At present these usually belong either to the statistical approaches (such as Principal Component Analysis, Linear Discriminant Analysis, etc.) or artificial intelligence area (Genetic Algorithms, Support Vector Machines, etc.). Within the latter group there are included artificial neural networks [4] and rough sets [11], which combined together as a hybrid solution [10] were used within experiments described in the paper.

3 ANN-Based Classification

Artificial neural networks have emerged as generalisations of mathematical models invented to describe nervous systems existing in biological organisms and constitute connectionist approach to classification problems. Neural networks have proven themselves as efficient classifiers in cases when observation of subtle relations among attributes is needed, providing that input data is representative enough to satisfy generalisation requirement.

Neural network specification comprises definitions of the number of neurons and their organisation, neuron activation functions and offsets, connections between neurons with their weights, and the network learning (or training) rule. The most popularly used type of neural networks employed in pattern classification tasks is the feedforward network constructed from layers, possessing unidirectional weighted connections between neurons. The most commonly used example of such network is Multilayer Perceptron MLP, with the sigmoid activation function

$$y(n) = \frac{1}{1 + e^{-\beta n}}, \qquad n = \mathbf{W} \cdot \mathbf{X} = \mathbf{W}^T \mathbf{X} = \sum_{j=0}^{J} w_j x_j, \tag{1}$$

where n (net) is a scalar product of the weight $\mathbf{W}$ and input vectors $\mathbf{X}$, with $j = 0$ reserved for offset t, by setting $x_0 = 1$ and $w_0 = -t$.

The popular training rule employed is the classical backpropagation algorithm which modifies the vector of all weights $\mathbf{W}$ accordingly to the descent direction of the gradient

$$\Delta \mathbf{W} = -\eta \nabla e(\mathbf{W}) = -\eta \nabla \left(\frac{1}{2} \sum_{m=1}^{M} \sum_{i=1}^{I} (d_i^m - y_i^m(\mathbf{W}))^2 \right) \tag{2}$$

(with η being the learning rate) of the error occurring on the output of the network that is a sum of errors for all M training facts on all output neurons, each defined by the difference between the expected outcome d_i^m and the one generated by the network $y_i^m(\mathbf{W})$.

For classification purposes as many distinct features are defined for objects which are analysed that many input nodes are required. The number of network outputs typically reflects the number of classification classes. The number of hidden neurons is essential to classification ability and accuracy. When it is unnecessarily high the network easily learns but poorly generalises on new data, yet when there are too few hidden neurons the network may never learn the relationships amongst the input data. There is no precise indicator how many neurons should be used in the construction of a network yet there are several indicators and one of them is that the number of hidden neurons should be equal to half a sum of inputs and outputs. This rule was followed within the performed research.

4   Input Data

As the input data for experiments there were taken literary texts of two famous writers from XIXth century, Henry James and Thomas Hardy. To create the training set there were taken 30 samples from three novels for each writer resulting in the total of 180 samples. In the same manner for the testing set there were taken 10 samples from other three novels accumulating to 60 testing samples.

For these samples there was arbitrarily proposed a set of features that allows for classification with satisfactory average accuracy and this initial set of textual descriptors consisted of 25 attributes corresponding to frequencies of usage for textual markers coming from two groups:

- lexical (17 function words): but, and, not, in, with, on, at, of, this, as, that, what, from, by, for, to, if,
- syntactic (8 punctuation marks): a fullstop, a comma, a question mark, an exclamation mark, a semicolon, a colon, a bracket, a hyphen.

The frequencies of defined descriptors were calculated with help of the dedicated software and the software used for simulation of ANN was California Scientific Brainmaker. There were two classes to be distinguished as there were two authors to be recognised thus there were two outputs from the network. Since the number

of outputs was significantly lower than inputs, it was disregarded when finding the number of hidden neurons. The network constructed in all experiments had one hidden layer with the number of neurons equal to $\lceil N/2 \rceil$, N being the number of inputs to the network.

Within the training procedure one of crucial factors is the initiation of weights associated with interconnections as depending on it the training can last much longer, the network can end up in oscillations around local minima and even when it finally converges the classification accuracy of the trained network can significantly vary. To take all these issues into account instead of a single training procedure there was applied multi-starting approach – for each network configuration the training was performed 100 times with storing accuracies of classification and basing on such series there was obtained the worst network with the lowest accuracy, the best with the highest, and finally the average network. Results for the original network are presented in Fig. 1. The worst network gave correct classification for 78.33%, the best 88.33% and the average 83.33%.
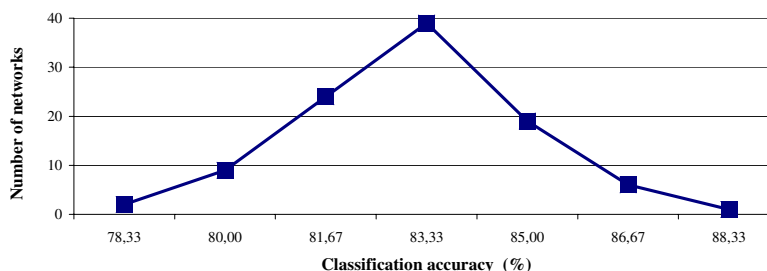
Fig. 1 Original network classification accuracy

Stylometric analysis cannot unambiguously answer the question which textual features from the original set of 25 attributes are important from the point of view of author recognition and which could be disregarded without worsening the classification accuracy and that is where the rough set based approach comes in handy by employing the concept of reducts in feature selection.

5 Rough Set-Based Feature Reduction

Generally feature selection procedure can be performed with variety of techniques [3], even neural networks themselves. In the described experiments to arrive at a subset of conditional attributes with the lowest cardinality that still preserves the classification accuracy there were applied elements of rough set theory, which by themselves can be successfuly employed not only at the stage of feature selection [8] but as the processing technique for stylometric tasks [11].

The first step in the rough set-based approach, invented by Zdzisław Pawlak [6] in the early 1980s, is defining a decision table that contains the whole knowledge

about the Universe U. Columns of the decision table are defined by conditional C and decision D attributes while rows X specify values of these attributes ($A = C \bigcup D$) for each object of the Universe, which allows to partition U into equivalence classes $[x]_A$ with indiscernibility relation [6]. The indiscernibility relation and resulting from it equivalence classes enable to describe sets of objects by their lower $\underline{A}X$ and upper approximations $\overline{A}X$.

Information held by a decision table is often excessive in such sense that either not all attributes or not all their values are needed for correct classification of 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 and value reducts [5]. These tools result in finding decision rules that form decision algorithms, which are examples of rule-based solutions to classification tasks [12].

However, the classical rough set approach deals only with discrete data and the frequencies studied within the experiments, as can be seen in the original decision table created (Table 1), are continuous values. Thus either there had to be applied some discretisation strategy by defining a discretisation factor [1], or modified indiscernibility relation applicable for continuous attributes [2], or instead of using classical rough set approach (CRSA) there could be employed DRSA: dominance-based rough set approach [9] integrating dominance relation with rough approximation, used in multicriteria decision support. The last solution shows promise, yet from the stylometric point of view it cannot be definitely stated whether attributes are *preference* ordered (as DRSA assumes).

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
5	0.0057	0.0254	0.0078	0.016		0.0006	0.0006	0.0221	hardy
⋮									
176	0.0082	0.0177	0.0043	0.0146		0.0019	0	0.0494	james
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

In the research to simplify the task there was used discretisation by median values for all attributes. The resulting binary decision table was next checked with respect to its consistency measure $\gamma_C(D^*)$, which answers the question whether the table is

deterministic. All decision rules provided by rows of DT were compared, one by one against all others, to find at least two with the same values of conditional attributes but different for decision attributes *D*, because in such case the table would not be deterministic. Fortunately the table turned out to be deterministic which enabled to proceed with the process of finding relative reducts and the relative core.

In case of a binary decision table the process of finding relative reducts is the same as that of parallel reduction of arguments for a Boolean function hence there was employed Pandor software implementing procedures of Esspresso system for minimisation of logic functions. It returned 96 distinct subsets of input variables defining the output function, all consisting of 8 attributes, corresponding to relative reducts. Their intersection, which is the relative core of the decision table, turned out to be empty, which means that no attribute was present in all reducts. On the other hand the union of all reducts was equal to the whole initial set of attributes indicating that no feature from these studied could be disregarded without further analysis.

6 Obtained Results

Since neither stylometry nor rough set approach could precisely answer the question which textual markers should be selected as features for classification with the highest accuracy ratio, several subsets were tried and compared with the original results.

Firstly as feature extractors there where selected three reducts:

- random reduct R32: and, not, from, of, in, semicolon, colon, hyphen,
- purely lexical R4: but, and, not, with, at, of, what, from,
- half-lexical and half-syntactic R91: and, not, that, from, comma, fullstop, bracket, hyphen.

ANN-based classifiers with these three sets of inputs had just 4 hidden neurons and results of their training are given by Table 2.

It is immediately apparent from the table that all three classifiers perform worse then the original one, with the best results obtained for the classifier basing on the lexical reduct and the worst for maximal number of syntactic markers. Thus direct application of a relative reduct within selection of attributes gives unsatisfactory results, yet the conclusion that reducts are of no use for ANN-based classifiers would be much too premature.

Table 2 Classification accuracy for networks with inputs selected by reducts

	R32	R4	R91
Worst case	65.00%	73.33%	66.67%
Average case	76.67%	80.00%	75.00%
Best case	86.67%	85.00%	81.67%

As can be seen even for just the three reducts selected for tests, some of attributes occur in more than one relative reduct. Hence instead of direct usage of a reduct as feature selector there were first calculated numbers of occurrences within reducts for all attributes. Then they were ordered with respect to calculated numbers and this frequency was treated as the significance indicator for each attribute. The results are provided by Table 3. The left-most table lists 8 the least frequently used descriptors while the right-most gives those 8 most frequently employed.

Table 3 Reduct-based attribute occurrence indicators

Attribute	Occur. ind.	Attribute	Occur. ind.	Attribute	Occur. ind.
exclam. mark	6	as	14	hyphen	41
on	7	semicolon	15	to	46
in	7	quest. mark	16	at	47
if	8	that	16	bracket	59
comma	8	colon	18	but	73
what	10	of	19	not	81
for	12	by	21	from	85
this	12	with	25	and	87
		fullstop	35		

These occurrence indicators were next exploited for creation of networks with reducing the set of inputs by removing from it either 5 least frequently used attributes (exclamation mark, on, in, if, comma) or 4 most frequently used (and, from, not, but), with results shown in Fig. 2. The results for both types of networks gave average correct classification ratio of 83.33%, which is the same as in the case of the original network configuration of 25 inputs, with the worst case slightly lower than before. Reduction of the set of inputs is by 16–20%.

Next there were simulated networks with 17 most frequently and 17 least frequently employed inputs. The results are provided by Fig. 3. Removing from the set of inputs either 8 (reduction by 32%) the most or the least frequently occurring attributes results in both cases in increase of average classification ratio, however,
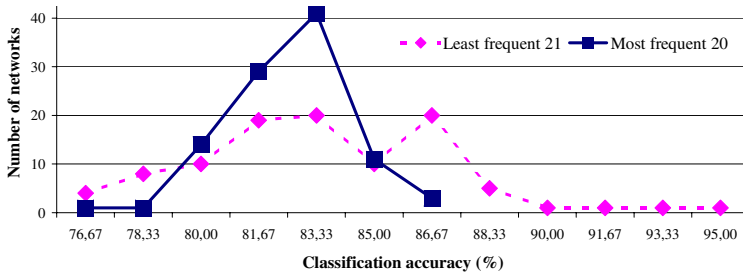

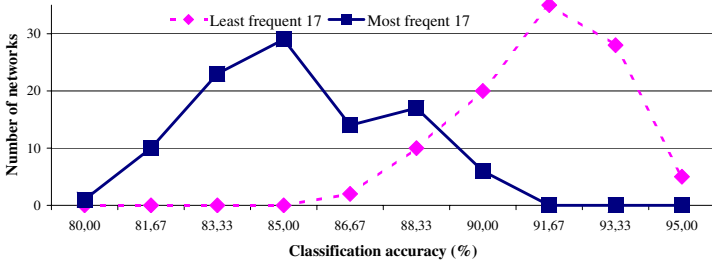**Fig. 2** Network performance for 16-20% fewer inputs

Fig. 3 Network performance for 32% fewer attributes
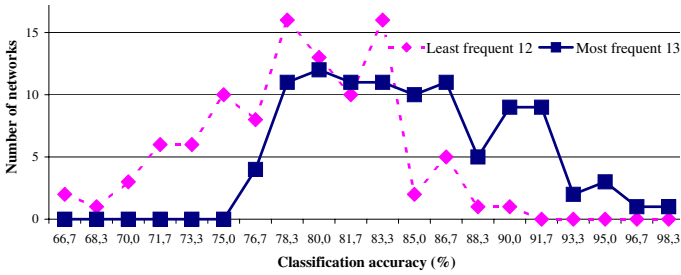
Fig. 4 Network performance for 48–52% fewer features

the results are much more improved in case of leaving these attributes that are less often used in reducts. The average reaches then 91.67% not to mention the best case of 95%.

Final simulation of the network was done for 13 (52% of the original set of inputs) most and 12 (48% of inputs) least frequently occurring inputs, the two subsets complementing each other with respect to the whole set of inputs. As the result (Fig. 4) in the first case there is still the average 85% correct classification of testing samples and the network also gives the best result of all networks with 98.33% recognition, while in the latter it falls down to 78.33%.

Further reduction of attributes would get too close to obtaining networks based on reduct-size sets of inputs, which, as already shown, does not provide enough information for the ANN-based classifier to maintain its correct recognition ratio.

7 Conclusions

The paper addresses the hybrid approach to classification by applying the concept of relative reducts from the classical rough set approach at the stage of feature selection for ANN-based classifier. As individual reducts do not provide sufficient knowledge about the Universe to maintain the correct recognition ratio of the classifier when employed in selection of inputs, instead attribute occurrences within all reducts are

calculated and the set of conditional attributes is reduced by those most and least frequently exploited to form relative reducts. Basing on such approach there are constructed networks with at least the same, if not better, recognition ratios even when the set of inputs is reduced by as much as 48%.

References

1. Cyran, K.A., Mrózek, A.: Rough sets in hybrid methods for pattern recognition. International Journal of Intelligent Systems 16, 149–168 (2001)
2. Cyran, K.A., Stańczyk, 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)
3. Doumpos, M., Salappa, A.: Feature selection algorithms in classification problems: an experimental evaluation. WSEAS Transactions on Information Science & Applications 2(2), 77–82 (2005)
4. Matthews, R.A.J., Merriam, T.V.N.: Distinguishing literary styles using neural networks. In: Fiesler, E., Beale, R. (eds.) Handbook of neural computation, pp. G8.1.1–6. Oxford University Press, Oxford (1997)
5. Moshkow, M.J., 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)
6. Pawlak, Z.: Rough set rudiments. Tech. rep., Institute of Computer Science Report, Warsaw University of Technology, Warsaw, Poland (1996)
7. Peng, R.D., Hengartner, H.: Quantitative analysis of literary styles. The American Statistician 56(3), 15–38 (2002)
8. 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 (2006)
9. 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)
10. Smolinski, T.G., Chenoweth, D.L., Zurada, J.M.: Application of rough sets and neural networks to forecasting university facility and administrative cost recovery. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 538–543. Springer, Heidelberg (2004)
11. Stańczyk, U., Cyran, K.A.: 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)
12. 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 (2006)

Enhanced Ontology Based Profile Comparison Mechanism for Better Recommendation

Revoti Prasad Bora, Chhavi Bhandari, and Anish Mehta

Abstract. Recommender Systems, based on Collaborative filtering techniques, recommend contents based on the opinions of other users, which makes it imperative to aggregate similar users as accurately as possible. Most of the Collaborative Filtering based Recommender Systems use profile vector directly for similarity evaluation. A few recent researches have explored the possibility of using Ontology for evaluating similarity but all the aspects of Ontology have not yet been exploited. In this paper we propose an 'Enhanced Ontology based Profile Comparison' mechanism for better similarity evaluation. This mechanism expands the profile vector components in one user profile on the basis of explicit semantic relations. The expanded results are then compared with the other user's expanded profile vector components using a predefined Relationship Weightage Scheme. This facilitates better recommender aggregation and hence improves the quality of recommendations. The proposed mechanism can be used for Content Based Filtering as well.

Keywords: ontology, collaborative filtering, recommender systems, content based filtering, relationship weightage scheme, profile vector, spreading activation method.

1 Introduction

1.1 Recommender Systems

The Internet is growing at a tremendous pace and hence the number of contents in it is also growing exponentially. Due to the overwhelming amount of contents available, people usually find it difficult to find contents of their choice. Although search engines ease the job of searching a content by filtering pages that match explicit

Revoti Prasad Bora · Chhavi Bhandari · Anish Mehta
Samsung India Electronics Limited
e-mail:{revoti.bora, chhavi.b, anish.mehta}@samsung.com

queries but most of the times people are unable to provide the exact query. More-over search engines do not take into the account the taste of the User for searching contents. To solve these problems Recommender Systems came into existence. A Recommender Systems can be defined as a specific type of information filtering (IF) technique that attempts to present contents those are likely of interest to the user. Typically a Recommender System dynamically maps the behavior of a user into a 'Profile', compares it with some reference characteristics and tries to calculate the user's liking for that particular content. The characteristic may be extracted from the information of the item, i.e., Content Based technique or other Users, i.e., Collaborative Filtering. In this paper we will consider only the latter but this scheme is equally applicable for Content Based filtering. [3] identifies two basic subclasses of algorithms for CF: Model Based and Memory Based. Model based algorithms build a model of the data and use the model for prediction, whereas memory based algorithms do not build a model, but use the data directly for prediction. Memory based algorithms relies on similar Users' Rating for a content in making predictions. Hence the performance of CF is directly dependent on finding similar Users. In order to calculate similarity, it is a common practice to represent the user profiles in the so-called bag-of-words (BOW) format. A BOW format is a set of weighted terms that best describe the entity so that the similarity between two entities can now be computed using some standard techniques like Cosine, Pearson Coefficient, Jaccard Coefficient etc.

1.2   Ontology

Ontology is a conceptualization of a domain into a human-understandable, machine-readable format consisting of entities, attributes, relationships, and axioms [5]. Ontology can provide a rich conceptualization of the domain of any dataset (e.g., Movies tags) representing the main concepts and relationships among the entities. We use the term ontology to refer to the classification structure and instances within a knowledge base. The latest Recommender Systems consider domain knowledge to improve the accuracy and effectiveness of recommendations. An item can be expressed simply by a feature vector where each dimension represents an item feature. It is assumed that item features are dependent on user perspective and therefore understanding item features and their relationships can help us gain more information about the user's preferences. With Ontology, item features and relationships among them can be expressed in machine readable format and therefore can be used to compute more accurate recommendations.

1.3   Problem and Our Contribution

The similarity calculations, using the BOW format, so far takes into account only the 'Sameness Quotient' between the entities in question, i.e., the technique tries to determine the words (or tags) that are common to the BOWs of both the users. Users with more matching tags are considered more similar than users with lesser

matching tags. Spreading Activation Methods (incorporating concepts of ontology) have been used to improve similarity calculations but their approach is also limited to finding more and more similar tags, for instance synonyms. However no weightage is given, to any polarities that the two profiles might have which in turn has the potential to make recommendations less accurate. Thus we intend to explore this aspect in greater detail.

In this paper, we extend the notion of ontology to improve the recommendation systems. Our approach uses ontology not just to find similar tags (weight = 1) but also unrelated tags (weight = 0) and dissimilar or opposite tags (weight = -1) in the two profiles. This approach will help the Recommender System to better understand not only the similarity between the users but also the dissimilarity between them, thus improving the recommender list and consequently the recommendations.

This paper is organized as follows. In the next section we provide a brief background to the currently used techniques. In Sect. 3 we describe the proposed. In Sects. 4, 5, and 6, we discuss the assumptions, the experimental results and the future work respectively.

2   Background

The bag-of-words model is a simplifying assumption used in natural language processing and information retrieval. In this model, a text (such as a sentence or a document) is represented as an unordered collection of words, disregarding grammar and even word order. In the BOW format an entity (E) is represented by a set of pairs, denoted as (t_i, w_i) where t_i is a term that describes the entity and $t_i \, \varepsilon$ terms of (E), w_i is the weight or importance of the respective term in describing the entity. The similarity computations are then done directly on the BOWs of different User Profiles.

There are several mathematical formulae to compute similarity between two profiles for example Pearson's correlation, cosine similarity, Jaccard similarity coefficient, Dice's coefficient, Mountford's index of similarity etc. We use cosine-based similarity method to compute similarity and compare the results with our approach. In this method, two items are thought of as two vectors in the m-dimensional userspace. The similarity between them is measured by computing the cosine of the angle between these two vectors. Formally, in the $m \times n$ rating matrix, similarity between items a and b, denoted by $\mathrm{sim}(a,b)$ is given by:

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{|a|\,|b|} \, , \tag{1}$$

where '·' denotes the dot-product of the two vectors.

A lot of research is being done on using the concept of ontology to improve similarity. Semantic similarity measures play an important role in information retrieval and information integration. Both [12] and [1], use a unique approach to extend

the notion of semantic similarity between two entities to consider inherent relationships between concepts/words appearing in their respective BOW representation, through a process called Spreading Activation, the process of including additional related terms to an entity description by referring to Ontology. In the spreading activation stage of [1], a network is activated as a set of nodes representing product descriptions or phrases (i.e., indexing terms of the product descriptions) with relationships between the nodes specified by labeled links. The 2-level node activation process starts in one direction placing an initial activation weight on the hot phrase node and then proceeds to spread the activation through the network one link at-a-time, activating product description and phrase nodes relevant to the hot phrase. In the other direction, the network originating with the hot phrase node activates its synonym nodes that in turn activate their relevant product and phrase nodes. [8] presents an approach to computing semantic similarity that relaxes the requirement of a single Ontology and accounts for differences in the levels of explicitness and formalization of the different ontology specifications. A similarity function determines similar entity classes by using a matching process over synonym sets, semantic neighborhoods, and distinguishing features that are classified into parts, functions, and attributes. In [3], an Ontology concept is represented using a profile with words describing the concept. A propagation technique to enrich the profile with words from neighboring concepts is also presented. Such profiles are subsequently used to determine closeness (using cosine metric) between concepts they represent.

As described above, all the approaches till now concentrate on finding similarity that may not be explicit, between two entities. Thus Ontology is used to find two entities that are similar but are overlooked by traditional calculations due to implicit matching failure. We present a novel notion to use Ontology to find dissimilarities between entities.

3   Our Solution

Similarity between two users can be represented as a function of the two user's profiles as follows:
$$\text{Sim} = f(P_1, P_2),$$
where P_1 and P_2 are the profiles of the two users.

In the Bag of words representation, the usual connotation of this function would imply, $f(P_1, P_2) = P_1 \cap P_2$, where $\cap$ represents the intersection between the two sets of profiles(containing tags). Classically, intersection would result in all those tags which are common to both the user profiles, i.e., it is based on simple string match function. More the number of string matches occur, more similar the two users are.Mathematically, $\cap$ would represent only exact string matches in the classical approach.

Recent research has extended the concept of ontology to this field such that first each tag in a user profile is expanded using ontology and only then the comparisons

occur. This helps in identifying relations which may not be explicit but are implicitly fabricated in real terms. In other words, intersection in this scenario is equivalent to any synonymous match that might exist between the two profiles. Hence, $\cap$ represents StringMatch$(1)*+$SynonymMatch$(1)*$, where $*$ Quantity in bracket refers to the weightage given to the relation.

In our paper, we further extend the notion of Ontology to include implicit dissimilarities between the two profiles. This approach works in two directions simultaneously, i.e., on one hand finding similarities in order to bring two users closer to each other and on the other hand finding dissimilarities in order to pull the two users away from each other. Therefore keeping in mind both likeness and unlikeness between the two users which would result in better recommendations eventually. Mathematically,

$$\cap = \text{StringMatch}(1)*+\text{SynonymMatch}(1)*+\text{AntonymMatch}(-1)*.$$

To take a simplified example, consider three profiles A, B, C with their tags or BOW as follows:

- $A = \{War, Comedy\}$,
- $B = \{Peace, Comedy\}$,
- $C = \{Romantic, Comedy\}$.

Now using the present techniques available A, B, and C each will have the same 'Similarity' values w.r.t. each other because all of the users have one overlapping tag and one non-overlapping tag, i.e.,

$$\text{Sim}(B,C) = \text{Sim}(B,A).$$

But going by human perception, it is clear that $B\&C$ are more likely to share better recommendations than $B\&A$ (since $B\&A$ have clashing interests in some areas), i.e.,

$$\text{Sim}(B,C) > \text{Sim}(B,A).$$

This fundamental aspect has been taken into account in our approach.

4 Assumptions

It is assumed that well developed Ontology exists in every domain where a recommended system may be used. Availability of exhaustive ontology on the required subject is a prerequisite. Though presently there are still domains on which Ontology are unavailable, it is assumed that in near future they will be available considering a lot of study and research is going on in this area and the domains are expanding with every passing day. In order to evaluate our experiment, we built an in house basic ontology that was minimalistic in nature and served the purpose.

5 Simulation Results

For Simulation we made our own simulator which had the capability of recognizing similar and dissimilar words and calculate the Cosine Similarity values for 100 pairs of User Profiles randomly extracted from Netflix Dataset. We calculated the Similarity values for both Cosine Method as well as ours and normalize the results to the interval $[0, 1]$. The results of both the schemes match when the profiles are identical but in other cases variations are observed due to the consideration of synonymous and antonymous tags in the proposed scheme.
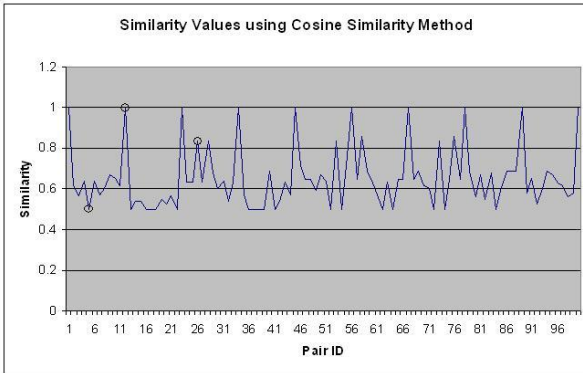

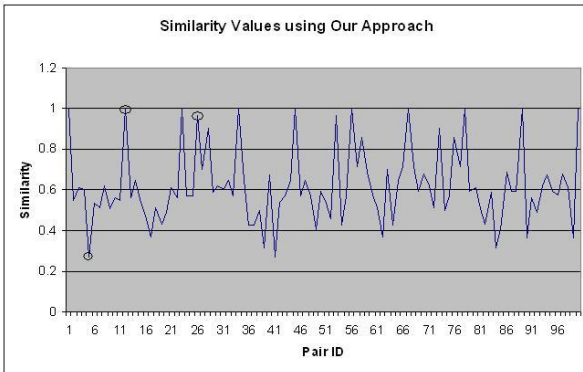**Fig. 1** Similarity evaluated using Cosine Similarity

Fig. 2 Similarity using the proposed scheme

To elucidate further on this point, three different points have been marked on each graph:

1. Pair ID 05 – It can be observed that the similarity value in Fig. 2 is lower than that of Fig. 1. The reason for this difference is that there were more antonymous tags in the two profiles than similar tags.
2. Pair ID 12 – It can be observed that the similarity values in Fig. 2 is equal to that of Fig. 1, which is equal to 1, i.e., User Pairs are exactly similar. This happens when the two user profiles are identical and thus result is 1 irrespective of the method used to calculate similarity.
3. Pair ID 27 – It can be observed that the similarity value in Fig. 2 is higher than that of Fig. 1. The reason for this difference is that there were more identical or synonymous tags in the two profiles than antonymous tags.

6   Conclusion and Future Work

We have presented a novel notion to use Ontology to find not only the similarities but also dissimilarities between entities and use these results to evaluate similarity values. We feel these values are closer to human-perceived similarity. There is a lack of experimental results in the literature for systems using real people. This is a failing of this research field, and it makes direct comparison of systems that address real problems hard. Also, in our present work, values assigned to Ontological relations are limited to integral values. In future, ways can be devised to assign all values in the range $[-1, 1]$ to the Ontological relations in order to get still better user similarity values and thus better recommendations. This scheme can also be extended to Content Based Filtering.

References

1. Aswath, D., D'cunha, J., Ahmed, S.T., Davulcu, H.: Boosting item keyword search with spreading activation. In: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 704–707 (2005)
2. Balabanoic, M., Shoham, Y.: Content-based, collaborative recommendation. Communications of the ACM 40(3), 67–72 (1997)
3. Breese, J., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Uncertainty in Artificial Intelligence, pp. 43–52 (1998)
4. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of the International Joint Conferences on Artificial Intelligence, pp. 1607–1611 (2007)
5. Guarino, N., Giaretta, P.: Ontologies and knowledge bases: towards a terminological clarification. In: Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, pp. 25–32 (1995)
6. Jin, X., Mobasher, B.: Using semantic similarity to enhance item-based collaborative filtering. In: Proceedings of the 2nd IASTED International Conference on Information and Knowledge Sharing (2003)
7. Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J.: Grouplens: Applying collaborative filtering to usenet news. Communications of the ACM 40, 77–87 (1997)

8. Mao, M.: Ontology mapping: An information retrieval and interactive activation network based approach. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 931–935. Springer, Heidelberg (2007)

9. de Melo, G., Siersdorfer, S.: Multilingual text classification using ontologies. In: Proceedings of the 29th European Conference on Information Retrieval, Rome, Italy (2007)

10. Middleton, S.E., Shadbolt, N.R., De Roure, D.C.: Ontological user profiling in recommender systems. ACM Transactions on Information Systems 22, 54–88 (2004)

11. Rodriguez, M.A., Egenhofer, M.J.: Determining semantic similarity among entity classes from different ontologies. IEEE Transactions on Knowledge and Data Engineering 15(2), 442–456 (2003)

12. Thiagarajan, R., Manjunath, G., Stumptner, M.: HP laboratories. Computing semantic similarity using ontologies. In: Proceedings of the International Semantic Web Conference, Karlsruhe, Germany (2008)

Privacy Preserving Classification
for Ordered Attributes

Piotr Andruszkiewicz

Abstract. In privacy preserving classification for centralized data distorted with a randomization-based techniques nominal and continuous attributes are used. Several methods of preserving privacy classification have been proposed in literature, but no method focused on the special case of attributes – ordered attributes. This paper presents a new approach for ordinal and integer attributes. This approach takes the order of attributes into account during distortion and reconstruction procedure. Effectiveness of the new solution has been tested and presented in this paper.

Keywords: privacy preserving data mining, classification, probability distribution reconstruction, ordered attributes.

1 Introduction

There are several methods of privacy preserving classification for centralized data distorted with a randomization-based techniques. They treat attributes with integer values as continuous and do not take an order into account in case of attributes with finite set of ordered values, we will call them ordinal attributes. Thus, distorted integer attributes have to be stored as continuous (not discrete) attributes and we loose information connected with an order of ordinal attributes.

The new approach to ordered attributes presented in this paper allows miner to use integer attributes, distort them and store as integers. For ordinal attributes we take their order into account and distort them according to their natural order.

The experiments show that presented methods give results with high accuracy when we use these methods for integer and ordinal attributes.

Piotr Andruszkiewicz
Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: `P.Andruszkiewicz@ii.pw.edu.pl`

1.1   Related Work

Privacy preserving classification has been extensively discussed recently [2, 6, 1, 5, 9, 10, 8, 3].

Papers [6] and [9] represent the cryptographic approach to privacy preserving. We use a different approach – a randomization-based technique.

Preserving privacy for individual values in distributed data is considered in [10] and [8]. In these works databases are distributed across a number of sites and each site is only willing to share mining process results, but does not want to reveal the source data. Techniques for distributed database require a corresponding part of the true database at each site. Our approach is complementary, because it collects only modified tuples, which are distorted at the client machine.

There are various different approaches to privacy preserving classification, but we mention only the relevant ones.

Agrawal and Srikant [2] proposed how to build a decision tree over (centralized) disturbed data with the randomization-based technique. In this solution they also presented the algorithm (we will call it AS) for probability distribution reconstruction for continuous attributes.

Paper [1] extends the AS algorithm and presents the EM-based reconstruction algorithm, which does not take into account nominal attributes either.

Multivariate Randomized Response technique was presented in [5]. It allows creating a decision tree only for nominal attributes.

The solution showed in [3] differs from those above, because it enables data miner to classify (centralized) perturbed data containing continuous and nominal attributes modified using the randomization-based techniques to preserve privacy on individual level. This approach uses the EM/AS algorithm to reconstruct probability distribution for nominal attributes and the algorithm for assigning reconstructed values to samples for this type of attributes to build a decision tree simultaneously with continuous attributes.

The EQ algorithm for reconstructing probability distribution of nominal attributes was proposed in [4]. The algorithm achieves better results, especially for high privacy level, i.e., low probability of retaining an original value of a nominal attribute.

1.2   Contributions and Organization of This Paper

The solution proposed in this paper enables miner to use ordinal and integer attributes to preserve privacy on individual level in classification with the usage of the randomization-based technique.

The remainder of this paper is organized as follows: In Sect. 2, we present our new approach to ordered attributes in privacy preserving classification. The experimental results are highlighted in Sect. 3. Finally, in Sect. 4, we summarize the conclusions of our study and outline future avenues to explore.

2   Proposal for Ordinal and Integer Attributes

Solutions presented in literature do not take into account a special case of attributes: ordinal and integer which can be treated as continuous (real) attributes with only discrete (integer) values. The difference between these two types of attributes (ordinal attributes and numeric attributes with integer values) is that ordinal attributes have finite number of values contrary to integer attributes with an infinite domain. According to this difference these attributes should be considered separately.

2.1   Ordinal Attributes

The solutions proposed in literature treat ordinal attributes as nominal attributes (without an order), what leads to loss of some information. In privacy preserving data mining it is even more important, because we want to change the values of an attribute using a randomization-based technique to hide individual values of the users' characteristics.

Let the example be the *salary* attribute. We want to preserve privacy on individual level, so we discretise it. We get values: <1000, 1000–2000, 2000–5000, 5000–10000, and >10000, which have the natural order.

Discretisation may not be enough to achieve desired level of privacy, hence we use a randomization-based technique. Standard approach[1] does not take into account the order of the attribute, thus value 1000–2000 may be changed to <1000 and >10000 with the same probability.

It is more probable that there are more people with <1000 value than >10000 and changing value 1000–2000 to the highest value of *salary* will change distribution of the attribute in nonnatural way, what may lead to discloser of information about users' individual values.

To solve this issue we may assign the probabilities of changing the values in the following way: for ordinal attributes we choose the probability of retaining the original value. Then we assign the probabilities of changing the value of the attribute to the (right and left) nearest neighbours. In the next step we choose the probability of changing the value to the (right and left) second nearest neighbours, etc.

To implement this solution we may use **P** matrix proposed in [4].

Definition 1. P *matrix of retaining/changing values of nominal attribute with k possible values is equal to:*

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,k} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{k,1} & a_{k,2} & a_{k,3} & \cdots & a_{k,k} \end{pmatrix},$$

---

[1] i.e., p is the probability of retain original value and the probabilities of changing the value are equal to $\frac{1-p}{k-1}$.

where $a_{r,p} = Pr(v_p \rightarrow v_r)$ *(probability that value v_p will be changed to value v_r).
The sum of the elements in columns is equal to 1.*

The matrix **P** for this solution and the probability of retaining the original value
equal to 0.7 will look as follows:

$$\begin{pmatrix} 0.7 & 0.15 & 0 & 0 & 0.15 \\ 0.15 & 0.7 & 0.15 & 0 & 0 \\ 0 & 0.15 & 0.7 & 0.15 & 0 \\ 0 & 0 & 0.15 & 0.7 & 0.15 \\ 0.15 & 0 & 0 & 0.15 & 0.7 \end{pmatrix}.$$

For 1000–2000 we change the value to <1000 with probability 0.15 and to 2000–
5000 with the same probability. The value cannot be changed to >10000.

With probability 0.15 we change <1000 to >10000 and from >10000 to <1000.
We will call this the *loop effect*.

When we do not want to change the highest value to the lowest and the lowest to
the highest, we may use **P** with eliminated *loop effect*, which is shown below:

$$\begin{pmatrix} 0.7 & 0.15 & 0 & 0 & 0 \\ 0.15 & 0.7 & 0.15 & 0 & 0 \\ 0 & 0.15 & 0.7 & 0.15 & 0 \\ 0 & 0 & 0.15 & 0.7 & 0.15 \\ 0 & 0 & 0 & 0.15 & 0.7 \end{pmatrix}.$$

Columns of **P** should be normalized to 1. This process was omitted for better visu-
alization of the *loop effect* removal.

To reconstruct the original distribution of the attribute distorted according to
proposed method we may use EM/AS [3] and EQ [4] algorithms.

2.2 Integer Attributes

In case of an integer attribute we would like to change its value by adding some
noise, i.e., random values from known distribution, e.g., uniform, normal etc.
Adding (continuous) noise would change integer attribute to continuous one. As
an example we will consider *age* attribute, which has positive integer values.

To address this issue a special cumulative distribution function can be used – we
will call it *staircased* cumulative distribution function. This function increases only
in integer values, thus choosing random values using this function we obtain only
integer values (in special case positive integer values).

Second approach is to draw the noise value from known distribution (e.g., uni-
form) and round the drawn value. The miner should be careful about the method of
rounding.

To reconstruct the original distribution of the attribute distributed in accordance
with described method we may use AS [2] and EM [1] algorithms.

3 Experiments

This section presents the results of the experiments conducted with ordinal and integer attributes.

We use the definition of privacy based on the *differential entropy* [1].

Definition 2. *Privacy inherent in a random variable A is defined as follows:*

$$\Pi(A) = 2^{h(A)},$$

where $h(A)$ is differential entropy *of variable A.*

Definition 3. *Differential entropy of a random variable A is defined as follows:*

$$h(A) = - \int_{\Omega_A} f_A(a) \log_2 f_A(a) \, da,$$

where Ω_A is the domain of variable A.

A random variable A distributed uniformly between 0 and a has privacy equal to a. For general random variable C, $\Pi(C)$ denote the length of the interval, over which a uniformly distributed random variable has the same uncertainty as C.

$n\%$ privacy means that we use the distorting distribution with privacy equal to $n\%$ of the range of values of the distorting distribution, e.g., for 100% privacy and the random variable A with the range of its values equal to 10 we distort it with the distribution with privacy measure equal to 10 (for the uniform distribution we may use the random variable distributed between -5 and 5)[2].

The lack of precision in the reconstruction is called information loss. The information loss is defined as follows [1]:

Definition 4. *Information loss $\mathscr{I}(f_X, \hat{f}_X)$ equals half the expected value of L_1 norm between the original probability distribution f_X and its estimate $\hat{f}_X$.*

$$\mathscr{I}(f_X, \hat{f}_X) = \frac{1}{2} E[\int_{\Omega_X} \mid f_X - \hat{f}_X \mid].$$

Information loss $\mathscr{I}(f_X, \hat{f}_X)$ lies between 0 and 1. 0 means the prefect reconstruction and 1 implies that there is no overlap between original distribution and its estimate.

All statistics were computed as a mean of 100 multiple runs.

3.1 Probability Distribution Reconstruction for Ordered Attributes

Table 1 shows original, distorted and reconstructed probability distribution of modified *employment* attribute (set *credit-g*[3]). We increased the number of values of the

---

[2] For details about privacy measures see [2] and [1].

[3] All sets used in tests can be downloaded from UCI Machine Learning Repository (http://www.datalab.uci.edu/).

attribute, to better show the *loop effect*. According to the results there are no values with index 7 in case of without the *loop effect*, so we take the order of the attribute into account – we do not change the lowest value to the highest.

Table 1 Distortion and reconstruction without and with the *loop effect*. Probability: 0.6 0.2, the EM/AS algorithm

Index	Original	Distorted	Reconstructed	Distorted-loop	Reconstructed-loop
0	62	84	61	80	49
1	172	176	165	195	225
2	339	281	343	261	277
3	174	234	197	218	209
4	253	177	225	189	232
5	0	48	4	49	4
6	0	0	1	0	0
7	0	0	0	8	0

Information loss (privacy after distortion) without the *loop effect* equals to 0.0603 (5.30), with the *loop effect* 0.0643 (5.50). When we do not take the order into account information loss (privacy after distortion) is equal to 0.0623 (6.95), probability of retaining the original value equals 0.6. Without the *loop effect* we have lower information loss and privacy comparing to the case without the order. To obtain the same level of privacy we should lower the probability of retaining the original value.
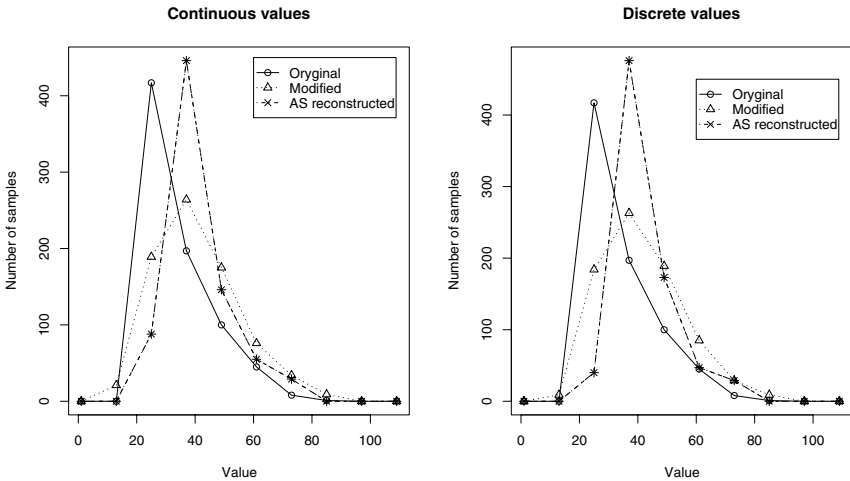

Fig. 1 Probability distribution reconstruction for continuous (on the left) and discrete (on the right) domain of attribute

Figure 1 shows original, distorted and reconstructed probability distribution of *age* attribute (set *diabetes*) for continuous and discrete domain (*staircased* cumulative distribution function was used and uniform distribution).

Information loss for continuous case is equal to 0.4529 and for discrete 0.4585. Both values are high in general. The reason is that reconstructed peak is moved to the right, what causes information loss to grow. Information loss for discrete case is slightly higher, but distorted attribute has higher privacy – 56.33 (55.85 for continuous case), because, in general, the higher privacy, the higher information loss, i.e., the privacy causes information loss.

3.2   Classification with Ordered Attributes

In the experiments with classification and decision tree we show only two best types of reconstruction: *By class* – we reconstruct probability distribution of an attribute separately for each class but only in the root node and *Local* – reconstruction is performed in every node divided into classes. The algorithms for building a decision tree over distorted data proposed in [2] and [3] were used.

Table 2 presents the results of classification of *credit-g* set for two cases: when we do not take the order into account and when the attributes with the order are distorted as shown in Sect. 2.

Table 2 Accuracy, sensitivity, specificity, precision, and F measures of classification with the usage of ordered attributes for Local (LO) and By class (BC) reconstruction types, AS.EA algorithms and 100% of privacy (set credit-g)

Ordered	Acc.		Sens.		Spec.		Prec.		F	
	LO	BC	LO	BC	LO	BC	LO	BC	LO	BC
yes	0.6796	0.6952	0.3682	0.3654	0.8152	0.8380	0.4622	0.4921	0.4009	0.4100
no	0.6777	0.6963	0.3674	0.3758	0.8129	0.8352	0.4568	0.4951	0.3980	0.4183

Comparing these two cases we can say that all measures[4] are really close, the difference is less than 0.01, but the case of sensitivity for By class reconstruction (the difference is slightly higher than 0.01). The results for different sets show the same pattern.

The experiments prove that we can use integer attributes, distort them and store as integers. Moreover, we can take the order into account in case of ordinal attributes. These methods allow miner to obtain the results with the same quality preserving the same domain of attributes and distorting attributes in accordance with their natural order.

---

[4] The definitions of used measures can be found in [7].

4 Conclusions and Future Work

We presented the new approach to ordered attributes in privacy preserving classification using the randomization-based technique and stored in a centralized database.

We proposed how to proceed with ordinal and integer attributes. Presented methods enable miner to preserve integer domain of attributes and distort the original values of ordinal attributes taking their natural order into account during distortion and reconstruction procedure.

Effectiveness of the new approach has been tested on real data sets. The results of the experiments show that proposed methods achieve the same results and allow miner to use the same domain for integer attributes and take a natural order of ordinal attributes into account.

In future works, we plan to investigate the possibility of extension of our results to preserve privacy for target (class) attribute.

References

1. Agrawal, D., Aggarwal, C.C.: On the design and quantification of privacy preserving data mining algorithms. In: Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 247–255 (2001)
2. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 439–450. ACM Press, New York (2000)
3. Andruszkiewicz, P.: Privacy preserving classification for continuous and nominal attributes. In: Proceedings of the 16th International Conference on Intelligent Information Systems (2008)
4. Andruszkiewicz, P.: Probability distribution reconstruction for nominal attributes in privacy preserving classification. In: Proceedings of the International Conference on Convergence and Hybrid Information Technology (2008)
5. Du, W., Zhan, Z.: Using randomized response techniques for privacy-preserving data mining. In: Getoor, L., Senator, T.E., Domingos, P., Faloutsos, C. (eds.) KDD, pp. 505–510. ACM, New York (2003)
6. Lindell, Y., Pinkas, B.: Privacy preserving data mining. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 36–54. Springer, Heidelberg (2000)
7. van Rijsbergen, C.J.: Information Retrieval. Butterworth-Heinemann, Newton (1979)
8. Xiong, L., Chitti, S., Liu, L.: Mining multiple private databases using a knn classifier. In: Proceedings of the ACM symposium on Applied Computing, pp. 435–440 (2007)
9. Yang, Z., Zhong, S., Wright, R.N.: Privacy-preserving classification of customer data without loss of accuracy. In: Proceedings of the 5th SIAM International Conference on Data Mining, pp. 21–23 (2005)
10. Zhang, N., Wang, S., Zhao, W.: A new scheme on privacy-preserving data classification. In: Grossman, R., Bayardo, R., Bennett, K.P. (eds.) KDD, pp. 374–383. ACM, New York (2005)

Incorporating Detractors into SVM Classification

Marcin Orchel

Abstract. As was shown recently [19], prior knowledge has a significant importance in machine learning from the no free lunch theorem viewpoint. One of the type of prior information for classification task is knowledge on the data. Here we will propose another type of prior knowledge, for which a distance from decision boundary to selected data samples (detractors) is maximised. Support Vector Machines (SVMs) is a widely used algorithm for data classification. Detractors will be incorporated into SVMs by weighting the samples. For the reason, that standard C-SVM sample weights are not suitable for maximising the distance to selected points, additional SVM weights will be proposed. We will show that detractors can enhance the classification quality for areas with lack of training samples and for time series classification. We will demonstrate that incorporating detractors to stock price predictive models can lead to increased investment profits.

Keywords: SVM, statistical classification, machine learning, detractors.

1 Introduction

Support Vector Machines (SVMs) is a widely used method for statistical classification. SVMs have been already used in many domains, such as: credit scoring [6], stock price movements [7, 1, 5], weather forecasting [15], customer relationship management [2]. Prior knowledge could significantly enhance SVM classification quality [9]. There are two main types of prior information: class invariance to some transformations of the input data, and knowledge on the data. Particular cases of the latter are unlabelled samples, imbalance of the training set and different quality of the data. There are two main possibilities to incorporate additional

Marcin Orchel
AGH University of Science and Technology,
Mickiewicza Av. 30, 30-059 Cracow, Poland
e-mail: marcino@agh.edu.pl

knowledge to classifier, either modify a feature set or modify classification algorithm. The most known example of modification of classifier is inclusion of polyhedral type of knowledge, which disallow particular class inside polyhedrons, explored in [4, 3].

In this article we propose another type of prior knowledge, that is to maximise a distance from decision boundary to some chosen points, called *detractors*. We will incorporate detractors by modifying SVM classifier.

2 Detractors

We use sample weights to incorporate detractors into SVM algorithm. Sample weights are investigated for C-SVM problem formulation in [8, 10, 16, 20] and for v-SVM in [17, 18]. In this article we will consider C-SVM formulation.

A 1-norm soft margin SVM optimisation problem with sample weights C_i is:

Optimisation problem 1. *Minimisation of:*

$$f\left(\mathbf{w}, b, \alpha, \xi\right),$$

where

$$f\left(\mathbf{w}, b, \alpha, \xi\right) = \frac{1}{2}\left|\mathbf{w}\right|^2 + C_i \sum_{i=1}^{n} \xi_i,$$

with constraints:

$$y_i g\left(A_i\right) \geq 1 - \xi_i,$$
$$\xi_i \geq 0,$$

for $i \in \{1, \ldots, n\}$.

Weights C_i are not suitable for maximizing a distance to detractors, because when for point i, $\xi_i = 0$, increasing its weight does not change a decision bound. The another possibility of using C_i weights for detractors would be decreasing weights for all points instead of the chosen point. Although lowering weights disturbs the solution, because errors are more acceptable. In order to overcome these difficulties with using C_i weights for detractors we introduce additional weights b_i in a following way:

Optimisation problem 2. *Minimisation of:*

$$f\left(\mathbf{w}, b, \alpha, \xi\right),$$

where

$$f\left(\mathbf{w}, b, \alpha, \xi\right) = \frac{1}{2}\left|\mathbf{w}\right|^2 + C_i \sum_{i=1}^{n} \xi_i,$$

with constraints:

$$y_i g(A_i) \geq 1 - \xi_i + b_i \,,$$
$$\xi_i \geq 0 \,,$$

for $i \in \{1, \ldots, n\}$.

When $b_i = 0$ we get original formulation. When $b_i < 0$ a point could lie closer to the decision bound, and when $b_i > 0$ a point could lie farther from decision bound. For detractors idea the interesting case is when $b_i > 0$. The example with different b_i values is presented in Fig. 1. Analysis shows that increasing b_i parameter leads to expected results, only when C_i parameter is high enough.
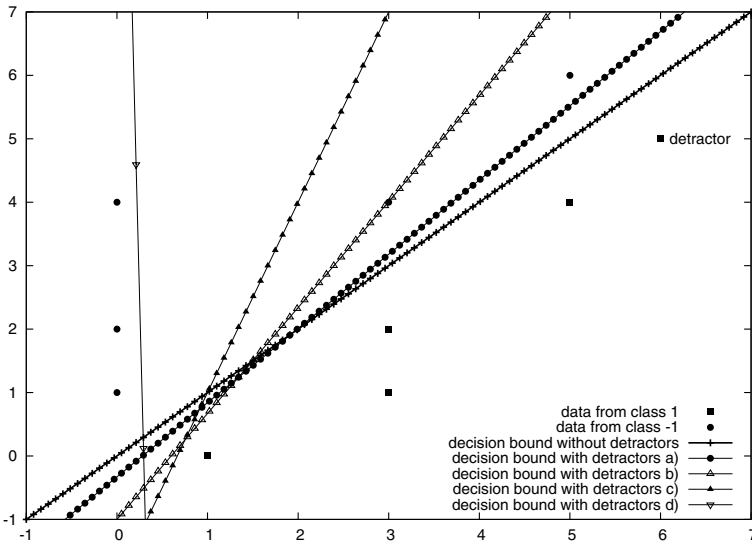
Fig. 1 Comparison of original SVM problem and SVM problem with detractors

In order to construct an efficient algorithm for the modified SVM problem we will derive its dual form. The dual problem is:

Optimisation problem 3. *Maximisation of:*

$$d(\alpha, \mathbf{r}) \,,$$

where

$$d(\alpha, \mathbf{r}) = \min_{\mathbf{w}, b} h(\mathbf{w}, b, \alpha, \xi, \mathbf{r}) \,,$$

$$h(\mathbf{w}, b, \alpha, \xi, \mathbf{r}) = \frac{1}{2} |\mathbf{w}|^2 + C_i \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \left(y_i g(A_i) - 1 + \xi_i - b_i \right) - \sum_{i=1}^{n} r_i \xi_i \,,$$

with constraints

$$\alpha_i \geq 0\,,$$
$$r_i \geq 0\,,$$

for $i \in \{1,\ldots,n\}$.

Partial derivative with respect to w_i is:

$$\frac{\partial h\left(\mathbf{w},b,\alpha,\xi,\mathbf{r}\right)}{\partial w_i} = w_i - \sum_{j=1}^{n} \alpha_j y_j a_{ji} = 0\,,$$

for $i \in \{1,\ldots,m\}$.
Partial derivative with respect to b is:

$$\frac{\partial h\left(\mathbf{w},b,\alpha,\xi,\mathbf{r}\right)}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0\,.$$

Partial derivative with respect to ξ_i is:

$$\frac{\partial h\left(\mathbf{w},b,\alpha,\xi,\mathbf{r}\right)}{\partial \xi_i} = C_i - r_i - \alpha_i = 0\,.$$

After substitution of above equations to $d\left(\alpha,\mathbf{r}\right)$ we finally get:

Optimisation problem 4. *Maximisation of:*

$$d\left(\alpha\right)\,,$$

where

$$d\left(\alpha\right) = \sum_{i=1}^{n} \alpha_i \left(1 + b_i\right) - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_i y_k y_i \sum_{j=1}^{m} a_{ij} a_{kj}\,,$$

with constraints

$$\sum_{i=1}^{n} \alpha_i y_i = 0\,,$$
$$\alpha_i \geq 0\,,$$
$$\alpha_i \leq C_i\,,$$

for $i \in \{1,\ldots,n\}$.

In the above formulation similarly as for original SVM problem it is possible to use kernel function instead of scalar product, thus a new formulation can be used for non-linear SVM classification:

Optimisation problem 5. *Maximisation of:*

$$d\left(\alpha\right)\,,$$

where

$$d\left(\alpha\right) = \sum_{i=1}^{n} \alpha_i \left(1 + b_i\right) - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_k \alpha_i y_k y_i K_{ik} ,$$

with constraints

$$\sum_{i=1}^{n} \alpha_i y_i = 0 ,$$

$$\alpha_i \geq 0 ,$$

$$\alpha_i \leq C_i ,$$

for $i \in \{1, \ldots, n\}$.

SVM with detractors for nonlinear case is depicted in Fig. 2.
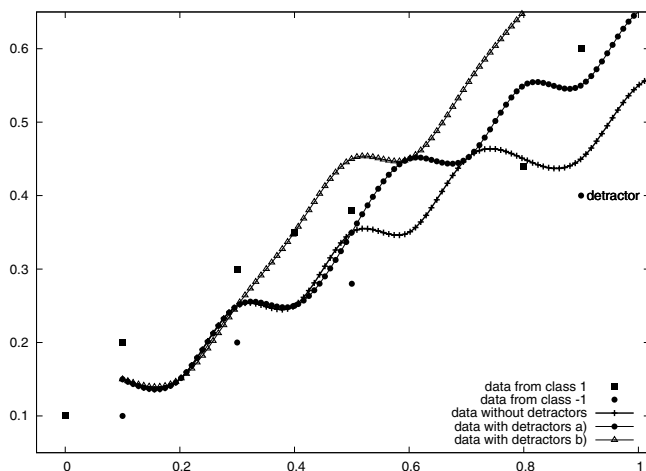
Fig. 2 Comparison of original SVM problem and SVM problem with detractors for nonlinear case

We will derive an efficient algorithm for solving Opt. problem 5, which is similar to Sequential Minimization Algorithm (SMO) [14], which solves original SVM dual optimisation problem.

For two parameters the new objective function has a form:

$$d\left(\alpha_1, \alpha_2\right) = \alpha_1 b_1 + \alpha_2 b_2 + d_{\text{old}} .$$

After substituting:

$$\alpha_1 = \gamma - y_1 y_2 \alpha_2 ,$$

where

$$\gamma = \alpha_1^{\text{old}} + y_1 y_2 \alpha_2^{\text{old}}$$

we get:

$$d(\alpha_1, \alpha_2) = b_1\gamma - b_1 y_1 y_2 \alpha_2 + \alpha_2 b_2 + d_{\text{old}}.$$

After differentiating we get:

$$\frac{\partial d(\alpha_1, \alpha_2)}{\partial \alpha_2} = b_2 - b_1 y_1 y_2 + \frac{\partial d_{\text{old}}(\alpha_1, \alpha_2)}{\partial \alpha_2}.$$

And a solution is

$$\alpha_2^{\text{new}} = \alpha_2 + \frac{y_2(E_1 - E_2)}{\kappa},$$

where

$$E_i = \sum_{j=1}^{n} y_j \alpha_j K_{ij} - y_i - y_i b_i \tag{1}$$

$$\kappa = K_{11} + K_{22} - 2K_{12}.$$

After that, α_2 is clipped in the same way as for SMO with different weights:

$$U \leq \alpha_2 \leq V,$$

where, when $y_1 \neq y_2$

$$U = \max\left(0, \alpha_2^{\text{old}} - \alpha_1^{\text{old}}\right),$$

$$V = \min\left(C_2, C_1 - \alpha_1^{\text{old}} + \alpha_2^{\text{old}}\right),$$

when $y_1 = y_2$

$$U = \max\left(0, \alpha_1^{\text{old}} + \alpha_2^{\text{old}} - C_1\right),$$

$$V = \min\left(C_2, \alpha_1^{\text{old}} + \alpha_2^{\text{old}}\right).$$

Parameter α_1 is computed in the same way as for SMO.

Karush Kuhn Tucker complementary condition is:

$$\begin{cases} \alpha_i(y_i g(A_i) - 1 - b_i + \xi_i) = 0, \\ (C - \alpha_i)\xi_i = 0. \end{cases}$$

Based on this condition it is possible to derive equations for SVM heuristic and SVM stop criteria. Equations for original heuristic are described in [12, 11]. After incorporating weights b_i a heuristic and stopping criteria are almost the same, with one difference, that E_i are computed as stated in (1).

Detractors modify classification bound based on external knowledge. An equivalent implementation would be to modify feature values, in the way that classification bound changes its position in the same way as original implementation does. But there are some differences between these approaches. In the second approach one

need to derive a method to modify properly feature values. Modification of feature values is computationally demanding task especially for large data sets.

3 Testing

Detractors are valuable in two cases. When data for some area are unavailable, and there is an external knowledge about empty areas, which could be expressed by detractors. The second case is for classifying time series data. Classification of time series data is analysed in [13]. The most common way of classifying time series is to transform them to fixed number of attributes, then apply static classification algorithm. Although it is sometimes desirable to create dynamically parametrized classification model, in which decision boundary depends on time periods and is controlled by detractors.

A concept of detractors for the time series case will be incorporated to stock price movements predictive model. In our example, we analyse NASDAQ daily data from 02-04-2007. We have 6 features, every feature is a percentage daily growth for a previous day, for a day before previous day, etc. Classification value is 1, when there was a growth from previous day, otherwise is -1. We have 100 training vectors and 356 testing vectors. In Table 1 we can see comparison of original algorithm, and algorithm with detractors. In our examples we choose arbitrarily two detractors. Although in real stock prediction systems detractors should be chosen based on used trading strategies.

Table 1 Comparison of SVM original algorithm and SVM algorithm with detractors

SVM algorithm	misclassified training data	misclassified test data
without detractors	3	182
with 2 detractors	5	156

We showed, that detractors can be incorporated into Support Vector Machines in an efficient way. Moreover detractors is a useful type of prior knowledge, which allows to control dynamic classification models. Though finding detractors is a domain specific task and could be a challenging one.

References

1. Chen, W.H., Shih, J.Y., Wu, S.: Comparison of support-vector machines and back propagation neural networks in forecasting the six major Asian stock markets. International Journal of Electronic Finance 1(1), 49–67 (2006)
2. Coussement, K., Van den Poel, D.: Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. Expert Systems and Applications 34(1), 313–327 (2008)
3. Fung, G.M., Mangasarian, O.L., Shavlik, J.W.: Knowledge-based nonlinear kernel classifiers. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT/Kernel 2003. LNCS, vol. 2777, pp. 102–113. Springer, Heidelberg (2003)
4. Fung, G.M., Mangasarian, O.L., Shavlik, J.W.: Knowledge-based support vector machine classifiers. In: Becker, S., Thrun, S., Obermayer, K. (eds.) Advances in Neural Information Processing Systems, vol. 15, pp. 521–528. Massachusetts Institute of Technology Press, Cambridge (2003)
5. Gao, C., Bompard, E., Napoli, R., Cheng, H.: Price forecast in the competitive electricity market by support vector machine. Physica A: Statistical Mechanics and its Applications 382(1), 98–113 (2007)
6. Huang, C.L., Chen, M.C., Wang, C.J.: Credit scoring with a data mining approach based on support vector machines. Expert Systems and Applications 33(4), 847–856 (2007)
7. Huang, W., Nakamori, Y., Wang, S.Y.: Forecasting stock market movement direction with support vector machine. Computer Operation Research 32(10), 2513–2522 (2005)
8. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of the 16th International Conference on Machine Learning, pp. 200–209. Morgan Kaufmann Publishers Inc., San Francisco (1999)
9. Lauer, F., Bloch, G.: Incorporating prior knowledge in support vector machines for classification: A review. Neurocomputing 71(7-9), 1578–1594 (2008)
10. Lin, C.F., Wang, S.D.: Fuzzy support vector machines. IEEE Transaction on Neural Networks 13(2), 464–471 (2002)
11. Orchel, M.: Support vector machines: Sequential multidimensional subsolver (SMS). In: Dabrowski, A. (ed.) Proceedings of Signal Processing: Algorithms, Architectures, Arrangements, and Applications, pp. 135–140. IEEE, Los Alamitos (2007)
12. Orchel, M.: Support vector machines: Heuristic of alternatives (HoA). In: Romaniuk, R.S. (ed.) Proceedings of Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments, vol. 6937. SPIE (2008)
13. Orsenigo, C., Vercellis, C.: Time series classification by discrete support vector machines. In: Artifical Intelligence and Data Mining Workshop (2006)
14. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. In: Advances in kernel methods: support vector learning, pp. 185–208. Massachusetts Institute of Technology Press, Cambridge (1999)
15. Trafalis, T.B., Adrianto, I., Richman, M.B.: Active learning with support vector machines for tornado prediction. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007. LNCS, vol. 4487, pp. 1130–1137. Springer, Heidelberg (2007)
16. Wang, L., Xue, P., Chan, K.L.: Incorporating prior knowledge into SVM for image retrieval. In: Proceedings of the 17th International Conference on Pattern Recognition, vol. 2, pp. 981–984. IEEE Computer Society, Washington (2004)
17. Wang, M., Yang, J., Liu, G.P., Xu, Z.J., Chou, K.C.: Weighted-support vector machines for predicting membrane protein types based on pseudo amino acid composition. Protein Engineering, Design & Selection 17(6), 509–516 (2004)

18. Wei, H., Jia, Y., Jia, C.: A new weighted nu-support vector machine. In: Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing. ACM, New York (2007)
19. Wolpert, D.H.: The lack of a priori distinctions between learning algorithms. Neural Computation 8(7), 1341–1390 (1996)
20. Wu, X., Srihari, R.: Incorporating prior knowledge with weighted margin support vector machines. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 326–333. ACM, New York (2004)

Bayes Multistage Classifier
and Boosted C4.5 Algorithm
in Acute Abdominal Pain Diagnosis

Robert Burduk and Michał Woźniak

Abstract. The medical decision problem – acute abdominal pain diagnosis is presented in the paper. We use two methods of classification, which are based on a decision tree scheme. The first of them generates classifier only based on learning set. It is boosted C4.5 algorithm. The second approach is based on Bayes decision theory. This decision algorithm utilizes expert knowledge for specifying decision tree structure and learning set for determining mode of decision making in each node. The experts-physicians gave the decision tree for performing Bayes hierarchical classifier.

Keywords: Bayes classifier, medical diagnosis, decision tree, expert knowledge.

1 Introduction

Medical diagnosis is very important and attractive area of implementation decision support systems. About 11% of expert systems are dedicated to the medical aided diagnosis and ca 21% of papers connected with application of mentioned methods are illustrated by the medical cases [8]. One of the first and well-known expert system dedicated to medical aided diagnosis is MYCIN [17], which is considered as the exemplar of expert system by many researchers. There are many papers which are corresponded to mentioned systems. Some of them present the reviews of description of working medical decision support software [5, 9], the others are related to the problem of choosing the best method of classification for the particular medical task [11, 19]. Our paper concentrates on designing software for the acute abdominal pain diagnosis.

Robert Burduk · Michał Woźniak
Chair of Systems and Computer Networks, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail:{robert.burduk,michal.wozniak}@pwr.wroc.pl

Paper describes our research into qualities of classifiers based on Bayesian approach which utilizes the schema of decision given by expert. In the paper we compare results of mentioned experiments to quality of classifier obtained via boosted C4.5 procedure which does not use expert knowledge during the learning.

The content of the work is as follows. Section 2 introduces idea of Bayesian approach, decision tree induction algorithm, and methods of improving and stabilizing classifiers. In the next section we describe mathematical model of the acute abdominal pain decision problem. Then we presents conditions and results of the experimental investigations of the proposed algorithms. The last section concludes the paper.

2   The Multistage Recognition Task

The basic idea involved in any multistage approach is to break up a complex decision into several simpler classifications [14]. The decision tree classifier and the hierarchical classifier are two possible approaches to multistage pattern recognition. Hierarchical classifiers are a special type of multistage classifiers which allow rejection of class labels at intermediate stages. The synthesis of hierarchical classifier is a complex problem. It involves specification of the following component:

- design of a decision tree structure,
- feature selection used at each noterminal node of decision tree,
- choice of decision rules for performing the classification.

Let us present shortly ideas of two approaches. The first of them use given decision tree structure and set of features for each tree's node. This method focuses its attention on decision rules construction based on Bayesian approach for each node. The second approach based on top down induction algorithm and focuses its attention on decision tree construction. In each node only test on individual feature is performed.

2.1   Bayesian Hierarchical Classifier

The procedure in the Bayesian hierarchical classifier consists of the following sequences of operations. At the first stage, some specific features x_0 are measured. They are chosen from among all accessible features x, which describe the pattern that will be classified. These data constitute a basis for making a decision i_1. This decision, being the result of recognition at the first stage, defines a certain subset in the set of all classes and simultaneously indicates features x_{i_1} (from among x) which should be measured in order to make a decision at the next stage.

Now at the second stage, features x_{i_1} are measured, which together with i_1 are a basis for making the next decision i_2. This decision, – like i_1 – indicates features x_{i_2} that are necessary to make the next decision (at the third stage as in the previous stage) that in turn defines a certain subset of classes, not in the set of all classes,

but in the subset indicated by the decision i_2, and so on. The whole procedure ends at the Nth stage, where the decision made i_N indicates a single class, which is the final result of multistage recognition.

Let us consider a pattern recognition problem, in which the number of classes equals M. Let us assume that the classes are organised in a $(N+1)$ horizontal decision tree. Let us number all the nodes of the constructed decision-tree with consecutive numbers of $0, 1, 2, \ldots$, reserving 0 for the root-node, and let us assign numbers of classes from the $\mathcal{M} = \{1, 2, \ldots, M\}$ set to terminal nodes so that each one of them can be labelled with the number of the class connected with that node. This allows us to introduce the following notation:

- $\mathcal{M}(n)$ – the set of nodes, which distance from the root is n, $n = 0, 1, 2, \ldots, N$. In particular $\mathcal{M}(0) = \{0\}$, $\mathcal{M}(N) = \mathcal{M}$,
- $\overline{\mathcal{M}} = \bigcup_{n=0}^{N-1} \mathcal{M}(n)$ – the set of interior nodes (non terminal),
- $\mathcal{M}_i \subseteq \mathcal{M}(N)$ – the set of class labels attainable from the ith node ($i \in \overline{\mathcal{M}}$),
- $\mathcal{M}^i$ – the set of nodes of immediate descendant node i ($i \in \overline{\mathcal{M}}$),
- m_i – node of direct predecessor of the ith node ($i \neq 0$),
- $s(i)$ – the set of nodes on the path from the root-node to the ith node, $i \neq 0$.

Our aim is now to calculate the so-called multistage recognition strategy $\pi_N = \{\Psi_i\}_{i \in \overline{\mathcal{M}}}$, that is the set of recognition algorithms in the form:

$$\Psi_i : X_i \rightarrow \mathcal{M}^i, \quad i \in \overline{\mathcal{M}} . \tag{1}$$

Formula (1) is a decision rule (recognition algorithm) used at the ith node that maps observation subspace to the set of immediate descendant nodes of the ith node. Analogically, decision rule (1) partitions observation subspace X_i into disjoint decision regions $D_{x_i}^k$, $k \in \mathcal{M}^i$, such that observation x_i is allocated to the node k if $k_i \in D_{x_i}^k$, namely:

$$D_{x_i}^k = \{x_i \in X_i : \Psi_i(x_i) = k\}, \; k \in \mathcal{M}^i, \quad i \in \overline{\mathcal{M}}. \tag{2}$$

Globally optimal strategy π_N^*. This strategy minimises the mean probability of misclassification on the whole multistage recognition process and leads to an optimal global decision strategy, whose recognition algorithm at the nth stage is the following:

$$\Psi_{i_n}^*(x_{i_n}) = i_{n+1} \quad \text{if} \tag{3}$$

$$i_{n+1} = \underset{k \in \mathcal{M}^{i_n}}{\arg\max} \, \mathrm{Pc}(k) p(k) f_k(x_{i_n})$$

where $\mathrm{Pc}(k)$ is the empirical probability of correct classification at the next stages if at the nth stage decision i_{n+1} is made.

As we mentioned bellow in practice, the unknown probabilistic characteristics (values of the prior probabilities and probability density functions) are replaced by their estimators obtained via parametric or nonparametric approaches [3].

2.2 Inductive Decision Tree

The decision tree induction algorithms have been used for several years [1, 10]. Generally speaking they propose an approximation discrete function method which is adopted to the classification task. It is one of the most important methods for classification which achieve very good classification quality in many practical decision support systems.

Many decision-tree algorithms have been developed. The most famous are CART [2], ID3 [12] and its modification C4.5 [13]. ID3 is a typical decision-tree algorithm. It introduces information entropy as the splitting attribute's choosing measure.

The central choice in the ID3 algorithm is selecting 'the best' attribute (which attribute to test at each node in the tree). The proposed algorithm uses the information gain that measures how well the given attribute separates the training examples according to the target classification. This measure based on the Shanon's entropy of learning set S:

$$Entropy(S) = \sum_{i=1}^{M} -p_i \log_2 p_i, \tag{4}$$

where p_i is the proportion of S belonging to class i ($i \in \mathbf{M}, \mathbf{M} = 1, 2, \ldots, M$). The information gain of an attribute A relative to the collection of examples S, is defined as:

$$Gain(S, A) = Entropy(S) - \sum_{c \in values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{5}$$

where $values(A)$ is the set of all possible values for attribute A and S_v is the subset of S for which $A = v$. As we mentioned above the C4.5 algorithm is an extended version of ID3. It improves appropriate attribute selection measure, avoids data over fitting, reduces error pruning, handles attributes with different weight, improves computing efficiency, handles missing value data and continuous attributes, and performs other functions. C4.5 instead of information gain in ID3 use an information gain ratio.

One of the main advantage of this decision tree is that we can easy convert obtained tree into the set of rules (each path from that form is a decision rule). This form of knowledge is the most popular form of knowledge representation in the most of expert systems [8].

2.3 Boosting

Boosting is general method of producing an accurate classifier on base of weak and unstable one [16]. It is often called metaclassifier. The idea derives from PAC (*Probably Approximately Correct*) theory. The underlying idea of boosting is to combine simple classifiers to form an ensemble such that the performance of the single member of ensemble is improved [15]. As we see the main problem of the boosting is how to construct ensemble. The one of the most popular algorithm AdaBoost [4] produces at every stage, a classifier which is trained with the data. The output of the classifier is then added to the output of classifier ensemble, with the strength

is proportional to how accurate obtained classifier is. Then, the data is reweighted: examples that the current learned function gets wrong are 'boosted' in importance, so that the classifier obtained at the next stage attempts to fix the errors. The main advantage of boosting is that it often does not suffer from overfitting.

3 Model of Acute Abdominal Pain Diagnosis

The first mathematical model of acute abdominal pain was given in [6]. We simplified it hover the experts from the Clinic of Surgery, Wroclaw Medical Academy, regarded that stated problem of diagnosis as very useful. It leads to the following classification of the AAP:

- cholecystitis,
- pancreatitis,
- non-specyfic abdominal pain,
- rare disorders of 'acute abdominal',
- appendicitis,
- divercitulitis,
- small-bowel obstruction,
- perforated peptic ulcer.

Although the set of symptoms necessary to correctly assess the existing APP is pretty wide, in practice for the diagnosis, results of 31 (non-continuos) examinations are used [6]. Since the abdominal area contains many different organs it is divided in smaller areas [18]. One division method, uses one median sagittal plane and one transverse plane that passes through the umbilicus at right angles. This method divides the abdomen into four left and right upper, left and right lower quadrants. For our study we use the more precise description of abdominal pain location [6].

The experts-physicians gave the decision tree depicted in Fig. 1 [7]. The interior nodes are corresponded with the following diagnosis:
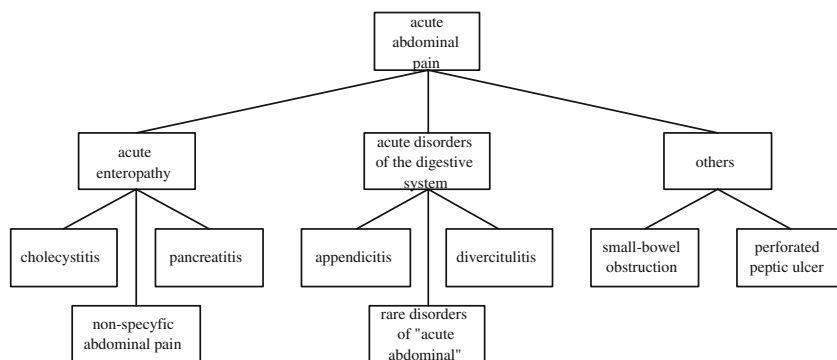

Fig. 1 Heuristic classifier for the APP diagnosis problem

- acute enteropathy,
- acute disorders of the digestive system,
- others.

4 Experimental Investigation

The aim of the experiment is to compare the errors of Bayesian classifiers with the quality of classifiers obtained via induction learning procedure. The following classifiers and fusion methods were chosen:

1. Multistage classifiers used heuristic decision tree and Bayesian classifier in each node according the global optimal strategy. For these classifiers the estimators of the conditional probability density function were obtained via *k-n-Nearest Neighbor*.
2. Classifier obtained via C4.5 procedure boosted by AdaBoost.M1.

The conditions of experiments were as follow:

1. All experiments were carried out in WEKA environment [21] and own software created in Matlab environment.
2. Errors of the classifiers were estimated using the ten fold cross validation method.

4.1 Computer Experiment Evaluation

Firstly, one has to note that we are aware of the fact that the scope of computer experiments were limited. Therefore, making general conclusions based on them is very risky. In our opinion mentioned below statements should not be generalized at this point, but they should be confirmed by other experiments in much broader scope.

The results of experiments are presented in Table 1. The following conclusions can be drawn from following experiments:

1. C4.5 algorithm generated the worst classifier but we might suspect that we could improve it quality by the presentation of more numerous learning set. It was conformed by the experiment with its boosted version. The AdaBoost.M1 algorithm improved the quality of decision tree more than 10%.
2. We have to notice that the qualities of the multistage classifier based on Bayes decision theory and boosted C4.5 classifier are similar. This observation leads us to the conclusion that for this medical case we could give the expert knowledge abut shape of decision tree up.
3. Additional advantage of boosted decision tree classifier over Bayesian one is that tree generated by C4.5 makes decision on the basis of the values of 18 attributes. The multistage classifier based on Bayes decision theory uses the whole set of attributes. It means that we can use less number of attribute (lower cost of diagnosis) for making decision on the similar level. Generating cheap

Table 1 Frequency of correct classifications

Class number	Globally optimal strategy	Boosted C4.5
1	95.32%	95.04%
2	62.62%	64.71%
3	100.00%	96.55%
4	86.84%	92.86%
5	96.21%	96.36%
6	85.48%	90.63%
7	99.30%	96.18%
8	94.21%	100.00%
Average	**94.82%**	**94.33%**

diagnosis aided computer tools is actual problem of researches so-called cost-sensitive methods [20].

4. Experts revised the structures of classifiers given by boosted C4.5 version. They confirmed that the most of rules were correct and maybe the heuristic tree given by them on the beginning are too simplified.

5 Final Remarks

The recognition method based on of compound, hierarchical Bayesian approach, inductive learning and concept of metaclassifier were presented. The classifiers generated by those algorithms were applied to the medical decision problem (recognition of Acute Abdominal Pain).

It must be emphasised that we have not proposed a method of 'computer diagnosis'. What we have proposed are the algorithms whose can be used to help the clinicians to make their own diagnosis. The presented empirical results for the inductive learning classifiers and heuristic one demonstrates the effectiveness of the proposed concepts in such computer-aided medical diagnosis problems. Advantages of the proposed methods make it attractive for a wide range of applications in medicine, which might significantly improve the quality of the care that the clinicians can give to their patients.

References

1. Alpaydin, E.: Introduction to Machine Learning. The Massachusetts Institute of Technology Press, London (2004)
2. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Decision trees. Wadsworth, Belmont (1984)

3. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons, Chichester (2000)
4. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and application to boosting. Journal of Computer and System Science 55(1), 119–139 (1997)
5. Kaplan, B.: Evaluating informatics applications – clinical decision support systems literature review. International Journal of Medical Informatics 64, 15–37 (2001)
6. Kurzyński, M.: Diagnosis of acute abdominal pain using three-stage classifier. Computers in Biology and Medicine 17(1), 19–27 (1987)
7. Kurzyński, M.: On the multistage bayes classifier. Pattern Recognition 21, 355–365 (1988)
8. Liebowitz, J. (ed.): The Handbook of Applied Expert Systems. CRC Press, Boca Raton (1998)
9. Mextaxiotis, K., Samouilidis, J.E.: Expert systems in medicine: academic illusion or real power? Information Management and Computer Security 8(2), 75–79 (2000)
10. Mitchell, T.M.: Machine Learning. McGraw-Hill Company Incorporated, New York (1997)
11. Polat, K., Gunesa, S.: The effect to diagnostic accuracy of decision tree classifier of fuzzy and k-NN based weighted pre-processing methods to diagnosis of erythemato-squamous diseases. Digital Signal Processing 16(6), 922–930 (2006)
12. Quinlan, J.R.: Induction on decision tree. Machine Learning 1, 81–106 (1986)
13. Quinlan, J.R.: C4.5: Program for Machine Learning. Morgan Kaufman, San Mateo (1993)
14. Safavian, S.R., Landgrebe, D.: A survey of decision tree classifier methodology. IEEE Transactions on Systems, Man and Cybernetics 21(3), 660–674 (1991)
15. Schapire, R.E.: The boosting approach to machine learning: An overview. In: Proceedings of the MSRI Workshop on Nonlinear Estimation and Classification, Berkeley, US (2001)
16. Shapire, R.E.: The strength of weak learnability. Machine Learning 5(2), 197–227 (1990)
17. Shortliffe, E.: MYCIN: Computer-based Medical Consultations. Elsevier, New York (1975)
18. Townsend, C.M., Beauchamp, R.D., Evers, B.M., Mattox, K.L.: Sabiston Textbook of Surgery, 17th edn. WB Saunders, St. Louis (2004)
19. Ubeyli, E.D.: Comparison of different classification algorithms in clinical decision-making. Expert Systems 24(1), 17–31 (2007)
20. Viaene, S., Dedene, G.: Cost-sensitive learning and decision making revisited. European Journal of Operational Research 166, 212–220 (2005)
21. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers, San Francisco (2000)

Part VII
Pattern Recognition and Signal Processing

Skrybot – A System for Automatic Speech Recognition of Polish Language

Lesław Pawlaczyk and Paweł Bosky

Abstract. In this article we present a system for clustering and indexing of automatically recognised radio and television news spoken in Polish language. The aim of the system is to quickly navigate and search for information which is not available in standard internet search engines. The system comprises of speech recognition, alignment and indexing module. The recognition part is trained using dozens of hours of transcribed audio and millions of words representing modern Polish language. The training audio and text is then converted into acoustic and language model, where we apply techniques such as Hidden Markov Models and statistical language processing. The audio is decoded and later submitted into indexing engine which extracts summary information about the spoken topic. The system presents a significant potential in many areas such as media monitoring, university lectures indexing, automated telephone centres and security enhancements.

Keywords: speech recognition, speech processing, pattern recognition.

1 Introduction

Speech recognition has been an item of research for more than 40 years by now. It is a domain which proves to be constantly challenging, because of the vast differences in pronunciation, accent, noise and quality of recorded material. In this article we present a full speech recognition system which was based on customised open source code projects and programming environments and later used for decoding

Lesław Pawlaczyk
Insitute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: `leslaw.pawlaczyk@polsl.pl`

Paweł Bosky
e-mail: `poczta@przepisywanie.pl`
`www.przepisywanie.pl`

of spoken Polish language into text. The original part of the system is a design of automated procedures for training both acoustic and language models for Polish language. We devised a method of automatic cutting and aligning recorded speech with its transcription as well as a set of rules and phones for generating the lexicon of Polish words. The article is divided into 4 sections. In Sect 2.1 we provide a short theoretical background of the system's work and in Sects. 3 and 4 we present experiments along with a discussion of the results and plans for future work.

2 Theory

2.1 Speech Signal

Speech signal (SS) can be considered as a series of pressure changes among the listener and the sound source. The most popular way of modelling speech is an oscillogram. Recording of speech on a computer is done through a sound card which samples the incoming analog signal according to a desired frequency. In speech processing research [3] it is agreed that a typical frequency is 16 kHz and in case of telephony it's 8 kHz. In this article we assume that the sampling frequency used is 16 kHz. There are different methods for modelling pronunciation of words within the SS. We modelled words by dividing them into basic units called phones. Each phone had a different representation in terms of its duration and pitch changes.

In order to simplify decoding of SS, the signal was divided into equally long frames of duration 25 ms. Typically we could fit exactly 160 speech samples into one frame. The frame was then transformed into a vector with floating point decimal values. Before the analysis of SS we performed DC offset removal, pre-emphasizing using first order difference equation and Hamming windowing to remove discontinuities at the frame edges [5]. Finally, using linear prediction and filter bank analysis combined with vocal tract length normalisation and cepstral features, we could extract a vector of 39 features which was later a source in acoustic model training.

2.2 Speech Decoding Task

Let's treat SS input as an array of individual observations O, where each element in the array can be considered as a single vector: $O = o_1 o_2 \ldots o_T$. If we assume, that a sequence of words W, corresponding to SS, could be represented as: $W = w_1 w_2 \ldots w_N$, we can then define the task of speech decoding as finding the best matching sequence W for a given series of observations O. In [3], a formula for describing this process was devised:

$$\hat{W} = \operatorname*{argmax}_{W \in L} P(O \mid W) P(W) \,.$$

$P(W)$ is a language model (*LM*) prior probability, $P(O \mid W)$ is an acoustic model likelihood and L is a lexicon. The most popular algorithm for performing the task
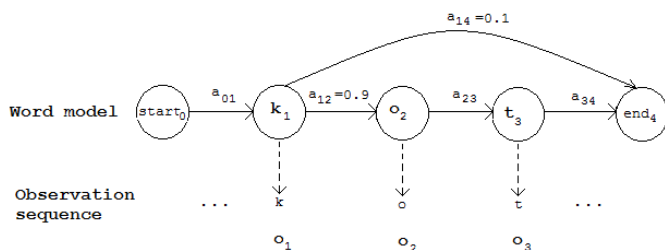
Fig. 1 Sample HMM for a word 'kot'. Transition probabilities a_{xy} between states x and y are 1.0, if not specified differently. Every circle depicts a single state inside HMM

of speech decoding, where both of the probabilities can be computed, is a so called Viterbi algorithm [2]. In the following part of this article we will shortly describe, how the speech decoder we employed is organised.
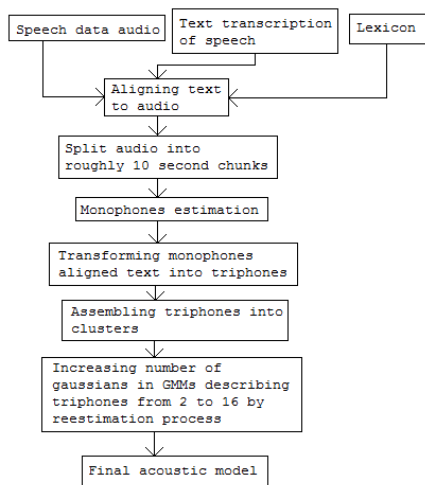
2.3 Acoustic Model

For the purpose of modeling Polish speech we decided to apply a widely known technique of Hidden Markov Models (HMMs) [5]. HMMs are regarded as finite state machines, where time acts as a parameter controlling the way a single HMM changes its state. On the input of an *HMM* we have a speech vector O, called the observation sequence $O = o_1 o_2 \ldots o_T$. To define an HMM we need to specify its states $Q = q_1, q_2, \ldots, q_N$, transition of probabilities $A = a_{01} a_{02} \ldots a_{n1} \ldots a_{nn}$, observation likelihoods $B = b_i(o_t)$, initial distribution over states $\pi_j = 0$ and a set of legal accepting states. In Fig. 1 we can see a sample HMM describing word 'kot':

In our system we used an iterative method of estimating the parameters of HMMs [5]. Short description of the process can be found in Fig. 2. The training data and transcription was divided into small fragments shorter than 1 minute. In our case we decided to go for audio chunks of roughly 10 seconds size. The size of each chunk was different, because they were divided on words boundaries, to avoid splitting words into different audio files. In the first step, transcription was roughly aligned with speech, based on estimated initial models of individual phones, using Viterbi algorithm [2]. The process of alignment was repeated several times until the underlying speech alignment error was lower, than a minimal threshold. The next step was to transit from monophone acoustic transcription to triphones [5]. Triphones are triplets of monophones with a centre monohophone surrounded by a left and right neighbour.

If we stopped training the acoustic model on the level of monophones, we would end up having a too generic statistical description of speech. Therefore by replacing monophones with triphones, we could improve the specificity of the acoustic model, by modelling how monophones occurr in a context of other phones. For example a word KOT modelled using monophones would be presented in a lexicon as

```
k o t,
```

Fig. 2 Acoustic model
training

whereas in case of triphones it would be modelled by

```
k+o k-o+t o-t.
```

After triphones estimation, we assembled the most similar sets of triphones into clusters. The goal of this step was to decrease the size of acoustic model which can grow in later stages of training procedure to considerable size, causing memory problems. The penultimate step of training acoustic model was to improve accuracy of HMMs internal states, by increasing the number of Gaussian mixtures from 2 up to 16. Gaussian mixtures were used for modelling feature vectors, which represent a specific state inside an HMM [3]. The final model was stored as a text file, containing triphones descriptions and their underlying HMMs.

2.4 Language Model

The most popular method of modelling language is by employing statistical methods [1], which describe probability of specific words occurrence in relation to other words. For this purpose we had to collect a so called corpus of Polish language. The corpus would cover the most popular topics and events which were in the focus of the speech recognition task. The procedure of training the language model required to normalise the text collected from various sources: websites, newspapers, etc. This process involved transforming abbreviations, numbers, and other formal symbols to their full text representations. Finally when the text was normalised, we replaced all the punctuation symbols (commas, dots, semicolons, exclamation and question marks) with a symbol <s>, which stands for a sentence end.

The language corpus contains a limited number of words. Our concern was to choose the most frequent words and extract all the fragments, where they appear in conjunction with other words. The most frequent N words would later form a

so called lexicon. The creation of the language model relied on the extraction of all possible pairs (bigrams) and triplets (trigrams) of the most frequent words, and assigning them a probability p. We used a method proposed in [1] for distributing the probability mass among the known bigrams and trigrams. After the probabilities were obtained, we had a model which allowed the speech decoder, based on the history of the last word or two last recognised words (depending on decoder), to predict a next word. For example if two trigrams:

```
ALA MA KOTA (p = 0.7)
ALA MA PSA (p = 0.1)
```

have two different probabilities in LM, we can say that it is 7 times more likely that the word KOTA would be next, instead of word PSA, knowing that the two previously recognised words were ALA MA.

2.5   *Pronunciation Lexicon*

In order to provide the speech decoder and acoustic model training script with a correct pronunciation of words, we devised a set of rules which mimic the pronunciation of words in Polish language. The rules were pretty simple: for example, if we spot a vowel in a word, it was represented by a phone with identical symbol. In case of consonants followed by a letter 'i' and a vowel, we used softening symbol for the corresponding phone. Below is a short extract from a sample lexicon:

```
WYSOKIE          [WYSOKIE]        w y s o k'' e sp
ZACHWYTU         [ZACHWYTU]       z a h w y t u sp
```

A symbol 'sp' corresponds to a short-pause phone, which informs the acoustic training module to insert a short pause, after each word was pronounced.

2.6   *Julius Decoder*

Julius is an open source speech decoder, written by Professor Akinobu Lee from Nagoya Institute of Technology [4], which is constantly being developed as a platform for both research and commercial purposes. Julius has an ability to work as a typical Continuous Speech Large Vocabulary Recogniser as well as a limited grammar decoder. We were mostly interested in the large vocabulary module. For the purpose of the Polish language decoding, we used a lexicon consisting of 262 144 words. The fact behind choosing this size was concluded after experimenting with different sizes: starting from 65 536 words, going through 131 072 and finishing on 524 288 words. It emerged, that the error was the smallest for 262 144 words lexicon size.

The decoder employs a two stage decoding procedure: in the first stage the Viterbi decoder combined with bigram language model produces a set of sentence hypotheses, which are later an entry point for the reverse trigram language model recognition. The Julius decoder is written in C language and is portable across all major

operating systems (Linux, Unix, and Windows). Its modular architecture and ability to work as a standalone speech recognition server makes it perfect for our speech decoding task.

2.7 Text Mining

After creating of acoustic and language models, we wrote a set of scripts which enabled decoding television and radio broadcast news in a batch mode, i.e., we could set an overnight speech decoding process on a typical PC. After the process had ended, we aligned the decoded text with a corresponding audio using Viterbi algorithm [2]. This enabled us to associate a time stamp for each word, saying that it has been uttered in a specific point of time and lasted for a given period. The set of decoded audio files later created an input for searching program which allows a user to quickly navigate to a particular point in the recording.

The initial implementation of search engine allows only to look for either single words or simple words combinations. Soon we observed the program, that automated speech decoding for radio and TV broadcasts, allowed us to browse through audio library quickly in search for topics of interest. For example we were interested in recordings, where a word `premier` or `prezydent` appeared. The research on the system for browsing recognised speech files is in a preliminary state, but it shows, how much knowledge is hidden from traditional search engines and how we much we lose on, by not being able to browse through such sources.

3 Experiments

In our experiments, we created an acoustic model based on 40 hours of transcribed audio from TV and radio broadcasts which concentrated on political news. The recordings were later automatically divided into 10 second chunks with corresponding transcription using Viterbi decoder [5]. We ended up building an acoustic model which covered all possible triphone combinations from language model lexicon plus words occurring in manual transcriptions.

Language model was prepared using normalised data from Polish speech corpus with 40 million words. During language model training we used cut-off weight of 1 for unigram, bigrams and trigrams. After experimenting with various vocabulary ranges, we decided to go for a vocabulary of size 262 144 words. The range was optimal both in terms of decoding accuracy and speed. Interestingly employing bigger language models not only did not improve speech recognition, but also slowed down the decoding process.

After repeating many tests with different acoustic and language models we observed, that the role both of these models play in the decoding process is essentially vital. Basically we could not achieve satisfying results with poor quality models (low amount of transcribed audio and small number of training words in language

corpus). The tests performed, gave us an average word error rate WER [3] of 27.2% in a 5 hour test set.

4 Discussion and Conclusions

Our system relies entirely on open source elements, starting from HTK [5] through Julius and Linux platform, where the majority of scripts and programs were written in bash, C++, and Perl languages. The concept was born, due to an almost complete lack of Polish language decoding system on the market. The development of the system up to the state, where WER = 27.2%, took us more than a year, but the final result overcame our most optimistic expectations. The system was called *Skrybot* and we intend to use it in both commercial and non-profit environments. Non-profit side will include helping disabled people in their communication with a computer, as well as potentially could be a tool for teaching correct pronunciation of individuals with speech impairment problems. We also aim to introduce the system into academic environment to trigger more research on automatic decoding of Polish speech and speech recognition in general. The benefits from using speech decoding can be also seen in automatic tracking and indexing of big audio databases. The inaccuracy of the system encourages us to review and employ fuzzy logic search algorithms which could lead to the creation of a full multimedia indexing system. Our future plans include further development of acoustic and language models along with our own open source speech decoder dedicated to a broader academic community.

References

1. Clarkson, P.R., Rosenfeld, R.: Statistical language modeling using the CMU-Cambridge toolkit. In: Proceedings of the European Conference on Speech Communication and Technology (1997)
2. Formey, G.D.: The Viterbi algorithm. Proceedings of the IEEE 61, 268–278 (1973)
3. Jurafsky, D., Martin, J.H.: Machine translation. In: Ward, N., Jurafsky, D. (eds.) Speech and Language Processing. Prentice-Hall, Englewood Cliffs (2000)
4. Lee, A., Kawahar, T., Shikano, K.: Julius – an open source real-time large vocabulary recognition engine. In: Proceedings of the European Conference on Speech Communication and Technology, pp. 1691–1694 (2001)
5. Young, S., et al.: The HTK book (for HTK version 3.4). Cambridge University Engineering Department (2006)

Speaker Verification Based on Fuzzy Classifier

Adam Dustor

Abstract. This article presents a new type of a fuzzy classifier applied to speaker verification. Obtained decision function is nonlinear. Achieved verification accuracy is compared with classical techniques like Gaussian mixture models and vector quantization. Polish speech corpora ROBOT was applied as a training and testing set. Obtained results are discussed.

Keywords: speaker verification, identification.

1 Introduction

Speaker recognition is the process of recognizing who is speaking by analysis speaker-specific information included in spoken utterances. This process can be divided into identification and verification. The task of speaker identification is to determine the identity of an unknown person from a sample of his or her voice. The aim of speaker verification is to decide whether a speaker is whom he claims to be.

The paper is organized in the following way. At first fuzzy nonlinear classifier based on fuzzy If-Then rules and classical Ho-Kashyap procedure is described. Next, description of an automatic speaker recognition system in Matlab environment and applied research procedure are presented. The last section includes summary of achieved results.

2 FHK Classifier

Fuzzy Ho-Kashyap classifier FHK is designed on the basis of the training set, $Tr = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in \mathscr{R}^t$ is a feature vector extracted from a frame of

Adam Dustor
Institute of Electronics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: adam.dustor@polsl.pl

speech, N is the number of vectors and $y_i \in \{-1, +1\}$ indicates the assignment to one of two classes ω_1 or ω_2. After defining the augmented vector $\mathbf{x}'_i = [\mathbf{x}^T_i, 1]^T$ the decision function of the classifier can be defined as

$$g(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}'_i \begin{cases} \geq 0, & \mathbf{x}_i \in \omega_1, \\ < 0, & \mathbf{x}_i \in \omega_2, \end{cases} \tag{1}$$

where $\mathbf{w} = [\widetilde{\mathbf{w}}^T, w_0]^T \in \mathscr{R}^{t+1}$ is a weight vector which must be found during training of the classifier. After multiplying by -1 all patterns from ω_2 class (1) can be rewritten in the form $y_i \mathbf{w}^T \mathbf{x}'_i > 0$ for $i = 1, 2, \ldots, N$. Let $\mathbf{X}$ be the $N \times (t+1)$ matrix

$$\mathbf{X} = \begin{bmatrix} y_1 \mathbf{x}'^T_1 \\ y_2 \mathbf{x}'^T_2 \\ \vdots \\ y_N \mathbf{x}'^T_N \end{bmatrix}, \tag{2}$$

then (1) can be written in the matrix form $\mathbf{Xw} > 0$. To obtain solution $\mathbf{w}$ this inequality is replaced by $\mathbf{Xw} = \mathbf{b}$ where $\mathbf{b} > 0$ is an arbitrary vector called a classifier margin. If data are linearly separable then all components of error vector $\mathbf{e} = \mathbf{Xw} - \mathbf{b}$ are greater than zero and by increasing the respective component of $\mathbf{b}$ (b_p) the value of e_p can be set to zero. If $e_p < 0$ then the pth pattern $\mathbf{x}_p$ is wrongly classified and it is impossible to retain the condition $b_p > 0$ while decreasing b_p. As a result the misclassification error can be written in the form

$$J(\mathbf{w}, \mathbf{b}) = \sum_{i=1}^{N} \mathscr{H}(-e_i), \tag{3}$$

where $\mathscr{H}(\bullet)$ is the unit step pseudo-function, $\mathscr{H}(e_i) = 1$ for $e_i > 0$ and $\mathscr{H}(e_i) = 0$ otherwise. Obtaining solution $\mathbf{w}$ requires minimization of the criterion (3). Unfortunately due to its non-convexity, criterion (3) must be approximated by

$$J(\mathbf{w}, \mathbf{b}) = \sum_{i=1}^{N} |e_i|, \tag{4}$$

or

$$J(\mathbf{w}, \mathbf{b}) = \sum_{i=1}^{N} (e_i)^2. \tag{5}$$

Better approximation of (3) and more robust to outliers is the criterion (4).

Vectors $\mathbf{w}$ and $\mathbf{b}$ are found by minimization the function [3]

$$J(\mathbf{w}, \mathbf{b}) = (\mathbf{Xw} - \mathbf{b})^T \mathbf{D} (\mathbf{Xw} - \mathbf{b}) + \tau \widetilde{\mathbf{w}}^T \widetilde{\mathbf{w}}, \tag{6}$$

where matrix $\mathbf{D} = \mathrm{diag}(d_1, d_2, \ldots, d_N)$ and d_i is the weight corresponding to the ith pattern, which can be interpreted as a reliability attached to this pattern. The second

term of (6) is responsible for the minimization of the complexity of the classifier. The regularization constant $\tau > 0$ controls the trade-off between the classifier complexity and the amount up to which the errors are tolerated. The optimum value of τ is found by cross-validation on the test set.

Differentiation of (6) with respect to $\mathbf{w}$ and $\mathbf{b}$ and setting the results to zero yields the conditions [3]:

$$\begin{cases} \mathbf{w} = (\mathbf{X}^T\mathbf{DX} + \tau\widetilde{\mathbf{I}})^{-1}\mathbf{X}^T\mathbf{Db}, \\ \mathbf{e} = \mathbf{Xw} - \mathbf{b} = \mathbf{0}, \end{cases} \tag{7}$$

where $\widetilde{\mathbf{I}}$ is the identity matrix with the last element on the main diagonal set to zero. Vector $\mathbf{w}$ depends on margin vector $\mathbf{b}$. If pattern lies on the right side of the separating hyperplane then corresponding margin can be increased to obtain zero error. If the pattern is misclassified then the error is negative and decreasing error value is possible only by decreasing the corresponding margin value. One way to prevent $\mathbf{b}$ from converging to zero is to start with $\mathbf{b} > 0$ and to refuse to decrease any of its components. This leads to iterative algorithm for alternately determining $\mathbf{w}$ and $\mathbf{b}$, where components of $\mathbf{b}$ cannot decrease. The vector $\mathbf{w}$ is determined based on the first equation of (7) $\mathbf{w}^{[k]} = (\mathbf{X}^T\mathbf{D}^{[k]}\mathbf{X} + \tau\widetilde{\mathbf{I}})^{-1}\mathbf{X}^T\mathbf{D}^{[k]}\mathbf{b}^{[k]}$, where $[k]$ denotes iteration index. Vector $\mathbf{b}$ is modified only if it results in an increase of its components:

$$\mathbf{b}^{[k+1]} = \mathbf{b}^{[k]} + \eta\left(\mathbf{e}^{[k]} + |\mathbf{e}^{[k]}|\right), \tag{8}$$

where $\eta \in (0,1)$ is a parameter. It should be noted that for $\mathbf{D} = \mathbf{I}$ and $\tau = 0$ the original Ho-Kashyap algorithm is obtained. Since real data have noise and outliers, classifier design method should be robust to them. Unfortunately minimization of squared error (5) leads to non-robust solution. One of the simplest techniques which lead to robustness to noise and outliers is minimization of an absolute error (4), which is very easy to obtain by taking $d_i = 1/|e_i|$ for all $i = 1,2,\ldots,N$. The absolute error minimization procedure for classifier design can be summarized in the following steps [3]:

1. fix $\tau \geq 0$, $0 < \eta < 1$, $\mathbf{D}^{[1]} = \mathbf{I}$, $\mathbf{b}^{[1]} > \mathbf{0}$, $k = 1$;
2. $\mathbf{w}^{[k]} = (\mathbf{X}^T\mathbf{D}^{[k]}\mathbf{X} + \tau\widetilde{\mathbf{I}})^{-1}\mathbf{X}^T\mathbf{D}^{[k]}\mathbf{b}^{[k]}$;
3. $\mathbf{e}^{[k]} = \mathbf{Xw}^{[k]} - \mathbf{b}^{[k]}$;
4. $d_i = 1/|e_i|$ for $i = 1,2,\ldots,N$; $\mathbf{D}^{[k+1]} = \text{diag}(d_1,d_2,\ldots,d_N)$;
5. $\mathbf{b}^{[k+1]} = \mathbf{b}^{[k]} + \eta\left(\mathbf{e}^{[k]} + |\mathbf{e}^{[k]}|\right)$;
6. if $||\mathbf{b}^{[k+1]} - \mathbf{b}^{[k]}||_2 > \xi$ then $k = k+1$ and go to step 2, otherwise stop.

If step 4 is omitted then criterion (5) is minimized. Procedure is convergent to local optimum [3] and leads to the linear discriminant function $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x}'$ that minimizes absolute (or squared) error.

Much better classification results should be obtained for nonlinear classifier constructed using I linear discriminant functions $g_i(\mathbf{x}) = \mathbf{w}^{(i)T}\mathbf{x}'$, $i = 1,2,\ldots,I$. The input space is softly partitioned into I regions. Each function $g_i(\mathbf{x})$ is found by minimization the criterion [3]:

$$J^{(i)}\left(\mathbf{w}^{(i)},\mathbf{b}^{(i)}\right) = \mathbf{A}^T\mathbf{D}^{(i)}\mathbf{A} + \tau\widetilde{\mathbf{w}}^{(i)T}\widetilde{\mathbf{w}}^{(i)}, \tag{9}$$

where $\mathbf{A} = \mathbf{X}\mathbf{w}^{(i)} - \mathbf{b}^{(i)}$ and $\mathbf{D}$ equals

$$\mathbf{D}^{(i)} = \mathrm{diag}\left(\frac{F^{(i)}(\mathbf{x}_1)}{\left|\mathbf{e}_1^{(i)}\right|}, \ldots, \frac{F^{(i)}(\mathbf{x}_N)}{\left|\mathbf{e}_N^{(i)}\right|}\right), \tag{10}$$

and

$$\mathbf{e}_k^{(i)} = \mathbf{w}^{(i)T}\mathbf{x}_k' - b_k^{(i)}. \tag{11}$$

Parameters $F^{(i)}(\mathbf{x}_n)$ denote the membership function of the pattern $\mathbf{x}_n$ from the training set to the ith region and $\mathbf{b}^{(i)}$ is the margin vector for the ith classifier.

The patterns from each class ω_1 and ω_2 are softly partitioned into I regions by the fuzzy c-means algorithm. Each cluster is represented parametrically by the Gaussian membership function with center $\mathbf{c}^{(i)(j)}$:

$$\mathbf{c}^{(i)(j)} = \frac{\sum\limits_{n=1}^{N_j} u_{in}^{(j)}\mathbf{x}_n}{\sum\limits_{n=1}^{N_j} u_{in}^{(j)}}, \tag{12}$$

and dispersion $\mathbf{s}^{(i)(j)}$:

$$\mathbf{s}^{(i)(j)} = \frac{\sum\limits_{n=1}^{N_j} u_{in}^{(j)}\left[\mathbf{x}_n - \mathbf{c}^{(i)(j)}\right]^{(\bullet 2)}}{\sum\limits_{n=1}^{N_j} u_{in}^{(j)}}, \tag{13}$$

where $i = 1, 2, \ldots, I$ is a cluster index and $j \in \{1, 2\}$ is a class index, $(\bullet 2)$ denotes component by component squaring, N_j number of patterns from class ω_j and $u_{in}^{(1)}$, $u_{in}^{(2)}$ are elements of fuzzy partition matrices for class ω_1, ω_2 obtained by fuzzy c-means algorithm.

Next, I nearest pairs of clusters belonging to different classes are found. As a distance measure between centers (prototypes) the L_1 norm is used. At first, the nearest pair of clusters from ω_1 and ω_2 is found. This pair is used to construct the first linear decision function $g_1(\mathbf{x}) = \mathbf{w}^{(1)T}\mathbf{x}'$. Subsequently, after exclusion of found clusters from the set of clusters, the next nearest pair is found. This pair defines the second function $g_2(\mathbf{x}) = \mathbf{w}^{(2)T}\mathbf{x}'$. The procedure ends, when the last pair of clusters is found. Finally I nearest pairs of clusters are found. Each pair is defined by four parameters $\mathbf{c}^{(i)(1)}$, $\mathbf{s}^{(i)(1)}$, $\mathbf{c}^{(i)(2)}$, $\mathbf{s}^{(i)(2)}$.

The membership function $F^{(i)}(\mathbf{x}_n)$ is calculated using the algebraic product as the t-norm and the maximum operator as the s-norm [3]:

$$F^{(i)}(\mathbf{x}) = \max\left(\exp\left(-\sum_{j=1}^{t}\frac{\left(x_j - c_j^{(i)(1)}\right)^2}{2s_j^{(i)(1)}}\right), \exp\left(-\sum_{j=1}^{t}\frac{\left(x_j - c_j^{(i)(2)}\right)^2}{2s_j^{(i)(2)}}\right)\right). \quad (14)$$

The final decision of the nonlinear classifier for the pattern $\mathbf{x}$ is obtained by the weighted average:

$$g(\mathbf{x}) = \frac{\sum\limits_{i=1}^{I} F^{(i)}(\mathbf{x})\mathbf{w}^{(i)T}\mathbf{x}'}{\sum\limits_{i=1}^{c} F^{(i)}(\mathbf{x})}. \quad (15)$$

This classifier can be named a mixture-of-experts classifier and works in a similar way as a Takagi-Sugeno-Kang fuzzy inference system. Unfortunately this means that none of the fuzzy consequents can be applied to decision process. This problem can be solved substituting moving singletons for fuzzy moving consequents. Position of the fuzzy sets in consequents of the If-Then rules depends on the input crisp values. As a result nonlinear discriminative function is obtained

$$g(\mathbf{x}) = \frac{\sum\limits_{i=1}^{I} \mathscr{G}\left(F^{(i)}(\mathbf{x}), w^{(i)}\right)\mathbf{w}^{(i)T}\mathbf{x}'}{\sum\limits_{i=1}^{I} \mathscr{G}\left(F^{(i)}(\mathbf{x}), w^{(i)}\right)}, \quad (16)$$

where $\mathscr{G}\left(F^{(i)}(\mathbf{x}), w^{(i)}\right)$ depends on applied fuzzy consequent [2] and for the simplicity $w^{(i)} = 1$. Each linear discriminative function $g_i(\mathbf{x})$ is found by minimization criterion (9) where $\mathbf{D}$ is given by the modified (10) ($\mathscr{G}\left(F^{(i)}(\mathbf{x}), w^{(i)}\right)$ instead of $F^{(i)}(\mathbf{x}_n)$).

Summarizing, the training procedure of this classifier denoted as a FHK (Fuzzy Ho-Kashyap) consists of the following steps:

1. fix type of fuzzy consequents – function $\mathscr{G}$ [2];
2. fix number of If-Then rules I and $w^{(i)} = 1$ for $i = 1, 2, \ldots, I$;
3. fuzzy c-means clustering of data belonging to ω_1 and ω_2; compute parameters $\mathbf{c}^{(i)(1)}$, $\mathbf{s}^{(i)(1)}$, $\mathbf{c}^{(i)(2)}$, $\mathbf{s}^{(i)(2)}$ for $i = 1, 2, \ldots, I$ – (12) and (13);
4. find I nearest pairs of clusters, each pair is defined by $\mathbf{c}^{(k)(1)}$, $\mathbf{s}^{(k)(1)}$ and $\mathbf{c}^{(n)(2)}$, $\mathbf{s}^{(n)(2)}$ for $k, n = 1, 2, \ldots, I$;
5. compute $F^{(i)}(\mathbf{x}_k)$ (14) and $\mathscr{G}\left(F^{(i)}(\mathbf{x}_k), w^{(i)}\right)$ for $k = 1, 2, \ldots, N$ and $i = 1, 2, \ldots, I$;
6. fix regularization constant τ;
7. train classifiers $g_i(\mathbf{x}) = \mathbf{w}^{(i)T}\mathbf{x}'$, $i = 1, \ldots, I$ in accordance with absolute error minimization procedure for the given τ and $\mathbf{D}^{(i)}$.

Speaker is represented in a speaker recognition system by the parameters $\mathbf{w}^{(i)}$ and $\mathbf{c}^{(i)(1)}$, $\mathbf{s}^{(i)(1)}$, $\mathbf{c}^{(i)(2)}$, $\mathbf{s}^{(i)(2)}$ for $i = 1, 2, \ldots, I$.

3   Matlab Based Recognition

In order to test described classifier Matlab application was written. Before feature extraction silence was removed from speech files. Voice activity detection was based on the energy of the signal. After silence removing speech was pre-emphasized with a parameter of $\alpha = 0.95$ and segmented into 20 ms frames every 10 ms. Hamming windowing was applied. For each frame 12th order LPC analysis [5] was applied to obtain LPC parameters which were then transformed into 18th order LPCC coefficients [5].

There were 10 speaker models. Each model was trained with approximately 5 s of speech after silence removing. All training utterances came from set Z3 of ROBOT [1] corpus. Text dependent recognition was implemented. Set Z4 was used to test described classifier. The test utterances came from 20 speakers of which 10
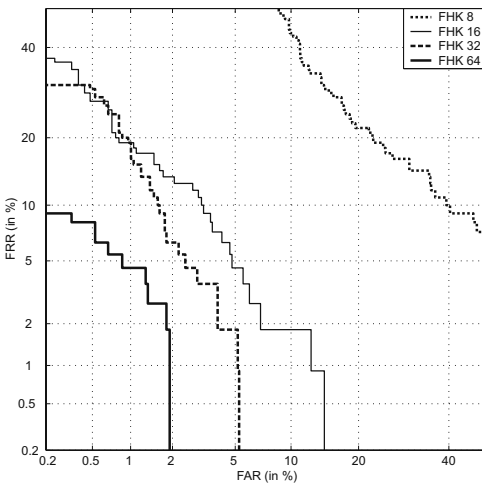
Fig. 1 Speaker verification for FHK classifier – Lukasiewicz consequent

Table 1 EER (in %) for classical models

Model	EER	Model	EER
VQ 8	25.45	GMM 8 diag	16.29
VQ 16	17.27	GMM 16 diag	11.65
VQ 32	12.73	GMM 32 diag	12.82
VQ 64	10.07	GMM-LR 2 diag	10
NN	8.18	GMM-LR 4 diag	10
GMM 2 full	7.32	GMM-LR 8 diag	8.37
GMM 4 full	10.91	GMM-LR 16 diag	7.27
GMM-LR 2 full	8.18	GMM-LR 32 diag	10
GMM-LR 4 full	10		
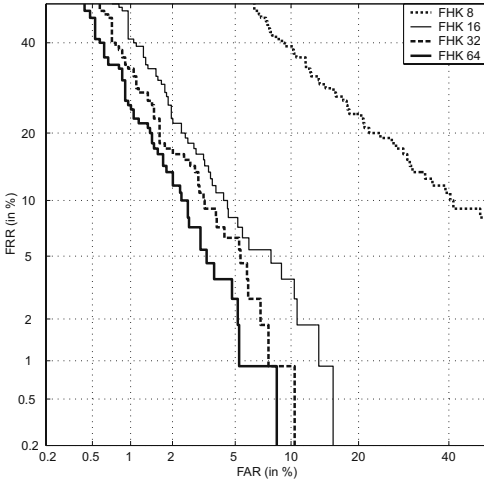
Fig. 2 Speaker verification for FHK classifier – Rescher consequent
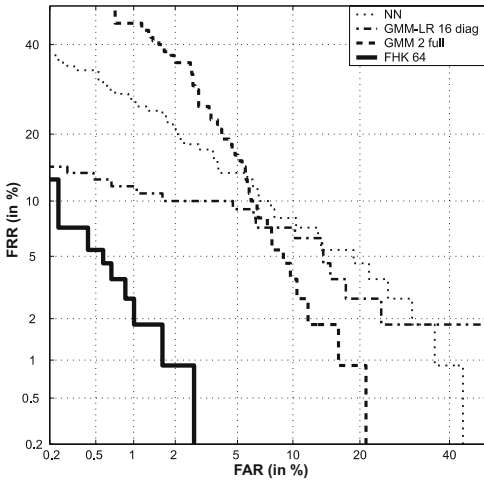
Fig. 3 Comparison of the best classical and new approaches to speaker modeling

were unseen for the system during its training. This procedure enables to obtain more realistic results when most of the speakers are not previously registered in the system and represent impostors. Each speaker provided 11 test sequences of approximately 5 s each. There were 110 (10×11) genuine attempts (possible false rejection error), 990 ($9 \times 11 \times 10$) impostor attempts (possible false acceptance error, speaker has its own model in a system – seen impostor) and 1100 ($10 \times 11 \times 10$) impostor tests (possible false acceptance error, speaker does not have model in a

Table 2 EER (in %) for the FHK classifier

Consequent	Order of the model			
	8	16	32	64
Lukasiewicz	21.82	4.67	3.64	1.82
Fodor	21.82	4.35	3.64	1.82
Reichenbach	24.52	3.56	2.85	1.82
Kleene-Dienes	24.71	3.64	3.61	1.72
Zadeh	27.11	6.27	3.8	1.82
Goguen	20.91	16.32	12.92	13.71
Gödel	26.36	16.36	14.35	11.91
Rescher	21	6.17	5.41	3.68

system – unseen impostor). As a result the overall number of tests was 2200 for each recognition method.

Verification performance was characterized in terms of two error measures: the false acceptance rate *FAR* and false rejection rate *FRR*. These correspond respectively to the probability of acceptance an impostor as a valid user or rejection of a valid user. These two measures calculated for varying decision thresholds were used to plot DET curves [4]. Another very useful performance measure is an equal error rate *EER* corresponding to decision threshold for which *FRR* equals *FAR*. The performance of identification was characterized in terms of identification rate.

For each speaker GMM model [6] was obtained (both full and diagonal covariance matrixes). The 'world' model for score normalization was trained using speech only from 10 speakers, the rest was unseen for the system. GMM of orders 2, 4, 8, 16, 32 were obtained (diagonal version) and 2, 4 for full versions. The performance of vector quantization VQ techniques was also checked, LBG procedure was applied to obtain speaker codebooks [5]. Number of codevectors per speaker was 2, 4, 8, 16, 32, and 64. Nearest neighbor NN algorithm was also tested (speaker model consisted of all training vectors).

In order to obtain FHK models, LBG algorithm was used to reduce cardinality of the training set. Classifier 'one from many' was applied (classifier discriminates one class from all other classes). For example, abbreviation FHK 8 means that the training set consisted of 8 codevectors belonging to recognized speaker (class ω_1) and 72 (9×8) codevectors of all other speakers (class ω_2). FHK models were obtained for 8, 16, 32, and 64 codevectors per speaker. Optimum number of fuzzy rules (linear classifiers) $I = 2, 3, \ldots, 16$ and regularization constant $\tau = 0.1, 0.2, \ldots, 2$ which yielded the lowest error rate were found by the crossvalidation. Initial parameters of the FHK procedure were the following: $\mathbf{b}^{[1]} = 10^{-6}$ and $\eta = 0.99$. The iterations were stopped as soon as the Euclidean norm in a successive pair of $\mathbf{b}$ was less than 10^{-4}. Speaker verification performance was examined for the following fuzzy implications: Lukasiewicz, Fodor, Reichenbach, Kleene-Dienes, Zadeh, Goguen, Gödel and Rescher.

4   Conclusions

In speaker verification decision of the classifier whether the utterance belongs to a given speaker or not is based on the whole utterance. This means that speaker verification is a different problem from a typical pattern classification task, where the decision about membership of the single vector to one of the two classes needs to be done. Despite of this, the FHK 64 classifier with Kleene-Dienes consequent achieved very low value of $EER = 1.72\%$. Comparing this result to results achieved for other classical methods ($EER = 7.27\%$) it can be concluded from Fig. 3 that achieved performance of FHK is a very good result and indicates that fuzzy approach to speaker recognition is very promising.

References

1. Adamczyk, B., Adamczyk, K., Trawiński, K.: Zasób mowy ROBOT. Biuletyn Instytutu Automatyki i Robotyki WAT 12, 179–192 (2000)
2. Czogała, E., Łęski, J.: Fuzzy and neuro-fuzzy intelligent systems. Physica-Verlag, Heidelberg (2000)
3. Łęski, J.: A fuzzy if-then rule based nonlinear classifier. International Journal of Applied Mathematics and Computer Science 13(2), 215–223 (2003)
4. Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M.: The DET curve in assesment of detection task performance. In: Proceedings of the 5th European Conference on Speech Communication and Technology, Rhodes, Greece, pp. 1895–1898 (1997)
5. Rabiner, L.R., Juang, B.H.: Fundamentals of speech recognition. Prentice-Hall, Englewood Cliffs (1993)
6. Reynolds, D.A.: Speaker identification and verification using Gaussian mixture speaker models. Speech Communication 17(1-2), 91–108 (1995)

Support Vector Classifier
with Linguistic Interpretation
of the Kernel Matrix in Speaker Verification

Mariusz Bąk

Abstract. The paper shows that support vector classifier with linguistic interpretation of the kernel matrix can be effectively used in speaker verification. The kernel matrix is obtained by means of fuzzy clustering, based on global learning of fuzzy system with logical interpretation of if-then rules and with parametric conclusions. The kernel matrix is data-dependent and may be interpreted in terms of linguistic values related to the premises of if-then rules. Simulation results obtained for SPIDRE corpus are presented for comparison with traditional methods used in speaker verification.

Keywords: support vector machine, kernel matrix, fuzzy system, speaker verification.

1 Introduction

The support vector machine was the first proposed kernel-based method [1]. It uses a kernel function to transform data from input space into a high-dimensional (possibly infinite-dimensional) feature space in which it searches for a separating hyperplane. Polynomial, Gaussian, sigmoidal and multiquadratic kernel functions are commonly used to transform input space into desired feature space. The SVM aims to maximise the generalisation ability, which depends on the empirical risk and the complexity of the machine [9]. The SVM can be applied to various classification and regression problems (support vector classifier – SVC and support vector regression – SVR, respectively).

Mariusz Bąk
Institute of Electronics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: mariusz.bak@polsl.pl

The idea of linguistic interpretation of the kernel matrix was first presented in [5] for support vector regression. The minimisation problems for global ε-insensitive learning of a fuzzy system and of a SVR machine were compared showing that learning of a fuzzy system can be presented as learning of a SVR machine with a specific kernel matrix. Because the kernel matrix is obtained from fuzzy clustering it can be interpreted in terms of linguistic values based on the premises of if-then rules of a fuzzy system.

This paper presents the possibility of using linguistic interpretation of kernel matrix in speaker verification. A fuzzy system with a logical interpretation of if-then rules is learned by solving minimisation problem for SVC machine with a specific kernel matrix. Performance of resulting automatic speaker recognition system is compared with performance of a Gaussian Mixture Model Likelihood Ratio (GMM-LR) system [8].

2 Fuzzy System with a Logical Interpretation of If-Then Rules

Let us assume that I fuzzy if-then rules with t singelton inputs $x_{01}, \ldots, x_{0t}$ and single output Y are given. The ith rule can be written in form [3]:

$$\mathfrak{R}^{(i)} : \text{ IF } \mathbf{x}_0 \text{ IS } \mathbf{A}^{(i)} \text{ THEN } Y \text{ IS } B^{(i)}(\mathbf{x}_0) , \tag{1}$$

where $\mathbf{x}_0 = [x_{01}, \ldots, x_{0t}]^\top \in \mathbb{R}^t$, Y is a linguistic variable and $\mathbf{A}^{(i)}$ and $\mathbf{B}^{(i)}(\mathbf{x}_0)$ are linguistic values. A single rule in (1) may be called a fuzzy if-then rule with a moving consequent.

In case of logical interpretation of the if-then rules the inferred output fuzzy set for the ith rule may be presented as [3]:

$$\mu_{B'^{(i)}}(y, \mathbf{x}_0) = I(\mu_{\mathbf{A}^{(i)}}(\mathbf{x}_0), \mu_{B^{(i)}}(y, \mathbf{x}_0)) , \tag{2}$$

where $I(\bullet, \bullet)$ is the selected fuzzy implication.

A crisp output value may be obtained from Modified Indexed Center of Gravity (MICOG) as [3]:

$$y_0(\mathbf{x}_0) = \frac{\sum_{i=1}^{I} y^{(i)}(\mathbf{x}_0) \text{Area}(\mu_B^{*(i)}(y, \mathbf{x}_0))}{\sum_{k=1}^{I} \text{Area}(\mu_B^{*(k)}(y, \mathbf{x}_0))} , \tag{3}$$

where $B^{*(i)}$ results from $B^{(i)}$ after removing its non-informative part.

Usually it is assumed that location of fuzzy set in conclusion of ith if-then rule may be expressed as linear combinations of inputs:

$$y^{(i)}(\mathbf{x}_0) = \mathbf{p}^{(i)\top} \mathbf{x}_0' , \tag{4}$$

where $\mathbf{p}^{(i)}$ denotes the parameter vector with bias element and $\mathbf{x}_0'$ denotes an extended input vector $\mathbf{x}_0' = [1, \mathbf{x}_0^\top]^\top$.

If we assume that consequents $B^{(i)}$ have isosceles triangle membership functions with the triangle base of width $w^{(i)}$ then a crisp value of the output can be expressed as [3]:

$$y_0(\mathbf{x}_0) = \sum_{i=1}^{I} S^{(i)}(\mathbf{x}_0)\mathbf{p}^{(i)\top}\mathbf{x}_0' , \qquad (5)$$

where $S^{(i)}(\mathbf{x}_0)$ can be called a normalized interaction degree of a datum $\mathbf{x}_0$ for the conclusion of the ith rule [5] and can be written in the form:

$$S^{(i)}(\mathbf{x}_0) = \frac{\mathscr{G}(\mu_{\mathbf{A}^{(i)}}(\mathbf{x}_0), w^{(i)})}{\sum_{k=1}^{I} \mathscr{G}(\mu_{\mathbf{A}^{(k)}}(\mathbf{x}_0), w^{(k)})} , \qquad (6)$$

where function $\mathscr{G}$ depends on the selected fuzzy implication $I(\bullet,\bullet)$. Then, (5) can be written as:

$$y_0(\mathbf{x}_0) = \mathbf{P}^{\top}\mathbf{d}(\mathbf{x}_0) , \qquad (7)$$

where

$$\mathbf{d}(\mathbf{x}_0) = \left[S^{(1)}(\mathbf{x}_0)\mathbf{x}_0'^{\top}, \ldots, S^{(I)}(\mathbf{x}_0)\mathbf{x}_0'^{\top} \right]^{\top} , \qquad (8)$$

$$\mathbf{P} = \left[\mathbf{p}^{(1)\top}, \ldots, \mathbf{p}^{(I)\top} \right]^{\top} . \qquad (9)$$

Linguistic values $\mathbf{A}^{(i)}$ in antecedents of if-then rules can be found by means of clustering, namely fuzzy c-means algorithm. The membership degrees u_{in} can be transformed into Gaussian membership functions by calculating their centers $\mathbf{c}^{(i)} = [c_1^{(i)}, \ldots, c_t^{(i)}]$ and dispersions $\mathbf{s}^{(i)} = [s_1^{(i)}, \ldots, s_t^{(i)}]$ as:

$$c_j^{(i)} = \frac{\sum_{n=1}^{N} u_{in} x_{nj}}{\sum_{n=1}^{N} u_{in}}, \qquad (10)$$

and

$$\left(s_j^{(i)}\right)^2 = \frac{\sum_{n=1}^{N} u_{in}\left(x_{nj} - c_j^{(i)}\right)^2}{\sum_{n=1}^{N} u_{in}}. \qquad (11)$$

It should also be mentioned that the Takagi-Sugeno-Kang fuzzy system and the Mamdani fuzzy system can be obtained from the presented system if specific fuzzy sets in conclusions of if-then rules and specific fuzzy implications are selected.

3 Support Vector Classification with Linguistic Interpretation of Kernel Matrix

It was shown in [5] that ε-insensitive global learning of the fuzzy system presented in previous section can be presented as learning of a support vector regression machine with a specific kernel matrix if following conditions are met:

1. the kernel matrix has the form $\mathscr{K} = \left[\widetilde{\mathbf{d}}(\mathbf{x}_n)^\top \widetilde{\mathbf{d}}(\mathbf{x}_j)\right]_{n,j=1}^N$,
2. the regularization parameter has the value $C = 1/\tau$,

where $\widetilde{\mathbf{d}}$ is a narrowed version of (8) with $\mathbf{x}_n$ replacing $\mathbf{x}'_n$, $\mathbf{x}_n$ and $\mathbf{x}_j$ denote input vectors from a training set $\mathscr{T}$, N is the cardinality of $\mathscr{T}$, C is the regularisation parameter for SVR machine and $\tau > 0$ controls the trade-off between the complexity of regression model and the degree of error tolerance for a fuzzy system. The kernel matrix may be interpreted in terms of linguistic values $\mathbf{A}^{(i)}$ present in the premises of if-then rules of a fuzzy system and therefore is said to have linguistic interpretation. The linguistic values are obtained by clustering, namely by fuzzy c-means clustering of a training $\mathscr{T}$.

Similar approach may be proposed in case of support vector classification if $\varepsilon = 0$. Learning of a fuzzy system for classification for training set $\mathscr{T}$ consisting of subsets of positive input vectors $\mathscr{T}^{(+1)}$ and negative input vectors $\mathscr{T}^{(-1)}$ (where labels are equal to $+1$ and -1, respectively) by using support vector classifier with linguistic interpretation of the kernel matrix may be divided into following stages:

1. Select the implication for interpreting fuzzy if-then rules and related $\mathscr{G}$ function.
2. Set the number of if-then rules I, the widths $w^{(i)}$ of the bases of triangle membership functions of linguistic values $B^{(i)}$ in conclusions of if-then rules, the regularisation parameter τ.
3. Perform clustering of subsets $\mathscr{T}^{(+1)}$ and $\mathscr{T}^{(-1)}$ using fuzzy c-means algorithm to obtain membership degree u_{in} for each input vector $\mathbf{x}_n \in \mathscr{T}$ and each if-then rule.
4. Transform membership degrees u_{in} of input vectors resulting from fuzzy clustering to obtain centers $\mathbf{c}^{(i)(+1)}, \mathbf{c}^{(i)(-1)}$ and dispersions $\mathbf{s}^{(i)(+1)}, \mathbf{s}^{(i)(-1)}$ of Gaussian membership functions. Group Gaussian distributions for positive and negative subset of $\mathscr{T}$ into pairs by selecting Gaussian distributions which the closest centers $\mathbf{c}^{(i)(+1)}, \mathbf{c}^{(i)(-1)}$.
5. Calculate the normalized interaction degree $S^{(i)}(\mathbf{x}_n)$ for each datum $\mathbf{x}_n$ from $\mathscr{T}$ and each if-then rule using selected $\mathscr{G}$ function.
6. Calculate the kernel matrix as

$$\mathscr{K} = \left[\widetilde{\mathbf{d}}(\mathbf{x}_n)^\top \widetilde{\mathbf{d}}(\mathbf{x}_j)\right]_{n,j=1}^N = \left[\sum_{i=1}^I S^{(i)}(\mathbf{x}_n) S^{(i)}(\mathbf{x}_j) \mathbf{x}_n^\top \mathbf{x}_j\right]_{n,j=1}^N. \tag{12}$$

7. Learn the SVC machine with kernel matrix $\mathscr{K}$ and parameter $C = 1/\tau$.
8. Calculate parameters of a fuzzy system using following formulas

$$\widetilde{\mathbf{P}} = \sum_{i \in I_{\mathbf{SV}}} \alpha_i \widetilde{\mathbf{d}}(\mathbf{x}_i) y_i, \tag{13}$$

$$b = \frac{1}{\overline{I_{\text{SV}}}} \sum_{i \in I_{\text{SV}}} y_i \widetilde{\mathbf{d}}(\mathbf{x}_i)^\top \widetilde{\mathbf{P}}, \tag{14}$$

where I_{SV} is the set of support vectors for which Lagrange multipliers obtained from learning the SVC machine $\alpha_i > 0$.

The crisp value of the output of the resulting fuzzy system for any input vector $\mathbf{x}$ can be expressed as:

$$y^{(i)}(\mathbf{x}) = \widetilde{\mathbf{d}}(\mathbf{x})^\top \widetilde{\mathbf{P}} + b, \tag{15}$$

and its computation requires storing centers $\mathbf{c}^{(i)(+1)}, \mathbf{c}^{(i)(-1)}$ and dispersions $\mathbf{s}^{(i)(+1)}$, $\mathbf{s}^{(i)(-1)}$ of Gaussian membership functions, the widths $w^{(i)}$ of the bases of triangle memberhip functions of linguistic values $B^{(i)}$ in conclusions of if-then rules and parameters of crisp output value of the fuzzy system $\widetilde{\mathbf{P}}$ and b.

4   Simulations

The SPIDRE 1996 corpus (a subset of the Switchboard corpus dedicated for speaker verification and identification) of English telephone speech was used to evaluate the performance of and robustness of the SVC with linguistic interpretation of the kernel matrix in speaker verification. The simulations were performed in MATLAB environment and the SVC with linguistic kernel function was simulated with modified version of LIBSVM software [2, 4]. The resulting Detection Error Trade-off (DET) curve [6] plots are presented and Equal Error Rate (EER) is calculated for each system. Each curve represents performace of the speaker verification system for varying value of decision threshold.

4.1   Speech Corpus

The SPIDRE corpus included 266 telephone conversations, each about 4–5 minutes long. There are 317 speakers are, including 44 target speakers, 113 non-target A speakers (non-target speakers taking part in at least one conversation with any target speaker) and 160 non-target B speakers (remaining non-target speakers). The quality of the speech signal is poor and depends strongly on the handset used. Sampling frequency is equal to 8 kHz and μ-law PCM encoding is used.

For each target speaker at least four conversations are included in corpus, in two of which the same handset is used by the speaker. Utterances from one of these two conversations was used to obtain the positive subset of the learning sets in case of the SVM-based method or the whole learning set in case of GMM-based method. The other utterance was used to test the performance of the obtained ASR system when same handset is used for learning in testing. The utterances from the remaining conversations were used to test the performance of the ASR system in case of different handsets being used during learning and testing phase. The utterances of non-target A speakers were used to obtain the negative subset of the learning sets for SVM-based method or the learning set representing the background model for GMM-based method. Non-target B speakers serve as impostors and their

utterances are used to evaluate the possibility of false acceptance of unauthorised speaker.

4.2 Front-End Processing and Feature Extraction

Front-end processing of speech data from SPIDRE corpus encompassed splitting each conversations into separate utterances, volume normalisation and pre-emphasis. Later speech signal was split into frames, each containing 160 samples (20 miliseconds long, with 10 miliseconds overlap). The Hamming window was applied to each frame. Subsequently, silent fragments of utterance were omitted, basing on the energy of the speech signal in each frame.

During the feature extraction, the Linear Predictive Coding (LPC) analysis [7] was performed and 13 LPC coefficients were obtained, using autocorrelation method to derive the set of equations and solving it in accordance with Durbin iterative algorithm. Obtained LPC coefficients provided the grounds for calculating 19 coefficients of LPC-based cepstrum. After droping the coefficient corresponding with the gain of each frame, the remaining 18 coefficients constituted the feature vectors.

4.3 Classification

The fuzzy system presented in this paper was compared with a GMM-LR system. The number of fuzzy rules was set to 8, while for the GMM-LR system 8 Gaussian distributions were used in each target speaker model and 64 Gaussian distributions were used in background model. The value of τ was changing from 100 to 0.01.

4.4 Results

Resulting DET curves for the same handset being used during learning and testing phase are presented in Fig. 1. The performance for high values of τ is significantly worse for s fuzzy system than for a GMM-LR system, but as τ reaches values of 0.1 and 0.01 the gap narrows (EER = 10.66% for a GMM-LR system and EER = 11.57% for a fuzzy system with $\tau = 0.01$). As the same value of τ is selected for all target speakers, the performance of a fuzzy system could be improved by choosing different values of τ for different target speakers which can be done by utilising cross-validation scheme.

Resulting DET curves for different handsets being used during learning and testing phase are presented in Fig. 2. In this case the difference in performance of two simulated systems is significant. The EER for higher values of τ is again higher for a fuzzy system. As τ takes values of 0.1 and 0.01 the gap becomes smaller (EER = 25.66% for a GMM-LR system and EER = 31.02% for a fuzzy system with $\tau = 0.01$).
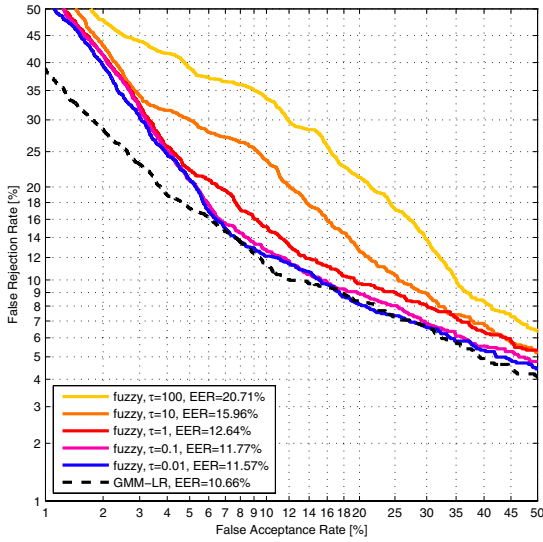
Fig. 1 DET curve plot for the same handset being used during learning and testing phase
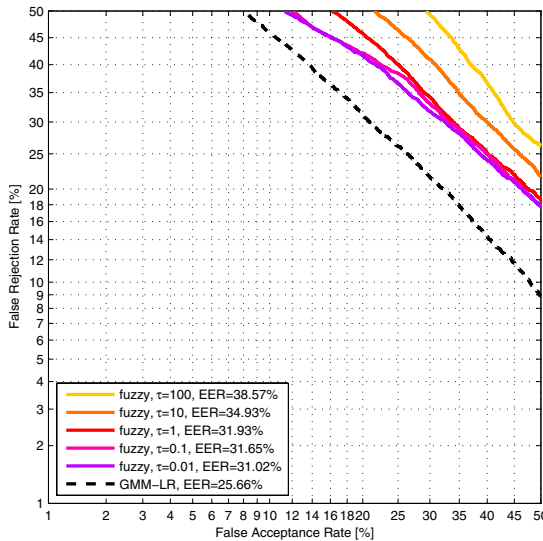

Fig. 2 DET curve plot for different handsets being used during learning and testing phase

5 Conclusions

The fuzzy system presented in this paper is obtained by learning SVC machine
with a specific kernel matrix which can be interpreted in terms of linguistic values.

The performance of obtained fuzzy system is comparable with statistical GMM-LR models if the same handset is used during learning and testing phase. Furthermore, feature space resulting from using proposed kernel is of finite dimensionality and normal vector of separating hyperspace can be used explicitly in decision function to speed up score calculation.

References

1. Boser, B.E., Guyon, I.M., Vapnik, V.: A training algorithm for optimal margin classifier. In: Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, Pittsburgh, US, pp. 144–152 (1992)
2. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
3. Czogała, E., Łęski, J.: Fuzzy and Neuro-fuzzy Intelligent Systems. Physica-Verlag, Heidelberg (2000)
4. Fan, R.E., Chen, P.H., Lin, C.J.: Working set selection using second order information for training SVM. Journal of Machine Learning Research 6, 1889–1918 (2005)
5. Łęski, J.: On support vector regression machines with linguistic interpretation of the kernal matrix. Fuzzy Sets and Systems 157, 1092–1113 (2006)
6. Martin, A., Doddington, F., Kamm, T., Ordowski, M., Przybocki, M.: The DET curve in assesment of detection task performance. In: Proceedings of the 5th European Conference on Speech Communication and Technology, pp. 1895–1898 (1997)
7. Rabiner, L.R., Schafer, R.W.: Digital processing of speech signals. Prentice Hall, Englewood Cliffs (1978)
8. Reynolds, D.A.: Speaker identification and verification using gaussian mixture speaker models. Speech Communication 17(1-2), 91–108 (1995)
9. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)

Application of Discriminant Analysis to Distinction of Musical Instruments on the Basis of Selected Sound Parameters

Alicja Wieczorkowska and Agnieszka Kubik-Komar

Abstract. The goal of the presented research was to recognize musical instruments in sound mixes for various levels of accompanying sounds, on the basis of a limited number of sound parameters. Discriminant analysis was used for this purpose. Reduction of the initial large set of sound parameters was performed by means of PCA (principal components analysis), and the factors found using PCA were utilized as input data for discriminant analysis. The results of the discriminant analysis allowed us to assess the accuracy of linear classification on the basis the factors found, and conclude about sound parameters of the highest discriminant power.

Keywords: discriminant analysis, PCA, audio recognition, linear classification.

1 Introduction

Everyday use of computers and ubiquitous portable devices became inherent part of our life last years. We communicate by means of phones and computers, with access to the Internet available from cell phones and PDAs, Internet phones connected to the web, we exchange of photos, movies, and music. Multimedia repositories became very popular, and they keep growing in size, popularity and availability. With this huge amount of audio-visual data, the possibility to browse them by content becomes very desirable. Therefore, a lot of research effort is put into automatic indexing and browsing of audio and visual data. Indexing of

Alicja Wieczorkowska
Polish-Japanese Institute of Information Technology,
Koszykowa 86, 02-008 Warsaw, Poland
e-mail: alicja@pjwstk.edu.pl

Agnieszka Kubik-Komar
University of Life Sciences in Lublin,
Akademicka 13, 20-950 Lublin, Poland
e-mail: agnieszka.kubik@up.lublin.pl

multimedia data for content-based search brought about the creation of the MPEG-7 standard [4], to facilitate our needs for automatic browsing of multimedia repositories. We would like to find photos of objects or people, and favorite pieces of music. As far as audio is concerned, the user may find the title and the performer of a piece of music on the basis of the excerpt given, and look for a specific tune using query-by-humming systems [6]. Pitch-tracking algorithms may assist extracting scores for a melody. However, automatic music transcription for polyphonic data, for multiple instruments playing together, or finding excerpts played by a specified instrument, is still a challenge. Recognition of sound timbre, i.e. identification of an instrument in sound mix, is difficult, especially when harmonic partials in their spectra overlap. In this paper, we address the problem of automatic recognition of an instrument dominating in sound mixes, using same-pitch sounds, posing the biggest challenge for any classifier because spectra overlap to high extend.

Basically, audio data represent long sequences of samples and parameterization is needed before classifiers are applied. Since there is no standard sound parameterization that fits all automatic classification purposes, quite numerous sound parameters are used. In our research, we focus on assessing descriptive power of various sound parameters, in order to find the most discriminative ones.

Our initial feature vector, i.e., set of sound parameters calculated for audio data, consisted of 217 parameters. Factorial analysis in form of PCA (principal components analysis) was applied in order to limit the number of these sound parameters. Performing PCA as preprocessing allowed us to avoid problems in case the data did not meet requirements needed to perform discriminant analysis, i.e., linear independence and multivariate normal distribution of the data, as well as homogeneity of variance in classes determined by the discriminant vector [8]. PCA reduces the number of the examined variables (i.e. sound parameters) by transforming them to a smaller number of new, non-correlated factors. As a result of this analysis, we obtain so called factors' loadings, representing correlation coefficients between our initial parameters and new factors, as well as factors scores. These data became a basis for further analysis.

The application of discriminant analysis aimed at answering two questions: whether the obtained factors allow good discernment of the investigated instruments, and which factors show the highest discriminant power. The answer to the first question can be presented in the form of classification matrix, containing information about the number and percentage of correctly classified objects for each instrument. The modulus of standardized coefficient of the first discriminant function allows determining 'discriminant power', or importance of factors [5], [10]. Additionally, the tables of squared Mahalanobis distances D^2, together with F-test, allow estimating the statistical importance of these values. Because of limitation of space only illustrative examples of the analyses performed will be presented in this paper; F-test results are not shown (they proved that all distances between instruments are statistically significant at 0.01 significance level).

The paper is organized as follows. In the next section, we provide additional background information about automatic recognition of instrument from audio data, as well as sound parameters used in the feature vector in our research. Next, we

describe audio data used in our experiments. In Sect. 4, experiments are described, and the last section concludes the paper.

2 Background

Automatic recognition of musical instruments from audio recordings has been investigated last years by various researchers. Sound parameterization for this purpose includes features describing Fourier transform based spectrum, wavelet analysis coefficients, MFCC (Mel-Frequency Cepstral Coefficients), MSA (Multidimensional Analysis Scaling) trajectories, also MPEG-7 audio descriptors [4], and so on. On the basis of the feature set obtained through parameterization, classification can be performed. Methods used for this purpose so far include artificial neural networks, k-nearest neighbors, rough set based algorithms, support vector machines, and so on. Very often, isolated sounds are used in classification experiments, for various sets of audio data; broad review of classification methods used, as well as parameterization applied and instruments chosen for research, can be found in [3]. The results can be difficult to compare because of different data sizes; generally, for more instruments the accuracy obtained is lower, reaching up to 100% for a few instruments, and going down to about 70% for more than 10 instruments. Still, the case of isolated sounds is relatively easy, and recognition of instruments is more difficult for polyphonic music, when more than one sound is played at the same time. But such recognition is vital as a step towards automatic score transcription, to facilitate extraction of scores from audio recordings, and such research is also recently performed, see, e.g., [1, 2].

In this paper, we decided to use the set of features that was already used in similar research. Most of these features represent average value of attributes calculated for consecutive frames of the entire analyzed sound, with Hamming window used for spectrum calculation, long (120 ms) analyzing frame in case of presence of the lowest audible sounds in the mix, and 2/3 overlap of the consecutive frames. The following 217 sound descriptors were used [12]:

- MPEG-7 audio main descriptors [4]:

  - *AudioSpectrumSpread* averaged for the entire sound – a RMS value of the deviation of the Log frequency power spectrum with respect to the gravity center in each analyzed frame;
  - *AudioSpectrumFlatness*: $flat_1, \ldots, flat_{25}$ – vector describing the flatness property of the power spectrum within a frequency bin for selected bins, averaged for the entire sound; 25 out of 32 frequency bands were used for each frame;
  - *AudioSpectrumCentroid* – power weighted average of the frequency bins in the power spectrum of all the frames in a sound segment, calculated with a Welch method;
  - *AudioSpectrumBasis*: $basis_1, \ldots, basis_{165}$; parameters calculated for the spectrum basis functions, used to reduce the dimensionality by projecting the spectrum (for each frame) from high dimensional space to low dimensional space

with compact salient statistical information. The spectral basis descriptor is a series of basis functions derived from the Singular Value Decomposition (SVD) of a normalized power spectrum. The total number of sub-spaces in basis function in our case is 33, and for each sub-space, minimum/maximum/mean/distance/standard deviation are extracted. Distance is calculated as the summation of dissimilarity (absolute difference of values) of every pair of coordinates in the vector. The calculated values were averaged over all analyzed frames of the sound;

- other descriptors:

  - *Energy* – average energy of spectrum of the parameterized sound;
  - MFCC – min, max, mean, distance, and standard deviation of the MFCC vector, through the entire sound;
  - *ZeroCrossingDensity*, averaged through all frames for a given sound;
  - *RollOff* – average (over the entire sound) frequency below which an experimentally chosen percentage of the accumulated magnitudes of the spectrum is concentrated. It is a measure of spectral shape, originating from speech recognition and used to distinguish between voiced and unvoiced speech;
  - *Flux* – average of the difference between the magnitude of the DFT points in a given frame and its successive frame; value very low, and multiplied by 10^7;
  - *AverageFundamentalFrequency* – average fundamental frequency of a given sound (maximum likelihood algorithm applied for pitch estimation);

- additional MPEG-7 descriptors:

  - *HarmonicSpectralCentroid* – the average of the instantaneous Harmonic Centroid, over each analyzing frame. The instantaneous Harmonic Spectral Centroid is the mean of the harmonic peaks of the spectrum, weighted by the amplitude in linear scale;
  - *HarmonicSpectralSpread* – the average, over entire sound, instantaneous harmonic spectral spread. Instantaneous harmonic spectral spread is calculated as the standard deviation of the harmonic peaks of the spectrum with respect to the instantaneous harmonic spectral centroid, weighted by the amplitude;
  - *HarmonicSpectralVariation*, average over the entire sound of the instantaneous harmonic spectral variation, i.e., of the normalized correlation between amplitudes of harmonic peaks of each 2 adjacent frames;
  - *HarmonicSpectralDeviation* – average over the entire sound of the instantaneous harmonic spectral deviation, representing the spectral deviation of the log amplitude components from a global spectral envelope;

- *tris*: $tris_1, \dots, tris_9$ – various ratios of harmonic partials; $tris_1$: energy of the fundamental to the total energy of all harmonics, $tris_2$: amplitude difference [dB] between partials 1 and 2, $tris_3$: ratio of partials 3–4 to the total energy of harmonics, $tris_4$: ratio of partials 5–7 to all harmonics, $tris_5$: ratio of partials 8–10 to all harmonics, $tris_6$: ratio of the remaining partials to all harmonics, $tris_7$:

brightness – gravity center of spectrum, $tris_8$: contents of even partials in spectrum, $tris_9$: contents of odd partials;

- time-domain related MPEG-7 parameters:

  - *LogAttackTime* – the decimal logarithm of the duration from the time when the signal starts to the time when it reaches its sustained part or its maximal value, whichever comes first;
  - *TemporalCentroid* – energy weighted average of the sound duration; this parameter shows where in time the energy of the sound is focused.

3   Audio Data

In this research, we aimed at recognizing the following 14, recorded from MUMS CDs [9]: B-flat clarinet, flute, oboe, English horn, C trumpet, French horn, tenor trombone, violin (bowed), viola (bowed), cello (bowed), piano, marimba, vibraphone, and tubular bells. Twelve sounds, representing octave no. 4 (in MIDI notation) were used for each instrument. Additional sounds were mixed with the main sounds, and the level of added sounds was adjusted to be of 6.25%, 12.5%, 25%, and 50% of the main sound. For each main instrumental sound, additional 4 mixes were prepared for each level: with white noise, pink noise, triangular wave, and saw-tooth of the same pitch as the main sound. As a result, we obtained mixes of sounds with highly overlapping spectra, thus being most difficult case for a classifier to recognize the main instrument. The obtained sounds were parameterized, and feature vectors were calculated as shown in Sect. 2.

4   Experiments and Results

PCA applied to the data representing parameterized sounds, as described in Sect. 2, allowed us to reduce the size of the feature vector. The examined variables, i.e., sound parameters, were transformed through PCA into a smaller number of new, non-correlated factors (factors obtained from PCA are non-correlated). So called factor loadings, resulting from PCA, represent correlation coefficients between our initial parameters and new factors, as well as factors scores. For our data, we obtained 67 factors for 50% level of added sounds, 68 factors for 12.5% and 25% level, and 70 factors for 6.25% level. To avoid presenting numerous tables, we will show the results using tables obtained for 25% level. The size of these tables makes it impossible to show the complete table in this paper; illustrative fragment of factor loadings for the data with 25th level of additional sounds is shown in Table 1. Coefficients for which modulus exceeds 0.7 are shown in bold.

The results of PCA are a basis for discriminant analysis, which allows checking whether the obtained factors allow good discernment of the investigated instruments, and which factors actually possess the highest discriminant power. Discriminant functions are usually linear combinations of the initial input data (obtained

Table 1 Factor-parameters correlations (factor loadings), for the data with 25% level of additional sounds (the fragment of the table, for the first 9 factors)

Parameter	Factors								
	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6	Factor 7	Factor 8	Factor 9
spectrumspread	0.331	0.353	−0.049	0.153	0.064	0.543	−0.070	−0.025	0.271
flat1	0.028	0.079	−0.009	0.072	0.077	0.056	0.439	0.030	0.037
flat2	0.046	0.081	−0.012	0.008	0.039	0.045	**0.941**	−0.009	0.013
flat3	0.062	0.060	0.010	−0.036	−0.054	0.046	−0.005	−0.028	0.049
flat4	0.091	−0.087	−0.007	−0.032	−0.052	0.013	**−0.928**	−0.019	0.101
flat5	0.112	−0.102	−0.038	0.025	−0.009	−0.010	−0.068	0.040	0.164
flat6	0.117	0.045	−0.028	0.005	0.029	0.020	**0.859**	0.111	0.240
flat7	0.147	0.073	0.007	−0.045	0.168	0.106	0.181	0.205	0.426
flat8	0.272	0.108	0.026	−0.027	0.106	0.148	−0.087	0.094	0.528
flat9	0.258	0.103	0.050	−0.006	0.093	0.072	0.371	0.174	0.457
flat10	0.406	0.101	0.063	−0.021	0.123	0.160	0.053	0.127	**0.716**
flat11	0.438	0.078	0.001	−0.034	0.074	0.157	0.052	0.086	**0.748**
flat12	0.499	0.073	−0.040	0.005	0.048	0.135	0.006	0.053	**0.749**
flat13	0.580	0.050	−0.041	0.078	0.049	0.131	−0.004	−0.011	0.698
flat14	0.638	0.084	−0.032	0.057	0.050	0.140	−0.001	0.012	0.632
flat15	**0.720**	0.064	−0.023	0.094	0.032	0.114	−0.012	0.006	0.529
flat16	**0.784**	0.033	−0.034	0.094	0.009	0.108	−0.004	−0.021	0.450
flat17	**0.880**	0.026	−0.040	0.106	−0.003	0.106	0.011	−0.044	0.295
flat18	**0.919**	0.050	−0.037	0.124	−0.010	0.105	−0.001	−0.034	0.216
flat19	**0.935**	0.019	−0.028	0.099	−0.014	0.096	0.009	−0.040	0.153
flat20	**0.956**	−0.002	−0.012	0.075	−0.014	0.083	−0.001	−0.039	0.100
flat21	**0.965**	−0.014	−0.005	0.028	−0.027	0.082	−0.008	−0.020	0.048
flat22	**0.952**	−0.033	0.008	0.015	−0.012	0.065	−0.004	−0.031	0.019
flat23	**0.926**	−0.024	−0.017	−0.024	−0.013	0.084	−0.031	0.005	0.017
flat24	**0.919**	−0.030	0.005	−0.015	−0.005	0.032	−0.001	0.086	−0.026
flat25	**0.811**	−0.008	0.042	−0.017	0.005	0.011	0.060	0.111	−0.060
…	…	…	…	…	…	…	…	…	…
energy	−0.035	0.009	0.005	**0.962**	0.006	0.040	0.005	−0.064	−0.069
mfccmin	−0.239	−0.031	0.026	**−0.842**	−0.027	−0.264	−0.026	0.065	−0.108
mfccmax	0.223	0.105	−0.060	**0.842**	−0.010	0.200	−0.009	−0.001	−0.042
mfccmean	0.099	0.041	−0.083	−0.631	−0.048	0.315	−0.118	−0.351	−0.098
mfccdis	−0.021	−0.014	0.005	**0.941**	0.022	0.096	0.026	−0.097	0.018
mfccstd	0.080	0.025	−0.012	**0.958**	0.014	0.148	0.021	−0.057	0.033
zerocross	0.383	0.037	−0.054	0.261	−0.042	0.670	0.003	−0.195	0.146
rolloff	0.129	0.022	−0.019	0.104	−0.008	**0.764**	0.014	−0.092	0.233
flux×10⁷	0.057	−0.023	−0.029	0.035	−0.060	−0.171	−0.024	0.075	0.393
avgfund	−0.208	−0.123	−0.002	0.272	0.103	−0.163	0.319	0.128	−0.225
harmoCen	0.370	0.004	−0.026	0.350	0.027	**0.749**	0.059	−0.024	0.102
hss	0.186	−0.030	−0.068	0.587	−0.101	0.281	−0.122	0.227	−0.120
HarmSpectralVariation	0.149	−0.061	−0.068	−0.040	0.238	0.097	0.018	0.116	0.650
HarmSpectralDeviation	−0.255	0.170	−0.125	−0.390	−0.007	−0.168	−0.024	−0.133	0.086
tris1	−0.053	−0.058	0.025	0.055	0.033	−0.282	0.087	**0.834**	0.176
tris2	0.063	−0.049	0.127	−0.251	0.028	0.039	0.107	0.032	0.009
tris3	−0.170	−0.036	−0.013	0.084	0.015	−0.064	−0.020	**−0.712**	−0.152
tris4	−0.093	−0.025	−0.015	0.118	−0.050	0.190	−0.151	−0.090	−0.190
tris5	0.107	−0.066	−0.018	0.050	−0.007	0.430	−0.055	−0.053	−0.165
tris6	0.293	−0.009	0.010	0.127	0.029	**0.791**	0.059	0.064	0.112
tris7	0.248	−0.018	−0.001	0.156	0.046	**0.819**	0.020	−0.058	0.214
tris8	0.001	−0.072	−0.027	0.187	−0.029	0.022	−0.137	−0.626	−0.251
tris9	−0.067	−0.024	−0.017	0.144	−0.028	0.258	−0.014	−0.466	−0.169
logAttack	−0.078	−0.114	−0.005	0.330	−0.167	−0.285	−0.023	−0.224	−0.212
TemporalCentroid	−0.135	−0.010	−0.142	0.159	−0.166	−0.119	0.002	−0.307	−0.380

from PCA). Our calculations were performed using Principal Components and Classification Analysis as well as Discriminant Function Analysis units of STATISTICA software system [11], so the obtained discriminant functions are linear combinations

of the PCA output. Since we had 14 instruments (classes), 13 discriminant functions were obtained for each data set, i.e., for each level of added sounds. Table 2 shows standardized canonical discriminant function coefficients (F1–F13) for the first nine factors, again for the data with 25% level of added sounds. Using this table, we can find out which factors were of highest importance to discriminate between instruments – in orded to do this, we check modulus of factors for the first discriminant function. The highest ones are shown in bold in Table 2, and they include Factor 9, Factor 4 and Factor 8. From Table 1 we can read which parameters show the highest correlation with these factors – these parameters are *flat10*, *flat11*, *flat12*, *energy*, *mfccmin*, *mfccmax*, *mfccdis*, *mfccstd*, *tris1*, and *tris3*. For 25% level, 68 factors explains 77% variance in the data.

Table 2 Standardized canonical discriminant function coefficients (F1–F13) for the first nine factors, again for the data with 25% level of added sounds

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
Factor 1	0.561	0.201	−0.342	−0.082	0.036	0.141	0.079	−0.353	−0.108	−0.032	0.045	0.050	−0.064
Factor 2	0.326	0.120	0.259	−0.352	0.090	0.181	−0.153	−0.206	0.031	−0.051	0.202	0.077	−0.065
Factor 3	0.105	−0.408	−0.204	0.174	0.088	0.040	−0.047	−0.047	−0.186	0.061	0.024	0.093	0.045
Factor 4	**−0.959**	−0.794	0.204	−0.395	−0.084	−0.108	0.045	−0.023	−0.028	0.067	−0.014	−0.002	0.009
Factor 5	0.481	−0.206	0.331	−0.309	0.209	−0.255	−0.086	−0.164	0.045	−0.116	−0.007	0.010	−0.026
Factor 6	0.524	−0.058	0.202	0.545	−0.180	−0.390	−0.103	0.012	−0.004	−0.266	−0.061	−0.045	0.105
Factor 7	0.029	0.097	0.074	0.064	0.106	−0.046	0.055	0.008	−0.029	0.073	−0.053	−0.022	−0.037
Factor 8	**0.852**	−0.781	−0.043	0.119	−0.468	0.251	0.169	−0.152	0.113	−0.050	0.024	0.009	−0.006
Factor 9	**1.267**	−0.003	0.208	−0.591	0.091	0.031	0.052	0.110	−0.132	0.196	−0.083	0.029	0.087

The quality of the obtained classification can be presented in form of a classification matrix, containing information about the number and percentage of correctly classified objects for each instrument. The classification matrix for 25% level of sound added in mixes is shown in Table 3. As we can see, the best recognition of instrument is for trumpet, piano, tubular bells (100%), vibraphone (98%), and the worst recognition is for oboe (78.33%), most often mistaken for English horn. This is still high level of accuracy for 14 classes, and English horn is actually a variation of oboe. Therefore, it is not surprising that these 2 instruments were mistaken for each other.

Additionally, the quality of the obtained discrimination is illustrated in Table 4, presenting squared Mahalanobis distances D^2 which illustrate how distantly from each other are distributed the centroids of classes (instruments). We also performed F-test to estimate the statistical importance of these values, and the results showed that distances between classes are statistically significant at 0.01 significance level.

As we can see from Table 4, the best discernment (the highest D^2) was between trumpet and marimba, oboe and marimba, oboe and piano, and violin and piano. The lowest discernment was between oboe and English horn, viola and cello, viola and violin, and trombone and French horn.

Our research performed for all mentioned levels of added sounds yielded similar results. All classes were distant with respect to D^2 at 0.01 significance level of statistical importance. For 6.25% level, 70 factors explain 75% of variation in the

Table 3 Classification matrix: rows – observed classifications, columns – predicted classifications, for the data with 25% level of added sounds

	Accuracy	bclarnet	cello	ctrpt	enhorn	flute	frhorn	mrmba	oboe	piano	trbone	tbells	vbphne	viola	violin
bclarinet	91.67	55	0	1	0	0	0	0	2	0	0	0	0	0	2
cello	88.33	0	53	0	0	0	0	0	0	0	0	1	0	5	1
ctrumpet	100	0	0	60	0	0	0	0	0	0	0	0	0	0	0
enghorn	83.33	0	0	0	50	3	0	0	7	0	0	0	0	0	0
flute	91.67	0	0	0	2	55	0	0	1	0	2	0	0	0	0
frhorn	81.67	0	0	0	2	1	49	0	0	0	7	0	1	0	0
marimba	85	0	0	0	0	0	0	51	0	0	0	0	9	0	0
oboe	78.33	1	0	0	11	0	0	0	47	0	0	0	0	1	0
piano	100	0	0	0	0	0	0	0	0	60	0	0	0	0	0
trombone	83.33	0	0	0	1	1	8	0	0	0	50	0	0	0	0
tbells	100	0	0	0	0	0	0	0	0	0	0	60	0	0	0
vibraphone	98.33	0	0	0	0	0	0	0	0	1	0	0	59	0	0
viola	80	0	5	0	0	0	0	0	1	0	0	0	0	48	6
violin	90	0	0	0	0	0	0	0	2	0	0	0	0	4	54
Total	89.41	56	58	61	66	60	57	51	60	61	59	61	69	58	63

Table 4 Squared Mahalanobis distances D^2 for 25% level of added sounds. All values statistically significant at the level of $\alpha = 0.01$

	bclrnet	cello	ctrpet	enhorn	flute	frhorn	mrmba	oboe	piano	trbone	tbells	vphne	viola	violin
bclarinet	0.00	40.94	26.54	33.22	43.22	45.50	81.69	29.27	71.30	42.41	43.56	54.35	33.29	31.37
cello	40.94	0.00	69.32	24.42	21.61	35.66	63.55	29.69	60.07	42.63	25.00	31.77	7.50	19.44
ctrumpet	26.54	69.32	0.00	34.07	56.88	40.97	**114.46**	38.74	88.40	24.68	53.16	76.91	60.63	44.10
enghorn	33.22	24.42	34.07	0.00	14.16	25.13	84.79	5.71	73.29	14.80	29.73	49.90	22.79	19.15
flute	43.22	21.61	56.88	14.16	0.00	26.00	87.92	16.45	73.60	23.36	38.13	49.83	19.80	21.88
frhorn	45.50	35.66	40.97	25.13	26.00	0.00	61.05	36.91	42.88	9.05	27.44	23.25	40.89	41.69
marimba	81.69	63.55	**114.46**	84.79	87.92	61.05	0.00	**98.38**	63.54	89.80	60.05	26.55	75.26	88.54
oboe	29.27	29.69	38.74	5.71	16.45	36.91	**98.38**	0.00	**90.71**	22.50	39.71	64.66	21.90	18.52
piano	71.30	60.07	88.40	73.29	73.60	42.88	63.54	**90.71**	0.00	68.78	31.53	22.67	80.78	**91.25**
trombone	42.41	42.63	24.68	14.80	23.36	9.05	89.80	22.50	68.78	0.00	33.66	47.67	41.02	34.08
tbells	43.56	25.00	53.16	29.73	38.13	27.44	60.05	39.71	31.53	33.66	0.00	20.05	36.50	41.33
vibraphone	54.35	31.77	76.91	49.90	49.83	23.25	26.55	64.66	22.67	47.67	20.05	0.00	46.65	57.83
viola	33.29	7.50	60.63	22.79	19.80	40.89	75.26	21.90	80.78	41.02	36.50	46.65	0.00	8.25
violin	31.37	19.44	44.10	19.15	21.88	41.69	88.54	18.52	**91.25**	34.08	41.33	57.83	8.25	0.00

data, and the obtained accuracy exceeds 91%. Best classification was obtained for trumpet, flute, piano, tubular bells (100%), and the worst for oboe, usually mistaken again for English horn. The most important factors (no. 7, 14 and 15) correspond to *flat10,…,flat17*, *energy*, *mfccmin*, *mfccmax*, *mfccdis*, *mfccstd*, and *tris1*. For 12.5% level, 68 factors explains 77.22% of variance, and 88.57% accuracy was obtained. The best recognition is for trumpet, piano, tubular bells (100%), and the worst for oboe (73%). Discriminant factors no. 1, 3, and 5 correspond to *flat15,…flat25*, *basis4*, *basis5*, *energy*, *mfccmin*, *mfccmax*, *mfccdis*, *mfccstd*, *tris1*, and *tris8*. For 50% level, 67 factors explain 77.43% of variation in the data, and 85.59% accuracy was reached. The best recognition was for trumpet, piano, tubular bells (95%, 96%) and the worst for oboe – 70%. Factors no. 4, 1, 5, and 12 correspond to: *energy*, *mfccmin*, *mfccmax*, *mfccdis*, *mfccstd*, *flat11,…,flat25*, *basis4*, *basis5*, *tris4*, and *tris9*. For all levels, correlations with all initial sound parameters were shown.

All PCA factors indicated above show mutual relationships between sound parameters, e.g., *energy*, *mfccmin*, *mfccmax*, *mfccdis*, *mfccstd* were always shown as elements of the same factor, and all of them (except *mfcmin*) were always positively correlated with this factor.

5   Conclusions

In this paper, we applied PCA and discriminant analysis to identify musical instruments in mixes of various levels of added sounds and high-dimensional initial feature vector. Generally, the lower level of added sound, the higher recognition accuracy for particular instruments in classification matrices, although the recognition for 12.5% level comparable to the results for 25% (even a bit worse). The results for 6.25% were very satisfying, although 100% accuracy was never reached. The results for 12.5% and 25% level were very similar. The best results were obtained for 6.25% level, and the worst for 50% level. The same parameters were indicated as of high discriminant power in all cases: *energy*, *mfccmin*, *mfccmax*, *mfccdis*, *mfccstd*, and *flat12*; also very often the following parameters were indicated: *flat10*,...,*flat25*, *tris1*, *basis4* and *basis5*. Other parameters were not that much important for classification. Therefore, the results showed that PCA used as preprocessing and then linear discriminant analysis (applied on PCA output data) allows recognition of instruments in sound mixes for the investigated data, the size of feature vector can be limited, and the parameters indicated as of highest discerning power work well for various levels of added sounds.

References

1. Cont, A., Dubnov, S., Wessel, D.: Realtime multiple-pitch and multiple-instrument recognition for music signals using sparse non-negative constraints. In: Proceedings of the 10th International Conference on Digital Audio Effects, Bordeaux, France (2007)
2. Dziubinski, M., Dalka, P., Kostek, B.: Estimation of musical sound separation algorithm effectiveness employing neural networks. Journal of Intelligent Information Systems 24(2-3), 133–157 (2005)
3. Herrera, P., Amatriain, X., Batlle, E., Serra, X.: Towards instrument segmentation for music content description: a critical review of instrument classification techniques. In: Proceedings of the International Symposium on Music Information Retrieval (2000)
4. International Organization for Standardisation: ISO/IEC JTC1/SC29/WG11 N6828: MPEG-7 Overview (2004),
   http://www.chiariglione.org/mpeg/standards/mpeg-7/
   mpeg-7.htm

 5. Krzyśko, M.: Wielowymiarowa analiza statystyczna. Uniwersytet Adama Mickiewicza, Poznań (2000)
 6. MIR systems, `http://mirsystems.info/?id=3,0.0`
 7. Molla, K.I., Hirose, K., Minematsu, N.: Single mixture audio source separation using kld based clustering of independent basis functions. In: Proceedings of the 3rd International Conference on Electrical and Computer Engineering, Dhaka, Bangladesh, pp. 605–608 (2004)
 8. Morrison, D.F.: Multivariate statistical methods, 3rd edn. McGraw-Hill, New York (1990)
 9. Opolko, F., Wapnick, J.: Mums - mcgill university master samples. CD's (1987)
10. Rencher, A.C., Scott, D.T.: Assessing the contribution of individual variables following rejection of multivariate hypothesis. Communications on Statistics and Simulations 19(2), 535–553 (1990)
11. StatSoft, I.: Statistica, version 6 (2001), `http://www.statsoft.com/`
12. Wieczorkowska, A., Kolczyńska, E., Raś, Z.W.: Training of classifiers for the recognition of musical instrument dominating in the same-pitch mix. In: Nguyen, N.T., Katarzyniak, R. (eds.) New Challenges in Applied Intelligence Technologies. Studies in Computational Intelligence, vol. 134, pp. 213–222. Springer, Heidelberg (2008)

Part VIII
Computer Vision, Image Analysis and Virtual Reality

Spatial Color Distribution Based Indexing and Retrieval Scheme

Maria Łuszczkiewicz and Bogdan Smołka

Abstract. In this contribution we present a novel approach to the problem of color image indexing and retrieval. The indexing technique is based on the Gaussian Mixture modeling of the histogram of weights provided by the bilateral filtering scheme. In this way the proposed technique considers not only the global distribution of the color pixels comprising the image but also takes into account their spatial arrangement. The model parameters serve as signatures which enable fast and efficient color image retrieval. We show that the proposed approach is robust to color image distortions introduced by lossy compression artifacts and therefore it is well suited for indexing and retrieval of Internet based collections of color images.

Keywords: color image, indexing, Gaussian mixture, histogram, spatial distribution.

1 Introduction

The rapid developments in communication and information technologies lead to an exponentially growing number of images being captured, stored and made available on the Internet. However, managing this vast amount of visual information, in order to retrieve required images still remains a difficult task.

In this paper we focus on the problem of color image retrieval, based on the novel concept of a color distribution histogram, which incorporates the information on the spatial distribution of the color pixels. In the proposed method we apply the *Gaussian Mixture Model* (GMM) framework, which serves as a general descriptor of the image color distribution. Its main advantage is that it overcomes the problems connected with the high dimensionality of standard color histograms. Additionally, the

Maria Łuszczkiewicz · Bogdan Smołka
Institute of Automatic Control, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail:{maria.luszczkiewicz, bogdan.smolka}@polsl.pl

proposed method based on weighted two-dimensional Gaussians is robust to distortions introduced by compression techniques and therefore it can be used for the retrieval of images contained in the web-based databases, which very often store images in lossy formats. Although, several formerly proposed image retrieval techniques utilize the Gaussian Mixture Models as a color distribution descriptors (i.e., [3, 2]), the aspect of the distortions caused by the lossy compression is not taken into account. These methods simply index all images in the database by fitting GMM to the data, according to some predefined rules.

Moreover, in order to provide results reflecting the spatial arrangement of the colors of the given query, we propose to incorporate an additional weighting step into the process of GMM histogram approximation. The weights, which provide the information on the spatial arrangement of the color pixels, are delivered by the coefficients of the bilateral filter [6], which takes into account the color and spatial similarity of neighboring pixels.

2 Modified Color Image Histogram

In this contribution we operate in the normalized *rgb* space, which is independent on the color intensity I: $I_{ij} = R_{ij} + G_{ij} + B_{ij}$, $r_{ij} = R_{ij}/I_{ij}$, $g_{ij} = G_{ij}/I_{ij}$, $b_{ij} = B_{ij}/I_{ij}$, where i, j denote image pixels coordinates. The histogram $\Phi(x, y)$ in the $r - g$ chromaticity space is defined as $\Phi(x, y) = N^{-1} \sharp \{r_{i,j} = x, g_{i,j} = y\}$, where $\Phi(x, y)$ denotes a specified bin of a two-dimensional histogram with r-component equal to x and g-component equal to y, $\sharp$ denotes the number of elements in the bin and N is the number of image pixels.

The next step of the indexing method is the construction of a $r - g$ histogram approximation using the Gaussian Mixture Model framework and the Expectation-Maximization (EM) algorithm [1], as was extensively explored in [4, 5]. Figure 1 illustrates the modeling process.
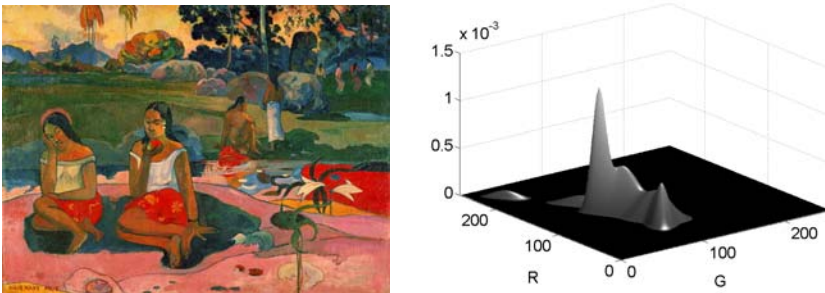
Fig. 1 Evaluation of GMM for *Gauguin* painting '*Nave, Nave Moe (Miraculous Source)*' from the database of WebMuseum [8] (**left**) and 3D visualization of GMM (**right**)

3 Color Dispersion and Bilateral Weighting

Although the efficiency of the retrieval based on the approximation of the r–g histogram evaluated using the GMM framework was proven to be very effective in previous authors' papers [4, 5], there is an obvious need for the incorporation of the information on the image spatial color structure into the retrieval process. If this information is not considered, it is possible that a set of images of the same color proportions but of different color arrangements will have the same chromaticity histogram, which does not reflect the differences in color spatial distribution as shown in Fig. 2. This step will enable to improve the quality of the results of the GMM based retrieval scheme.

We propose to include the spatial information into the process of the r–g histogram construction in form of the weighting coefficients evaluated by bilateral filter [6]. This approach takes into account two aspects: closeness (topology) and similarity (difference of colors). The bilateral filter (BF) provides the weights calculated for each image pixel according to the following formula:

$$BF(x,y) = \frac{1}{n} \sum_{i=1}^{n} e^{-(\frac{\|c_{x,y}-c_i\|}{h} \times \delta_1)^{k_1}} e^{(-\frac{\psi_1}{\psi_2} \times \delta_2)^{k_2}} , \tag{1}$$

where h is the smoothing parameter, $c_{x,y}$ and c_i denote the colors of the window center and of each of n surrounding pixels, ψ_1 is the topological distance between the window center and the particular pixel, ψ_2 is the distance parameter defining the importance of the topological distance, δ_1 and δ_2 are normalization coefficients. In the paper the filtering window size was equal to 10% of the size of the analyzed
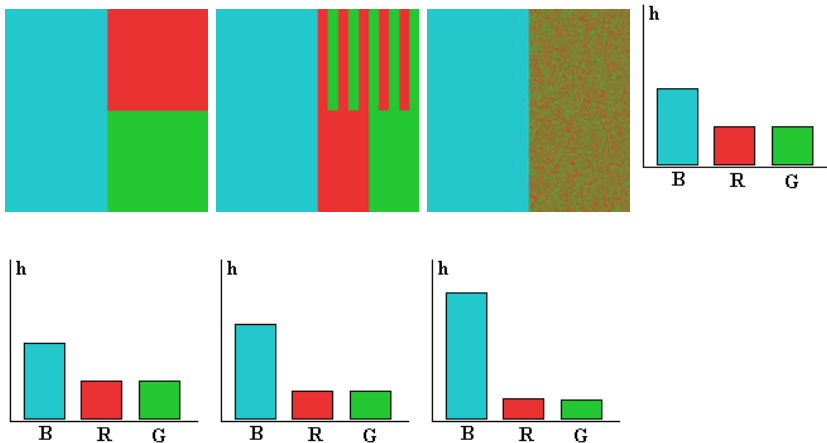
Fig. 2 Set of images consisting of color pixels in the same proportions but with different spatial arrangements produce the same chromaticity histogram (upper row). The main aim of the proposed method is to reflect the spatial structure of the image colors into the structure of the r–g histogram (bottom row)
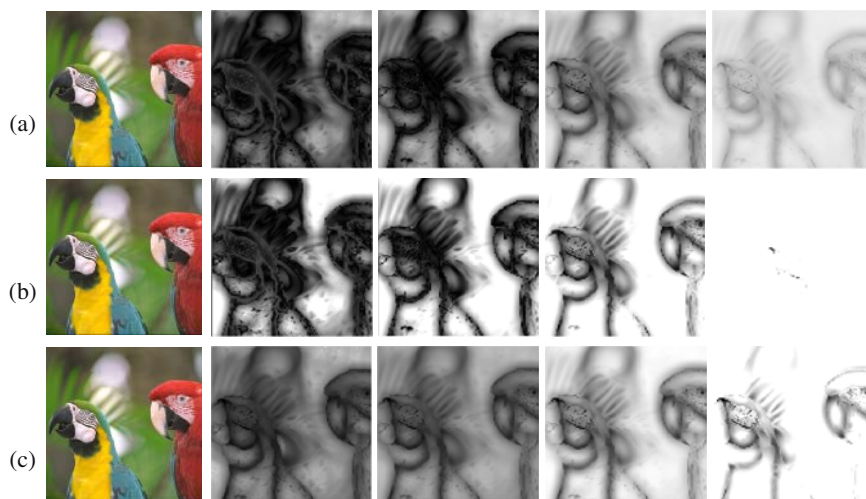
Fig. 3 The comparison of *BF* weights for various set of parameters: **a** $k_1 = k_2 = 1$, $h = \{0.05, 0.1, 0.3, 1\}$, **b** $k_1 = k_2 = 2$, $h = \{0.05, 0.1, 0.3, 1\}$, **c** $h = 3$, $k_1 = k_2 = \{0.3, 0.7, 1.2, 2.5\}$

image ($\gamma = 0.1$) and we assumed $k_1 = k_2$. Figure 3 illustrates the dependence of the bilateral weighting coefficients on the h parameter. The use of the bilateral filtering scheme provides a possibility to strengthen in the $r - g$ histogram all the pixels belonging to homogenous color areas. For the further experiments we have chosen: $h = 0.3$ and $k_1 = k_2 = 1$.

Having the BF calculated, the r–g histogram is computed assigning to each image pixel a corresponding weight of BF as shown in Fig. 4. The next step is to evaluate the GMM histogram approximation providing the set of model parameters which serve as image index for retrieval purposes.

On the basis of the previously described methodology extensive experiments were undertaken in order to test the effectiveness of the approach presented in this paper.

As it was proven [4, 5], the GMM (1) based retrieval technique is efficient independently on the applied compression scheme. Therefore, we compared the proposed methodology with other known indexing and retrieval methods operating on color histograms for GIF compression method, such as: *Binary Threshold Histogram* (BTH) (2) [10], RG$_2$ chromaticity histogram modification (3) [7], non-uniform quantization (64 bins → 16R, 4G, 1B) (4) and direct r–g histogram comparison (5). As the similarity measure for methods 2–5 we used L_1 metric and for the GMM based technique (1) we used EMD (*Earth Mover's Distance*) as the most desired because of its beneficial features [4, 5, 9]. Figure 5 clearly proves that the GMM framework combined with the bilateral weighting is capable to provide a meaningful retrieval results not only imitating the query color palette but also considering the spatial arrangement of the image colors. The extensive experiments show that also for other compression
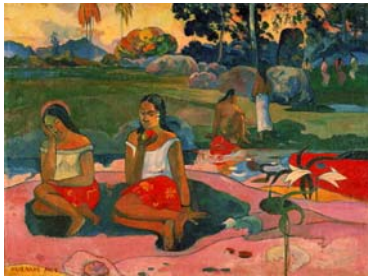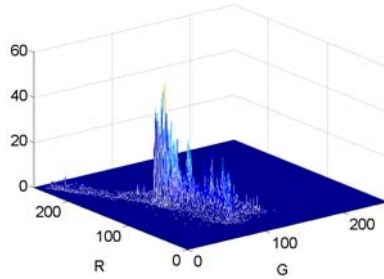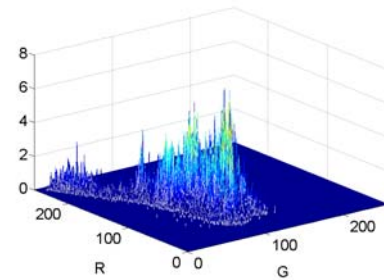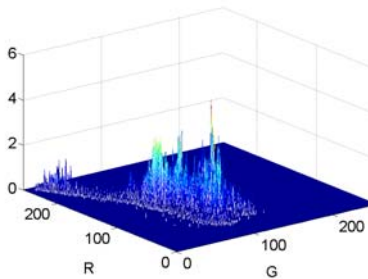
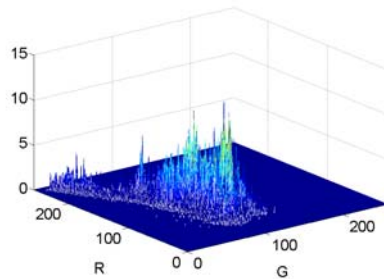Original image   r–g histogram of original
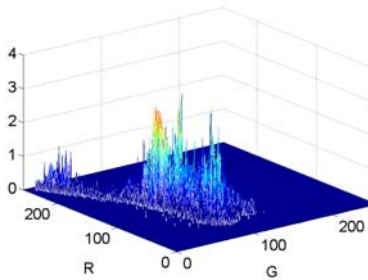
$k_1 = k_2 = 1, h = 0.05$ $k_1 = k_2 = 1, h = 0.3$

$k_1 = k_2 = 2, h = 0.05$ $k_1 = k_2 = 1, h = 0.3$

Fig. 4 The comparison of the the r–g histograms evaluated for the original color image for various sets of BF parameters. The histograms are computed by incorporating each image pixel into the histogram with corresponding weight of BF in order to reflect the spatial distribution of the colors within an image

schemes the same conclusion can be drawn. Figure 6 illustrates the comparison of these retrieval schemes evaluated for four sets of 31 randomly chosen images from the database of Wang [11]. This database, widely used for retrieval system testing purposes, comprise 1000 color images derived from *Corel* CD collections and divided into 10 distinct categories describing their content, e.g., *Horses*, *Beaches*, *Buses*. In this experiment as the relevance criterion the membership to the same category as
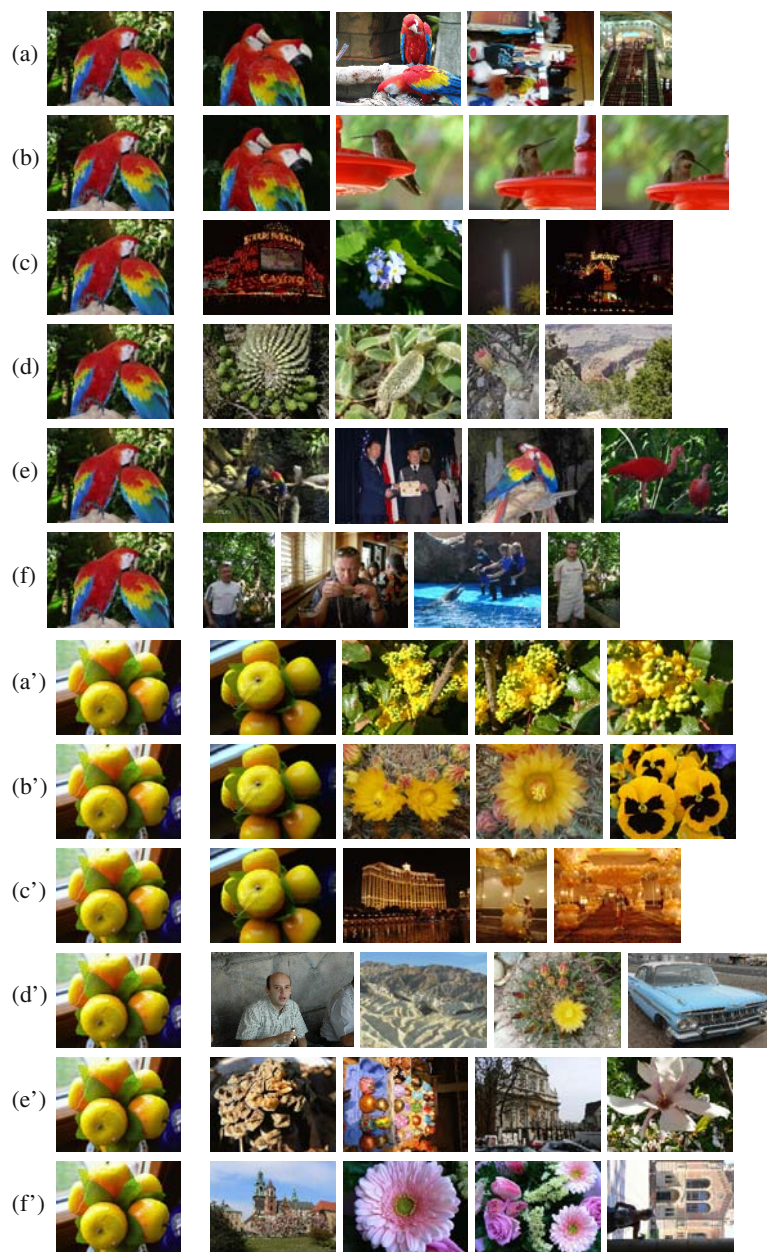
Fig. 5 The sets of the four highest ranked retrieval results (right) evaluated on the images of database of Authors comprising 6300 natural images and compressed with the GIF scheme and using the following methods: $\text{GMM}(a, a')$, $\text{GMM}_{k_1 = k_2 = 1, K = 0.3}$ (b,b'), BHT [10] (c,c'), direct r–g histogram comparison (d,d'), non-uniform quantization (e,e') and RG_2 [7] (f,f'). It can be observed that only the method proposed in this paper (b,b') considers the spatial arrangement of colors comprised by the query
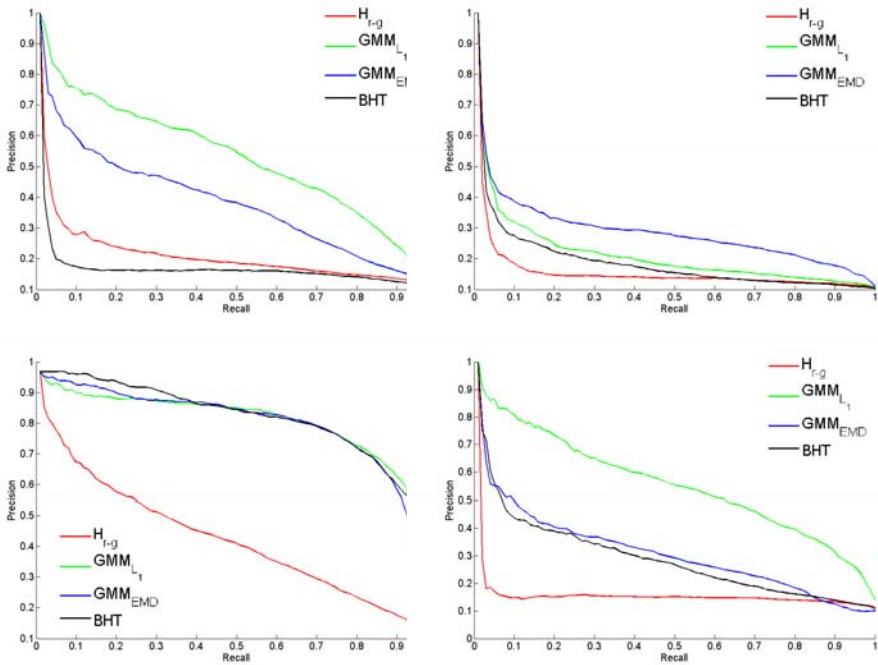
Fig. 6 The comparison of the efficiency of retrieval systems using the following methods: $GMM_{k_1=k_2=1, K=0.3}$ using as the similarity measure the L_1 metric (GMM_{L_1}) and EMD method (GMM_{EMD}), BHT [10], direct r–g histogram comparison using L_1 similarity measure (H_{r-g}). The experiments were evaluated for 4 randomly (uniform distribution) chosen sets of 31 images, compressed using GIF method and taken from the database of Wang

the query was assumed. Moreover, the images in categories of Wang collection are, in general, homogenous in color palette domain. The retrieval results illustrated by Figs. 5 and 6 show that the approach proposed in this paper is suitable for retrieval purposes which take into the account the criterion of similarity of color distribution within the image. Nevertheless, the retrieval results strongly depend on the content of the database, the relevance criterion and in the case the database of Wang one must be aware that retrieval results are affected by the assumption of the membership criteria. The presented results state that the proposed approach to color image retrieval based on the color distribution as a retrieval criterion is outperforming the other examined methods or at least is equally efficient as they are.

4 Conclusions and Future Work

In this paper we propose a novel approach to color image indexing and retrieval. The main contribution of this work is the adaptation of the concept of the bilateral filtering for the purposes of chromaticity histogram approximation in form

of *bilateral weighting*. The proposed scheme enables the retrieval system not only to take into account the overall image color palette but also the spatial arrangement of image colors. The satisfactory retrieval results were evaluated independently on the applied compression scheme. In our future work we will focus on the problem of adaptively choosing the optimal *BF* parameters in order to satisfy various user demands.

References

1. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data. Journal of the Royal Statistics Society 39, 1–38 (1977)
2. Jeong, S., Won, C.S., Gray, R.M.: Image retrieval using color histograms generated by Gauss mixture vector quantization. Computer Vision and Image Understanding 94(1-3), 44–66 (2004)
3. Kuo, W.J., Chang, R.F.: Approximating the statistical distribution of color histogram for content-based image retrieval. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 4, pp. 2007–2010 (2000)
4. Łuszczkiewicz, M., Smołka, B.: Gaussian mixture model based retrieval technique for lossy compressed color images. In: Kamel, M.S., Campilho, A. (eds.) ICIAR 2007. LNCS, vol. 4633, pp. 662–673. Springer, Heidelberg (2007)
5. Łuszczkiewicz, M., Smołka, B.: A robust indexing and retrieval method for lossy compressed color images. In: Proceedings of the IEEE International Symposium on Image and Signal Processing and Analysis, pp. 304–309 (2007)
6. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. International Journal of Computer Vision 39, 1–38 (2007)
7. Paulus, D., Horecki, K., Wojciechowski, K.: Localization of colored objects. In: Proceedings of International Conference on Image Processing, Vancouver, Canada, pp. 492–495 (2000)
8. Pioch, N.: Web museum, http://www.ibiblio.org/wm/ (1.03.2008)
9. Rubner, Y., Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In: Proceedings of International Conference on Computer Vision, pp. 59–66 (1998)
10. Walczak, K.: Image retrieval using spatial color information. In: Skarbek, W. (ed.) CAIP 2001. LNCS, vol. 2124, pp. 53–60. Springer, Heidelberg (2001)
11. Wang, J.Z., Li, J., Wiederhold, G.: SIMPLIcity: semantics-sensitive integrated matching for picture libraries. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(9), 947–963 (2001)
12. Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control 19, 716–723 (1974)
13. van Berendonck, C., Jacobs, T.: Bubbleworld: A new visual information retrieval technique. In: Pattison, T., Thomas, B. (eds.) Proceedings of the Australian Symposium on Information Visualisation, vol. 24, pp. 47–56. ACS, Adelaide (2003)
14. Bilmes, J.: A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and Hidden Markov Models. Tech. Rep. ICSI-TR-97-021, University of Berkeley (1997)
15. Hitchcock, F.L.: The distribution of a product from several sources to numerous localities. Journal of Mathematical Physics 23(20), 224–230 (1941)

16. Hsu, W., Chua, T.S., Pung, H.K.: An integrated color-spatial approach to content-based image retrieval. In: Proceedings of ACM Multimedia Conference, pp. 305–313 (1995)
17. Huang, J., Kumar, S.R., Mitra, M., Zhu, W.J., Zabih, R.: Image indexing using color correlograms. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 762–768 (1997)
18. Lei, X., Jordan, M.: On convergence properties of the EM algorithm for Gaussian mixtures. Neural Computation 8(1), 129–151 (1996)
19. Niblack, W., Barber, R.: The QBIC project: Querying images by content, using color, texture, and shape. In: Proceedings of SPIE: Storage and Retrieval for Image and Video Databases, pp. 173–187 (1993)
20. Pentland, A., Picard, R.W., Sclaroff, S.: Photobook: Tools for content-based manipulation of image databases. In: Proceedings of SPIE: Storage and Retrieval for Image and Video Databases, pp. 34–47 (1994)
21. Rabiner, L., Juang, B.H.: Fundamentals of Speech Recognition. Prentice Hall PTR, Englewood Cliffs (1993)
22. Rachev, S.T.: The Monge-Kantorovich mass transference problem and its stochastic applications. Theory of Probability and its Applications 4(XXIX), 647–676 (1984)
23. Redner, R., Walker, H.: Mixture densities, maximum likelihood and the EM algorithm. SIAM Review 26(2), 195–239 (1984)
24. Schwarz, G.: Estimating the dimension of a model. The Annals of Statistics 6(2), 461–464 (1978)
25. Smith, J., Chang, S.F.: Tools and techniques for color image retrieval. In: Proceedings of SPIE: Storage and Retrieval for Image and Video Databases, pp. 426–437 (1996)
26. Smołka, B., Szczepański, M., Lukac, R., Venetsanoloulos, A.: Robust color image retrieval for the World Wide Web. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pp. 461–464 (2004)
27. Swain, M.J., Ballard, D.H.: Color indexing. International Journal on Compututer Vision 7(1), 11–32 (1991)
28. Wu, J.: On the convergence properties of the EM algorithm. The Annals of Statistics 11(1), 95–103 (1983)

Synthesis of Static Medical Images with an Active Shape Model

Zdzisław S. Hippe, Jerzy W. Grzymała-Busse, and Łukasz Piątek

Abstract. A collection of new algorithms for the synthesis of selected features of melanocytic skin lesion images is briefly outlined. The key approach in the developed synthesis methodology is a semantic conversion of textual description of melanocytic skin lesions – by an in-house developed system – into hybrid, i.e., vector-raster type digital images. It seems that synthetic images generated by means of the developed algorithms can be successfully used as an alternative source of knowledge. It is assumed that the developed methodology can be successfully used in education of dermatology students and preferred medical doctors.

Keywords: TDS, melanocytic skin lesion, image synthesis.

1 Introduction

In the past few years an increasing interest in images of melanocytic skin lesions can be observed. These images can be treated as a visual support in diagnosing of malignant melanoma, currently one of the most dangerous type of tumors [10]. But on the other hand, the lack of professional computer databases containing images of above mentioned lesions is clearly noticed. This situation at least in this country yields various difficulties in the development such databases, among them owing the specific interpretation of the personal data protection act. The current interpretation of this act imposes the necessity to obtain patient's approval not only for making the picture of a real lesion (either in hospital, or in clinic), but also the permission for publishing or even handing it over to another scientific research institution, that specializes in processing digital images. These reasons inspired us to start some research on the development of generalized algorithms to synthesize medical images in general, but specifically to synthesize static images of melanocytic skin lesions.

Zdzisław S. Hippe · Jerzy W. Grzymała-Busse · Łukasz Piątek
Institute of Bioinformatics, University of Information Technology and Management,
35-225 Rzeszów, Poland
e-mail:`{zhippe,lpiatek}@wsiz.rzeszow.pl`

Jerzy W. Grzymała-Busse
Department of Electrical Engineering and Computer Science,
University of Kansas,
Lawrence, KS 66045-7621, US
e-mail: `Jerzy@ku.edu`

In our research guided along these lines, we have developed effective algorithms for semantic conversion of textual description of melanocytic lesions into respective images.

2 Structure of the Source Dataset

The source informational database describes real cases of melanocytic skin lesions for 53 anonymous patients [6], confirmed by histopatological tests. The dataset consists of 212 real digital images of those lesions and 53 textual descriptions, in the form of 15th component data vectors, which values transmit information about presence or lack of specific symptoms of a given lesion. These symptoms (in machine learning language called *descriptive attributes*) are: the type of *asymmetry*, character of the *border* of a lesion, combination of 6 allowed *colors* and 5 allowed *structures* observed in the lesion, the value of the TDS-parameter (*Total Dermatoscopy Score*) [3,4] and category to which each case has been assigned.

3 Methodology of the Research

In our research we concern on synthesis of melanocytic skin lesion images of two most dangerous groups of these lesions, namely *Nevus* and *Melanoma* [10]. Precisely, group *Nevus* included five types of lesions, i.e., *Junctional nevus*, *Junctional and dermal nevus*, *Atypical/dysplastic nevus*, *Dermal nevus* and *Palmo-plantar nevi*, whereas *Melanoma* group contained two types of lesions, namely *Superficial melanoma* and *Nodular Melanoma*. The new developed algorithms define the hybrid, i.e., vector-raster type approach to synthesis of medical images. Namely, the usual vector technique is combined with a method based on the Active Shape Model [3] (further called ASM), and then used to synthesize of lesion's *asymmetry*. On the other hand raster graphics operations are implemented for mapping of remaining symptoms of lesions. The mapping of the *border* of lesion image is accomplished by means of specific conversion based on low-capacity Gaussian filter [7] for the regarded fragments of the image. Mapping of the two remaining characteristic symptoms of lesion's images, i.e., *colors* and *structures*, is done by using pre-defined fragments (so called textures) of images of melanocytic lesions. In this paper only algorithm of mapping of lesion's asymmetry (see Sect. 4) is presented.

4 Synthesis of Lesion's Asymmetry

Mapping of lesion's asymmetry relies first on the use of ASM algorithm [2], and then application of de Casteljau algorithm [14], for the tentative creation of the synthesized images (see Fig. 1).

Active Shape Model is a kind of structural information about the mean shape of a digital image, joined with information about the deviation from the 'mean shape'.

ASM models can be obtain by statistical analysis of so called point distribution model [13], from set of points labeled onto the learning images, with the required condition, that points (landmarks) of each training images represents a required correspondence (Fig. 2). Every shape **x** from the training set is represented as an n-point polygon in images coordinates: $\mathbf{X} = (x_1, y_1, \ldots, x_{n-1}, y_{n-1}, x_n, y_n)^T$ (every point with coordinates (x_n, y_n) for n equal from 1 to 64 was defined manually at intersections of 64-fold symmetry axis with an edge of the lesion image (Fig. 3)). Then, each new shape is obtained according to (1):

$$\mathbf{X} = \blacksquare + \mathbf{P_t} \cdot \mathbf{b}, \tag{1}$$

where: $\blacksquare$ is the mean shape of all images from the training set, $\mathbf{P_t} = [u_1, u_2, \ldots, u_t]$ includes first **t** eigenvectors of the covariance matrix, and $\mathbf{b} = [b_1, b_2, \ldots, b_t]^T$ contains shape model parameters for each of the selected eigenvectors. Components of the weight vector b_i can be fixed within the following region:

$$-s \times \sqrt{\lambda_i} \leq b_i \leq s \times \sqrt{\lambda_i}. \tag{2}$$

Here λ_i is a selected eigenvalue, whereas s is a constant factor, receiving in our research constant value equal to 3.

Series of 64 new formed point's, received as a consequence of changing coordinates of mean shape point's (according to (1)) are connected by 64 segment lines (see Fig. 2).

Connecting the consecutive points define the control polygon of the curve. Finally, using algorithm of cutting corners [12], each segment line is splitted with a
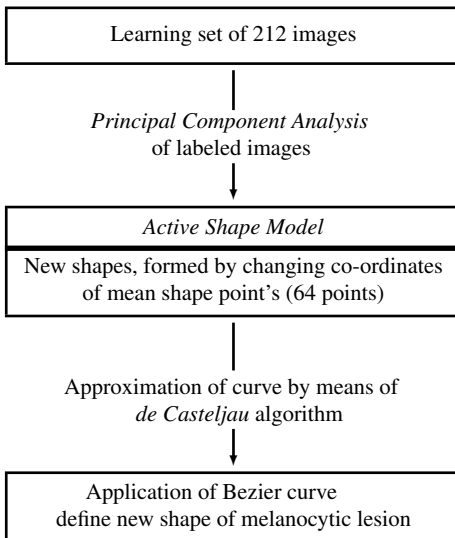

Fig. 1 A sequence of selected operations in process of synthesis of an image
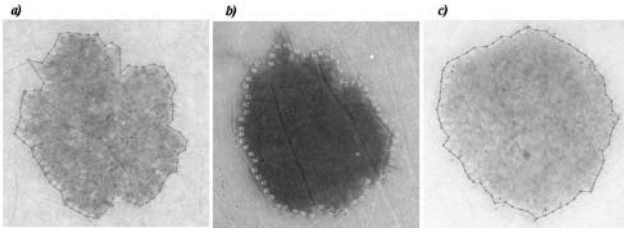
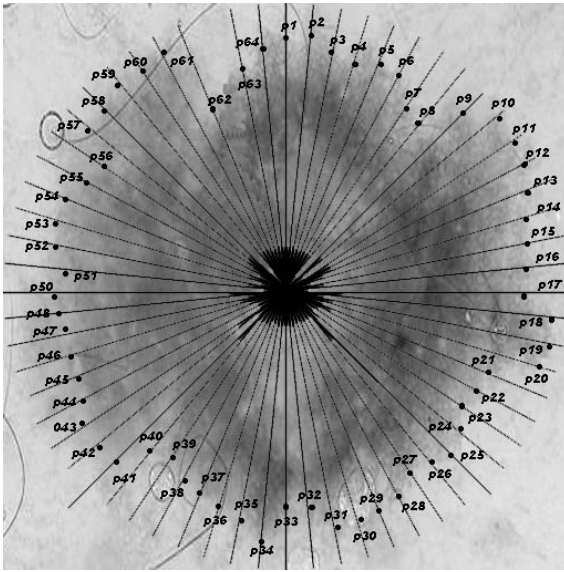**Fig. 2** Selected images from the learning set with marked landmark points

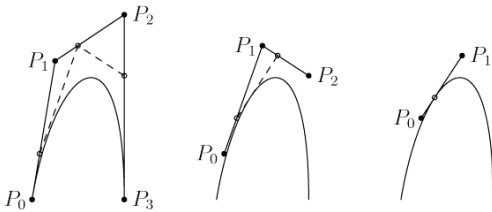**Fig. 3** An example image from the training set with marked 64 points

Fig. 4 An example of subsequent iterations of the de Casteljau algorithm

fixed ratio $t/(1-t)$. The process should be repeated until arriving at the single point (this is the point of a curve corresponding to the parameter t). This process is performed iteratively: a curve created via this method is known as Bezier curve [8] (see Fig. 4).
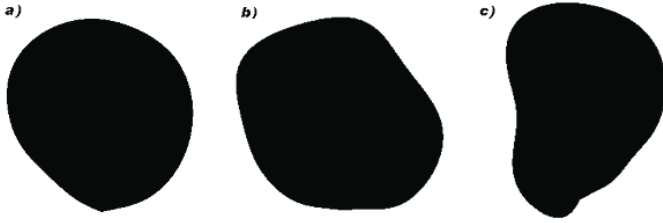
Fig. 5 Examples of surfaces defined with Bezier curves for: (**a**) *<symmetric lesion>*, (**b**) *<one-axial asymmetry lesion>*, (**c**) *<two-axial asymmetry lesion>*

Example of sample shapes obtain according with the discussed methodology, for 3 various types of lesion's asymmetry, i.e., *<symmetric lesion>*, *<one-axial asymmetry lesion>* and *<two-axial asymmetry lesion>* is presented in Fig. 5.

5 Program Implementation

Developed algorithms of synthesis melanocytic lesion's asymmetry are implemented in C++ language, combined with the use of MFC library [9] and OpenGL graphic library [11]. GUI (Graphic User Interface) of the application (at moment in Polish) is presented in Fig. 6.
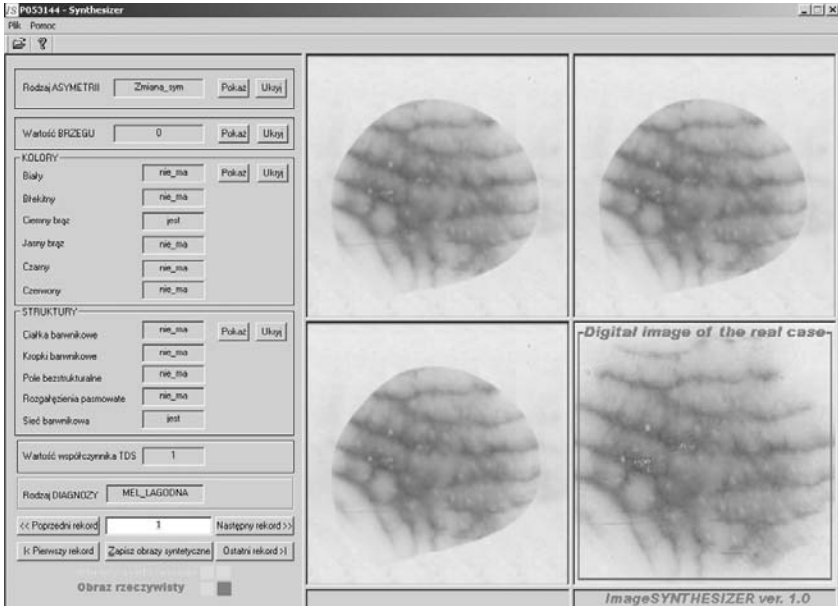

Fig. 6 The main screen of the developed *ImageSYNTHESIZER*, a computer program system for synthesis of static medical images (an example of melanocytic skin lesion of the category *Benign nevus* (*Junctional nevus*))

6 Conclusion

Combination of the new algorithm described here with algorithms discussed earlier [5] follow a new line to hybrid synthesis of images. In the first experiments we found that the developed algorithms can be easily used to create very large, multi-category informational database, which can be successfully used not only in teaching of medicine students, but also in a job practice of dermatologists and preferred medical doctors. It seems that synthetic images, generated by means of the developed algorithms, can be successfully used as an alternative source of knowledge in relation to digital medical images.

References

1. Braun-Falco, O., Stolz, W., Bilek, P., Merkle, T., Landthaler, M.: Das dermatoskop. Eine vereinfachung der auflichtmikroskopie von pigmentierten hautveranderungen. Hautarzt 40, 131 (1990)
2. Cootes, T.F., Hill, A., Taylor, C.J., Haslam, J.: The use of active shape models for locating structures in medical images. Image and Vision Computing 12(6), 355–366 (1994)
3. Cootes, T.F., Taylor, C.J.: Active shape models. In: Proceedings of the British Machine Vision Conference, Leeds, UK, pp. 387–396 (1992)
4. Hippe, Z.S.: Computer database NEVI on endangerment by melanoma. TASK Quarterly 3(4), 483–488 (1999)
5. Hippe, Z.S., Piątek, Ł.: Synthesis of static medical images - an example of melanocytic skin lesions. In: Kurzyński, M., Puchała, E., Woźniak, M., Żołnierek, A. (eds.) Computer Recognition Systems 2, pp. 503–509. Springer, Heidelberg (2007)
6. La Roche-Posay: Atlas of dermatoscopy of pigmented skin tumors, http://www.pless.fr/dermatoscopie/ (20.01.2009)
7. Lifshitz, L.M., Pizer, S.M.: A multiresolution hierarchical approach to image segmentation based on intensity extrema. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 529–540 (1990)
8. Manocha, D., Demmel, J.: Algorithms for intersecting parametric and algebraic curves I: simple intersections. ACM Transactions on Graphics 13(1), 73–100 (1994)
9. Microfinance Centre: Home page, http://www.mfc.org.pl/ (20.01.2009)
10. Stolz, W., Braun-Falco, O., Bilek, P., Landthaler, M., Burgdorf, W.H.C., Cognetta, A.B.: The Atlas of dermatology. Czelej D., Michalska-Jakubus M., Ziarkiewicz M., Czelej Sp. z o.o., Lublin, Poland (2006) (in Polish)
11. The Industry's Foundation for High Performance Graphics: OpenGL, http://www.opengl.org/ (20.01.2009)
12. Wikipedia Foundation, Inc.: Algorytm de casteljau, http://pl.wikipedia.org/wiki/Algorytm_de_Casteljau (20.01.2009)
13. Wikipedia Foundation, Inc.: Point distribution model, http://en.wikipedia.org/wiki/Point_Distribution_Model (20.01.2009)
14. Zorin, D., Schröder, P., Sweldens, W.: Interpolating subdivision for meshes with arbitrary topology. In: Proceedings of the 23rd International Conference on Computer Graphics and Interactive Technologies, New Orlean, US, pp. 189–192 (1996)

New Method for Personalization
of Avatar Animation

Piotr Szczuko, Bożena Kostek, and Andrzej Czyżewski

Abstract. The paper presents a method for creating a personalized animation of avatar utilizing fuzzy inference. First the user designs a prototype version of animation, with keyframes only for important poses, roughly describing the action. Then, animation is enriched with new motion phases calculated by the fuzzy inference system using descriptors given by the user. Various degrees of motion fluency and naturalness are possible to achieve. The proposed algorithm of the animation enrichment based on fuzzy description is thoroughly presented. The first part consists of creating fuzzy rules for the algorithm using results of subjective evaluation of the animated movement, the second one utilizes input descriptors for new motion phases calculation, which are finally added to the animation. Results of subjective evaluation of obtained animations are presented.

Keywords: animation, fuzzy logic, multimedia, subjective features.

1   Introduction

Animated computer characters are often used for virtual reality (VR) applications, computer games, educational software, and serve as actors in animated movies. Animated figure portraying a user in the VR is called avatar. Current trends in VR aim at providing full interaction and personalization of avatar's look, outfit, gender, or age. Therefore adding a new aspect of personality for adjustment – movement style – is a logic step in avatars development. The most advanced method for acquiring animated movement is Motion Capture, though it has very high technical requirements [7]. Very important drawback of that method is that capturing motion of a real actor does not allow to achieve exaggerated animation, typical for animated movies

Piotr Szczuko · Bożena Kostek · Andrzej Czyżewski
Multimedia Systems Department, Gdańsk University of Technology,
Narutowicza 11/12, 80-952 Gdańsk, Poland
e-mail:{bozenka,szczuko,andcz}@sound.eti.pg.gda.pl

and cartoons, and moreover changing of animation style is practically impossible. VR application creators can capture a few variants of actions for different types of avatars, but the work would be very tedious, and seems impractical. Therefore methods should be developed for changing a style of the motion, e.g., captured by real actor or of a prototype action prepared by the animator, and introducing new quality and range of styles defined by the user.

Some animation methods have been already developed [4, 5], aiming at achieving high quality motion without motion capture, allowing generation of motion variants e.g. depending on physical parameters in simulation. In the paper a new method for personalization of animated movement is proposed, combining the computer animation with the traditional cartoon one [1, 10]. It is a continuation of the work described by authors [9, 8]. Resulting from this research is a fluid, high quality animated movement achieved, with a style spanning between natural and exaggerated depending on user input. Employed advanced computer animation techniques guarantee effectiveness in designing animated movement utilizing well-known keyframe approach. On the other hand traditional animation approach serves as a source of practical rules for correct utilization of motion phases, assuring fluency of motion and correct display of character personality. The knowledge is available in an animation literature, gathered and constitutes during last hundred years of animation evolution [1, 10].

The new method proposed and verified in our work combines both mentioned domains of animation techniques utilizing fuzzy processing. The main goal of our work is to create an animation framework serving as a tool for animation preparation and employing fuzzy logic for automatic alteration of its style and fluidity depending on the user input. Fuzzy logic is used for modeling of the animation knowledge represented as relations between various motion parameters. The required knowledge is obtained from the literature and gathered during data mining of parameters of sample animations. Processing of that knowledge allows for establishing these parameters that have the strongest influence on subjective motion features: its *quality*, *fluidity*, and *style*. Fuzzy logic enables to operate with non crisp values, imprecise terms, and to process linguistic variables related to subjective descriptors originating in human perception of motion features. Using fuzzy logic in that scope is, as for now, unique approach for computer animation.

2   Animation Parameterization

It is assumed that a prototype animation is processed by the proposed algorithm, depending on user subjective requirements, resulting in a required style changes without influencing the action. The prototype animation is described by timing, and placing of animated objects (parameters in vector **A**). These animated objects are limbs of the character, and in fact these parameters describe sequence of character poses, defining the avatar action that should remain intact. Input data for processing are therefore **A** and subjective requirements related to *quality*, *fluidity*, and *style* (parameters contained in vector **Q**).
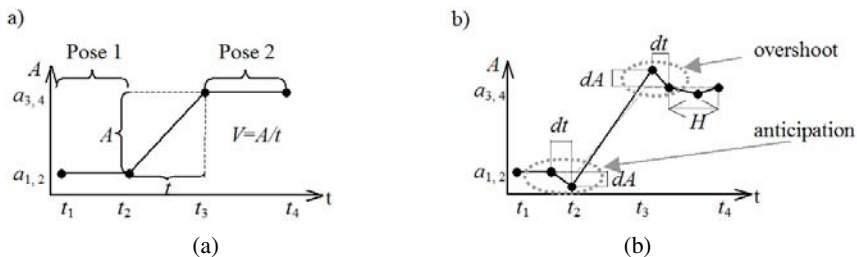
Fig. 1 Motion data: **a** transition between poses described by $\mathbf{A} = [V,A,t]$, **b** anticipation, overshoot and moving hold phases. H – range of variations of the pose

In case of typical handmade animation the sequence of poses is prepared by the animator, matching requirements defined by $\mathbf{A}$, but is also accompanied with additional motion phases. These phases do not change main action, but add subtle variations to transitions between already defined poses, influencing character *personality*, and motion *fluidity*, *quality*, and *style*. Similar phases are employed in the system. Parameters of additional phases are stored in vector $\mathbf{B}$. The animator task is to decide which parameters of $\mathbf{B}$ should be used for achieving best results. In our method that task is accomplished by fuzzy inference.

Additional phases used by the animator are phases of *anticipation* (a motion preceding main transition between two poses adjacent in time), *overshoot* (a motion after transition, when motion stops slowly not abruptly), and *moving hold* (subtle changes of pose while actions stops, and a character should remain alive). Anticipation displays preparation for the action, e.g., squat before jump, overshoot portrays inertia, and moving hold is responsible for showing balancing and maintaining aliveness. Correct utilization of these additional phases influence naturalness and fluency of motion, related to high subjective quality of animation.

For simplification of the problem following assumption has to be made:

1. *anticipation* and *overshoot* are alike, i.e., their times (t) and limbs rotations amplitudes (A) are assumed to be equal, therefore for parameterization only two values are used: $dA = A_a = A_o$ and $dt = t_a = t_o$;
2. times and amplitudes of phases are limited and cannot extend beyond subjectively accepted values: $dt \in (0;10)$ [frames], $dA \in (0,0.265\pi)$ [rad];
3. *moving holds* are calculated as random movements, with times and amplitudes taken from limited ranges (this is not in the scope of the paper).

Motion parameters for single segment of animation[1] are presented in Fig. 1. As mentioned before, parameters related to poses, are stored in vector $\mathbf{A}$. Parameters connected to subjective features and related to additional motion phases are stored in vector $\mathbf{B}$:

$$\mathbf{A} = [V,A,t], \quad \mathbf{B} = [dA,dt], \tag{1}$$

---

[1] Animation segmentation performed at the preprocessing stage of the algorithm is not in the scope of the paper.

where

$$A = a_3 - a_2, \quad t = t_3 - t_2, \quad V = A/t . \tag{2}$$

3 Motion Data Processing

Traditional animation rules [1, 10] describe a way of additional phases utilization for achieving particular subjective results. Fast motion with big amplitude should be preceded by large anticipation and finished with large overshoot; long motion should be preceded with long anticipation and finished with long overshoot. Taking it into consideration it was assumed that proportionality occurs between these parameters, which can be described as:

$$dA = \alpha \times V \times A, \qquad dt = \beta \times V \times t, \tag{3}$$

where α and β are new proposed proportionality coefficients, which are discussed later. In case when dA or dt exceeds assumed maximal value[2], the following $f(x)$ and $g(x)$ saturation functions are used:

$$dA = f(\alpha \times V \times A), \quad \text{where} \quad f(x) = 0.265 \times \pi \times \tanh\left(\frac{x}{0.22\pi}\right), \tag{4}$$

$$dt = g(\beta \times V \times t), \quad \text{where} \quad g(x) = 10 \times \tanh\left(\frac{x}{0.125}\right) . \tag{5}$$

While considering (3), (4), variables dA and dt depend only on coefficients α and β. If dA and dt changes influence subjective meaning of animation then a relation should exist also between subjective features and α and β. These relations were specified during data mining of results of subjective test. In tests simple animations where used, containing two poses and a transition, with anticipation, overshoot and moving hold phases. The participants' task[3] was to name features of motion utilizing following discrete scales: *style*={*natural*, *middle*, *exaggerated*}; *fluidity*={*fluid*, *middle*, *abrupt*}; *quality*={1, 2, 3, 4, 5}. Results of *fluidity* and *style* evaluation are stored in vector **Q** = [*style*, *fluidity*], and *quality scores* are processed individually as an additional criterion **QS**. Evaluation of visual stimuli was performed with respect to recommendations [3, 2].

First, a correlation between α and β of each evaluated animation and its *style* and *fluidity* scores was calculated. The results are presented in Table 1. Strong correlation indicates that a certain connection between selected subjective feature and proposed coefficient exists, therefore rules describing that relation can be created.

---

[2] It was verified during visual tests what maximum values of amplitudes dA and times dt are subjectively accepted by the viewers.

[3] Participants were accustomed with the research, and trained by watching possible variations of the movement, commented by the authors.

Table 1 Correlation coefficients between subjective and objective parameters of animations

	β-style	β-fluidity	β-quality	α-style	α-fluidity	α-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

3.1 Relations between Animation Features

The correlation described above is used for creating rules that connect objective and subjective parameters of animation. *Ambiguous* information about that relations is obtained during subjective evaluation tests, when participants for an animation described with **A** and **B** select values for subjective features **Q**. Particular animation described by $\mathbf{A} = [V, A, t]$ and $\mathbf{B} = [dA, dt]$ is not evaluated identically by every test participant. Projection from **B** to **Q** with given **A** is denoted as $f_\mathbf{A} : \mathbf{B} \to \mathbf{Q}$. That ambiguous function is called *evaluation function* and reflects viewers' answers. It is assumed that **A** and **Q** are constant and only **B** parameters are unknown variables. Therefore inverse function is being sought, $f_\mathbf{A}^{-1} : \mathbf{Q} \to \mathbf{B}$, which for given **A** and required values of features **Q** chooses correct **B**. That function is also ambiguous. For each **A** and **Q** a result is first generated as a set of objects – animations that have an action as the one defined in **A** and that were subjectively evaluated as having values matching given $\mathbf{Q} = [style, fluidity]$, but are differentiated by **B**. From this set one object is finally selected based on additional criterion – maximization of *mean quality score* **QS**. Therefore for any **A** it is possible to generate *unambiguous* rules connecting values of **Q** with **B**.

Equations (3) and (4) describe proportionality between values V, A, and t stored in **A**, and parameters of additional motion phases dA and dt stored in **B**. Coefficients of these proportionalities, i.e., α and β, are used for simplification of searching for $f_\mathbf{A}^{-1} : \mathbf{Q} \to \mathbf{B}$. The problem is first reduced to defining relations $\mathbf{Q} \to [\alpha, \beta]$, then based on (4) the inverse function $f_\mathbf{A}^{-1}$ is calculated for any given **A**. Obtained relations between subjective variables and proportionality coefficients are presented in Table 2.

Table 2 Calculation of α and β based on given *fluidity* and *style*

α		*fluidity*			β		*fluidity*		
		abrupt	middle	fluid			abrupt	middle	fluid
	natural	0.7	0.5	0.3		natural	3	5	7
style	middle	0.9	0.7	0.5	style	middle	1	5	5
	exaggerated	1.3	1.1	0.9		exaggerated	3	5	7

3.2 Generation of Rules

Based on knowledge gathered in subjective tests a set of rules was obtained, and implemented in fuzzy system (6), which replaces analytic (4):

$$IF\ V = \ldots \wedge t = \ldots \wedge style = \ldots \wedge fluidity = \ldots \ \text{THEN}\ dt = \ldots$$
$$IF\ V = \ldots \wedge A = \ldots \wedge style = \ldots \wedge fluidity = \ldots \ \text{THEN}\ dA = \ldots \tag{6}$$

For all parameters a *fuzzy membership functions* are required [6], therefore input parameters V, A, t are first discretized: $V = \{0.0, 0.05, 0.1, 0.15, \ldots, 0.4\}$, $A = \{0.1, 0.2, \ldots, 1.0\}$, $t = \{5, 10, 15, \ldots, 50\}$. Then calibration animations are prepared for evaluation, presenting an animated character arm motion with speed, amplitude or time chosen as one of the above discrete values. Features of these animations are rated utilizing linguistic descriptors: $speed = \{low, medium, high, very\ high\}$, $amplitude = \{low, medium, high, very\ high\}$, $time = \{short, medium, long, very\ long\}$. Based on evaluation results membership functions (mf) are created. For example for mf $speed = low$ as a kernel[4] a range of discrete values of feature *speed* for which participants selected linguistic value *low* more often than in 80% of cases is selected. Membership functions for variable amplitude are presented in Fig. 2a.

Crisp values obtained by calculation of (4) for all combinations of discrete values of input parameters V, A, t are also fuzzified. In the first step all crisp values were clustered using *k*-means algorithm, then triangle mfs were created, each having maximum in a center value of respective *k*th cluster, and spanning in a way that will fulfill sum-to-one condition, as presented in Fig. 2b.
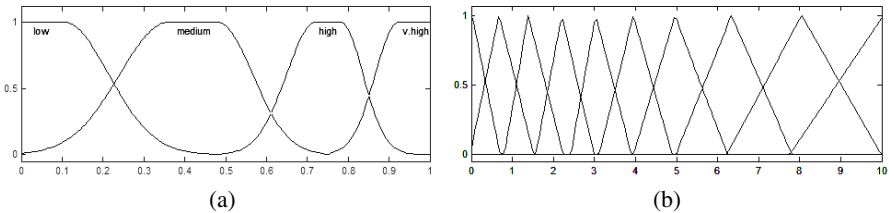

(a)   (b)

Fig. 2 Obtained membership functions for: **a** linguistic variable *amplitude*, **b** variable *dt*

Fuzzy rules (6) are formulated for all three linguistic values of variables *style* and *fluidity* and for all four values of discretized V, A, t. First for the given *style* and *fluidity* based on Table 2 coefficients α and β are calculated, then from (4) the outcome values are calculated which are finally fuzzified. Repeating that process for all combinations of input values all needed rules are being formulated. An example of rules obtained for calculation of fuzzy dA value is presented in Table 3.

The analysis of interpolation surfaces of fuzzy system (Fig. 3) reveals that the system is complete, i.e., a result exists for any combination of input parameters, and

---

[4] Values with membership equal 1.

Table 3 Rules for calculating fuzzy value of *dA* for given *natural style* and *fluid animation* (mf*n* – *n*th membership function for variable *dA*)

dA		V			
		low	medium	high	very high
	low	mf1	mf1	mf2	mf3
A	medium	mf2	mf2	mf3	mf5
	high	mf2	mf3	mf4	mf6
	v. high	mf2	mf3	mf4	mf7
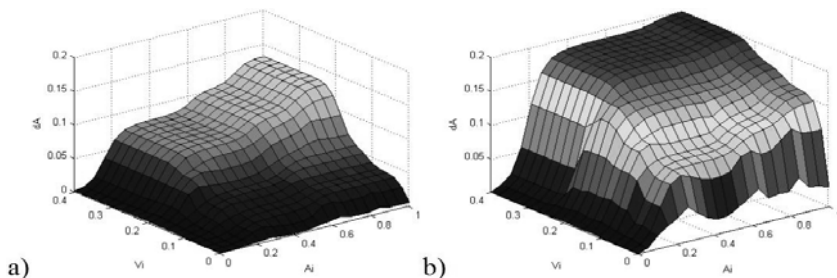
Fig. 3 Interpolation surfaces of fuzzy system – d*A* values depending on input *V* and *A* for: **a** natural style and fluid animation, **b** exaggerated style and fluid animation

changes of output value are fluid (continuous first derivative), which are important features for practical usage of the system.

4 Results

Effectiveness of animation processing system was verified in visual subjective test. Five prototype animated avatar actions served as the test material. These were enhanced using the system and all combinations of input descriptors for *style* and *fluidity*. All animations were rated using 5-point scale. For verification also

Table 4 Mean opinion score for animations depending on values of subjective descriptors used for style enhancement

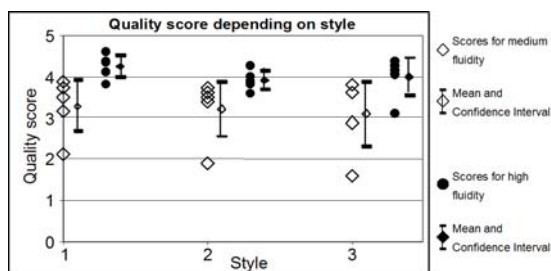| Mean opinion score		style		
		natural	medium	exaggerated
	abrupt	2.08	2.06	2.08
fluidity	medium	3.28	3.22	3.10
	fluid	4.24	3.92	4.02
non-processed			1.50	

Fig. 4 Quality scores depending on style for two levels of fluidity: medium and high. Scores and their means are shown. Style 1 is for *natural*, 2 – *medium*, 3 – *exaggerated*

non-processed versions of animations were displayed. Mean Opinion Score values for all animations are presented in Table 4.

Processed animations obtained statistically valid higher scores than non-processed. Moreover fluid motion was always rated higher than the abrupt one. Finally, variation of animation style does not had influence on quality scores (Fig. 4), therefore the method developed for animation creation can by applied to generation of many versions of a single prototype action, matching user requirements for style and fluidity.

5   Conclusions

Utilizing methodology and rules of traditional animation, combined with fuzzy processing, the system for animation enhancement was developed. In the system fuzzy rules are used for calculation of parameters of additional motion phases that should be inserted to animation for alteration of subjective features such as stylization and fluidity of motion. New proportionality coefficients α and β were defined that are strongly correlated with subjective features of animation.

References

1. Blair, P.: Cartoon Animation. Walter Foster Publishing, Laguna Hills (1995)
2. ITU Radiocommunication Assembly: Recommendation ITU-T P.800 (1996) (Methods for Subjective Determination of Transmission Quality)
3. ITU Radiocommunication Assembly: Recommendation ITU-R BT.500-11 (2002) (Methodology for the Subjective Assessment of the Quality of Television Picture)
4. Li, Y., Gleicher, M., Xu, Y., Shum, H.: Stylizing motion with drawings. In: Proceedings of SIGGRAPH, Los Angeles, US, pp. 309–319 (2003)
5. Neff, M., Fiume, E.: Artistically based computer generation of expressive motion. In: Proceedings of Symposium on Language, Speech and Gesture for Expressive Characters, pp. 29–39 (2004)
6. Pedrycz, W., Gomide, F.: Fuzzy Systems Engineering: Toward Human-Centric Computing. Wiley-IEEE Press, New Jersey (2007)

7. Remondino, F., Schrotter, G., Roditakis, A., D'Appuzo, N.: Markerless motion capture from single or multi-camera video sequence. In: Proceedings of the Interntional Workshop on Modelling and Motion Capture Techniques for Virtual Environments (2004)

8. Szczuko, P.: Application of fuzzy rules in computer animation. Ph.D. thesis, Gdańsk University of Technology, Poland (2008),
   http://sound.eti.pg.gda.pl/animacje

9. Szczuko, P., Kostek, B.: Analysis and generation of emotionally featured animated motion. In: Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Regina, Canada, pp. 333–341 (2005)

10. Thomas, F., Johnston, O.: Disney Animation – The Illusion of Life. Abbeville Press, New York (1981)

Multidimensional Labyrinth – Multidimensional Virtual Reality

Dariusz Jamroz

Abstract. In this paper, the author's method which allows checking the possibility of human acquisition of skills of comprehension and intentional movement in multidimensional space was presented. The method allows to check, whether a human is able to find an exit from the n-dimensional labyrinth (for any finite $n \geq 3$). It can be achieved by virtual movements in multidimensional space. The man in front of the computer screen on which he sees a picture of the interior of a multidimensional labyrinth with the goal of exiting from this labyrinth may feel like a laboratory rat which has been placed in unknown and inexplicable reality for him. In this paper we describe the computer system created on the basis of the presented theory. It allows moving in the virtual labyrinth placed in four and five-dimensional space, is also presented. This program besides research purposes may be treated as the first multidimensional game.

Keywords: multidimensional visualization, virtual reality.

1 Previous Work

The first author dealing with the visualisation of multidimensional blocks, was Noll [7]. He presented four-dimensional blocks in the form of wire-frames. Using a plotter he generated three-dimensional stereoscopic movies of the three-dimensional orthogonal and perspective projections of four-dimensional hyper-objects rotating in a four-dimensional space. Dewdney [3] and Banchoff [1] presented pictures of a four-dimensional block and a four-dimensional sphere (presented in the form of a set of three-dimensional tori). Another visualisation method was the presentation of four-dimensional objects in the form of three-dimensional slices [1]. Steiner and Burton [8] and Carrey et al. [2] used the scan-plane conversion to render four-dimensional

Dariusz Jamroz
AGH University of Science and Technology,
Mickiewicza Av. 30, 30-059 Cracow, Poland
e-mail: `jamroz@agh.edu.pl`

objects into three-dimensional voxel fields. Those fields were then projected on a two-dimensional screen. Hanson and Heng [4] rendered points, curves and surfaces in four dimensions.

Jamroz [5, 6] presented the method which is not a simple projection.It generates an image that allows the external appearance of the solid to be seen, in a form similar to its appearance in three-dimensional reality. The advantage of the method is that it makes possible to obtain pictures of the individual faces of a solid and of an arbitrarily large number of dimensions, including their inclination. This information is visualised in a natural way – by differences in the brightness of the faces. Using this visualisation method for three dimensions we obtain familiar views of the three-dimensional solids. This method is thus a generalisation of our natural perception of three-dimensional space into any number of dimensions.

2   Creation of Labyrinth View

The method presented in works of Jamroz [6] and [5] was applied to present the view of multidimensional labyrinth on screen. It was enlarged to present not only the single convex blocks, but also objects consisted from many convex blocks.

Computer screen is represented by a discreet grid of points that can be presented on the graphic screen. For each point of the grid (with the co-ordinates β_1, β_2 in relation to the grid) we calculate its distance ψ in *direction* $\mathbf{r}$ from each *hypersurface* $S_{(\mathbf{s},\mathbf{d})}$ containing *the face of the block*. It can be calculated by means of the following formula:

$$\psi = \frac{(\mathbf{w} + \beta_1 \mathbf{p_1} + \beta_2 \mathbf{p_2} - \mathbf{s}, \mathbf{d})}{(\mathbf{r}, \mathbf{d})} \,, \tag{1}$$

where $S_{(\mathbf{s},\mathbf{d})} \overset{\text{def}}{=} \{\mathbf{x} \in X : (\mathbf{x} - \mathbf{s}, \mathbf{d}) = 0\}$, $Z_{(\mathbf{s},\mathbf{d})} \overset{\text{def}}{=} \{\mathbf{x} \in X : (\mathbf{x} - \mathbf{s}, \mathbf{d}) \geq 0\}$.

It allows us to determine which *face* of the *block* is the nearest one, i.e., which one, when looking in *direction* $\mathbf{r}$ from a given point on *the observational plane,* shades the others and therefore is visible.

The fact that a given *hypersurface* is the nearest in *direction* $\mathbf{r}$ is not sufficient; it is also necessary to determine whether the point $\mathbf{a}$ on the *hypersurface* visible in *direction* $\mathbf{r}$ is a part of the *face* of the *block*, i.e., of the fragment of the *hypersurface* which belongs to *the solid*. It is possible to do this using the definition of the *solid*, by verifying whether point $\mathbf{a}$ is a part of each half-space that forms the solid. Thus, it is enough to state whether:

$$(\mathbf{a} - \mathbf{s_i}, \mathbf{d_i}) \geq 0 \tag{2}$$

is valid for $\forall Z_{(\mathbf{si},\mathbf{di})}$, where $i = 1, \ldots, k$ and $Z_{(\mathbf{s1},\mathbf{d1})}, Z_{(\mathbf{s2},\mathbf{d2})}, \ldots, Z_{(\mathbf{sk},\mathbf{dk})} - half\text{-}spaces$ which contain *the convex block* Y. If for the certain screen point, the distance ψ will be calculated for each *convex block* being in this space then the minimum of those distances will be the distance from the shape visible in certain point of the screen.

Finally, there is one more problem to consider: in what way is light reflected by the *face* of the *convex block* Y? It has been assumed that when looking in *direction* $\mathbf{r}$ the brightness of light emitted by the point on the *hypersurface* $S_{(\mathbf{s},\mathbf{d})}$ is directly
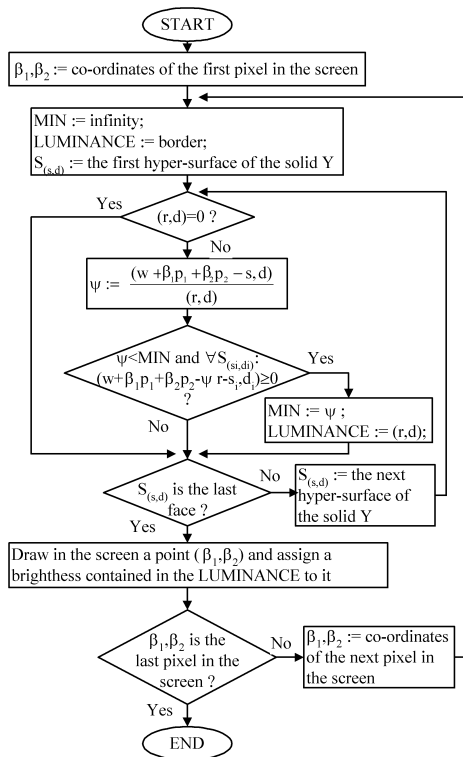
Fig. 1 Diagram of the procedure of drawing blocks in any number of dimensions

proportional to the value of the scalar product $(\mathbf{r}, \mathbf{d})$. The procedure described above will be called by the system with each change of one of vectors $\mathbf{w}, \mathbf{p_1}, \mathbf{p_2}, \mathbf{r}$. The algorithm of the drawing procedure is shown in Fig. 1. It is the same for any number of dimensions, n, where $n \geq 3$. Only the method of calculating the scalar product is dependent on the number of dimensions. It is calculated from the formula:

$$(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} x_i y_i \,, \tag{3}$$

where: $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$.

3 Labyrinth

To create the multidimensional labyrinth, we applied the simplest rules because the analysis of shape correctness for even the simplest multidimensional objects is difficult. For a complex object representing a labyrinth it is even more complicated. It

was assumed, that the created object is build from the *n*-dimensional cubes. They adjoin to the others in every direction constituting in that way a bigger cube. For the space of, for example, 4 dimensions and placing six cubes in each direction, the block will consist of $6 \times 6 \times 6 \times 6$ of cubes. In such block, the corridors can be created by removing some cubes being the part of this block. By making certain corridors, such block may show labyrinth.

Each cube, for example of 4 dimensions, is described by 8 faces. Each face in this space is represented by two vectors. Each vector in such space is described, of course, by 4 co-ordinates. Amount of memory needed to model such a block representing labyrinth is $6^4 \times 8 \times 2 \times 4 = 82\,944$ numbers. For the space of 5 dimensions it will be, respectively, $6^5 \times 10 \times 2 \times 5 = 777\,600$ numbers.

4 Results

Basing on the method described above, the computer program was created, allowing to virtually move into the labyrinth placed in space of 4 and 5 dimensions. The assumed way of moving in multidimensional space was presented in the paper [6]. To accelerate the program's operation, we applied distracted calculations and used the computational power of NVidia multicore accelerators. Furthermore, the program allows also simultaneous calculations on many computers connected in internet and showing the collected data in real time on the screen of one of them. Such acceleration is necessary, because, for example, one image of 4-dimensional labyrinth is calculated by one core processor 3.2 GHz during 37.3 seconds, by four core
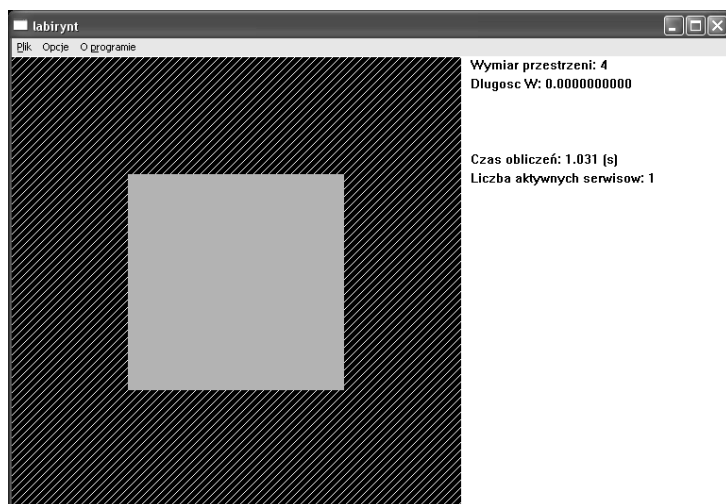

Fig. 2 Application window showing entrance to the labyrinth in 4-dimensional space
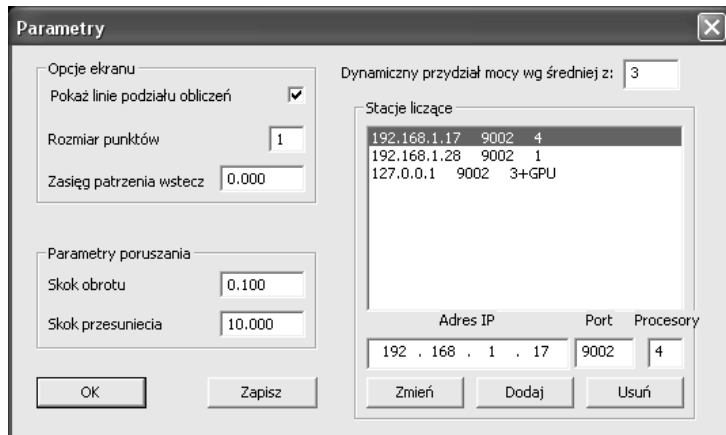
Fig. 3 Program options allowing setup of processor cores number, allocation of calculations to multicore NVidia accelerator and allocation of calculations for many computers in net
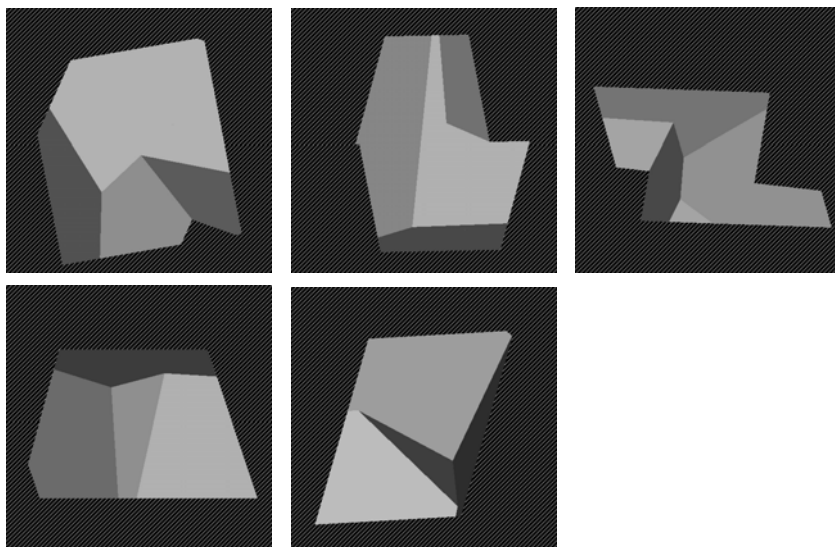

Fig. 4 Example of the 4-dimensional labyrinth interiors view

processor 3.2 GHz is calculated during 9.34 seconds and by 128 cores of accelerator NVidia is calculated in 1.32 seconds. Figures 4–5 show the given examples of the labyrinth interiors views.
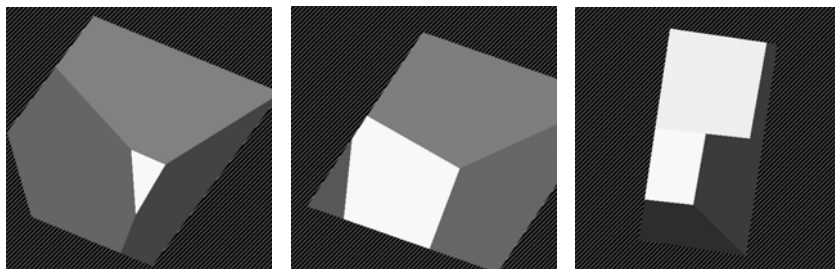
Fig. 5 Example of the 4-dimensional labyrinth exit view (white part is the exit)

5 Conclusions

The method described in this paper allows carrying on research over the possibilities of the adaptation of the human brain to receive and understand the new kind of information. People are not taught and are not accustomed to receive and understand spaces of more than 3 dimensions. Therefore it should be ascertained whether we can learn to think in terms of higher dimensional states.

The possibility of creating multidimensional computer games should be considered. Such games may become much more interesting than the 3D games; just as today 3D games give more possibilities than the older 2D ones. The 'multidimensional labyrinth' program presented in this work besides research purposes may be treated as the first game of this kind. It is possible that the idea of virtual reality may considerably change its meaning by extension of this reality from 3 into any number of dimensions.

References

1. Banchoff, T.F.: Beyond the third dimension. Scientific American Library, New York (1990)
2. Carey, S.A., Burton, R.P., Campbell, D.M.: Shades of a higher dimension. Computer Graphics World, 93–94 (1987)
3. Dewdney, A.K.: Computer recreations – a program for rotating hypercubes induces four-dimensional dementia. Scientific American, 14–23 (1986)
4. Hanson, A.J., Heng, P.A.: Illuminating the fourth dimension. IEEE Computer Graphics & Applications 12(4), 54–62 (1992)
5. Jamroz, D.: Looking in the n-dimensional space. Computer Science 3, 95–105 (2001) (in Polish)
6. Jamroz, D.: Visualization of objects in multidimensional spaces. Ph.D. thesis, AGH University of Science and Technology, Cracow, Poland (2001) (in Polish)
7. Noll, M.A.: A computer technique for displaing n-dimensional hyperobjects. Communications of the ACM 10(8), 469–473 (1967)
8. Steiner, K.V., Burton, R.P.: Hidden volumes: The 4th dimension. Computer Graphics World 10(2), 71–74 (1987)

Shape Recognition Using Partitioned Iterated Function Systems

Krzysztof Gdawiec

Abstract. One of approaches in pattern recognition is the use of fractal geometry. The property of the self-similarity of the fractals has been used as feature in several pattern recognition methods. In this paper we present a new fractal recognition method which we will use in recognition of 2D shapes. As fractal features we used Partitioned Iterated Function System (PIFS). From the PIFS code we extract mappings vectors and numbers of domain transformations used in fractal image compression. These vectors and numbers are later used as features in the recognition procedure using a normalized similarity measure. The effectiveness of our method is shown on two test databases. The first database was created by the author and the second one is MPEG7 CE-Shape-1PartB database.

Keywords: shape recognition, iterated function, self-similarity of fractals.

1 Introduction

Image recognition is one of the most diverse areas of machine vision. The aim of object recognition is to classify unknown images or areas of images, known as objects using known objects. In general, all objects in a known class have parameters extracted from them and these parameters are used to classify unknown objects. An ideal image recognition technique would be robust to changes in scale, rotation, illumination effects and noise while being fast to implement. Unfortunately, such a technique does not exist.

One of approaches in pattern recognition is the use of fractal geometry. Fractal geometry breaks the way we see everything, it gives us tools to describe many of the natural objects which we cannot describe with the help of classical Euclidean geometry. The property of the self-similarity of fractals was used as feature in several pattern recognition methods. The fractal recognition methods found applications in face recognition [2, 5], signature verification [4], character recognition [7], gait

Krzysztof Gdawiec
Institute of Mathematics, University of Silesia,
Bankowa 14, 40-007 Katowice, Poland
e-mail: kgdawiec@math.us.edu.pl

recognition [10] or as a general recognition method [8] [9]. Most of these methods as fractal features use Partitioned Iterated Function Systems (PIFS) and some of them fractal dimension. In this paper we present a new fractal recognition method which we will use in the recognition of 2D shapes. As fractal features we used PIFS. From the PIFS code we extract mappings vectors and numbers of domain transformations used in fractal image compression. These vectors and numbers are later used as features in the recognition procedure.

First we introduce the notion of a fractal (fractal as attractor [1]) which we will use in this paper and some basic information about fractals. Next, we briefly present fractal image compression [3], which gives us the PIFS code used later in the recognition process. Then, we present our fractal recognition method of 2D shapes which is based on mappings vectors and numbers of domain transformations obtained from PIFS code. The effectiveness of our method is shown on two test databases. First database was created by the author and the second one is MPEG7 CE-Shape-1 Part B database.

2   Fractals

The notion of a fractal differs from many others mathematical notions. It has several nonequivalent definitions, e.g., as attractor [1], as an invariant measure [1]. So firstly, we introduce the definition which we will use in this paper. Next, in this section we briefly present fractal image compression method.

2.1   *Fractal as Attractor*

Let us take any complete metric space (X, ρ) and denote as $\mathscr{H}(X)$ the space of nonempty, compact subsets of X. In this space we introduce a metric $h : \mathscr{H}(X) \times \mathscr{H}(X) \to \mathbb{R}_+$ which is defined as follows

$$h(R, S) = \max\{\max_{x \in R} \min_{y \in S} \rho(x, y), \max_{y \in S} \min_{x \in R} \rho(y, x)\}, \tag{1}$$

where $R, S \in \mathscr{H}(X)$.

The space $\mathscr{H}(X)$ with metric h is a complete metric space [1].

Definition 1. *A transformation* $w : X \to X$ *on a metric space* (X, d) *is called a* contraction mapping *if there exists a constant* $0 \le s < 1$ *such that for all* $x, y \in X$:

$$d(f(x), f(y)) \le sd(x, y). \tag{2}$$

Any such number s is called a contractivity factor for w.

Definition 2. *We say that a set* $W = \{w_1, \ldots, w_N\}$, *where* w_n *is a contraction mapping with contractivity factor* s_n *for* $i = 1, \ldots, N$ *is an* iterated function system *(IFS).*

So defined IFS determines the so-called Hutchinson operator which is defined as follows:

$$\forall_{A \in \mathscr{H}(X)} W(A) = \bigcup_{n=1}^{N} w_n(A) = \bigcup_{n=1}^{N} \{w_n(a) : a \in A\}. \tag{3}$$

The Hutchinson operator is a contraction mapping with contractivity factor $s = \max\{s_1, \ldots, s_N\}$ [1]. Let us consider the following recurrent sequence:

$$\begin{cases} W^0(A) = A \,, \\ W^k(A) = W(W^{k-1}(A)), \quad k \geq 1 \,, \end{cases} \tag{4}$$

where $A \in \mathscr{H}(X)$.

The next theorem is the consequence of the Banach Fixed Point Theorem [1].

Theorem 1. *Let (X, ρ) be a complete metric space and $W = \{w_1, \ldots, w_n\}$ be an IFS. Then exists only one set $B \in \mathscr{H}(X)$ such that $W(B) = B$. Furthermore, the sequence defined by (4) is convergent and for all $A \in \mathscr{H}(X)$ we have $\lim_{k \to \infty} W^k(A) = B$.*

Definition 3. *The limit from Theorem 1 is called an* attractor *of the IFS or fractal.*

2.2 Fractal Image Compression

The fractal image compression method uses the fact that every image has a partial self-similarity. So we use additional notion of a Partitioned Iterated Function System [3].

Definition 4. *We say that a set $P = \{(F_1, D_1), \ldots, (F_N, D_N)\}$ is a* Partitioned Iterated Function System *(PIFS), where F_n is a contraction mapping, D_n is an area of an image which we transform with the help of F_n for $n = 1, \ldots, N$.*

In practice, as the contraction mappings we use affine transformations $F : \mathbb{R}^3 \to \mathbb{R}^3$ of the form:

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} a_1 & a_2 & 0 \\ a_4 & a_5 & 0 \\ 0 & 0 & a_7 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} a_3 \\ a_6 \\ a_8 \end{bmatrix}, \tag{5}$$

where coefficients $a_1, a_2, a_3, a_4, a_5, a_6 \in \mathbb{R}$ describe a geometric transformation and coefficients $a_7, a_8 \in \mathbb{R}$ are responsible for the contrast and brightness.

The compression algorithm can be described as follows. We divide an image into a fixed number of non-overlapping areas of the image called range blocks. We create a list of a domain blocks. The list consist of overlapping areas of the image, larger than the range blocks (usually two times larger) and transformed using the following mappings:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \tag{6}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \tag{7}$$

These four mappings are transformations of the rectangle (identity, $180°$ rotation and two symmetries of the rectangle). Next, for every range block R we look for the domain block D so that the value $\rho(R, F(D))$ is the smallest, where ρ is a metric, F is a transformation determined by the position of R and D, the size of these in relation to itself and one of the four mappings defined by (6, 7) and coefficients a_7, a_8 are calculated with the help of (8) and (9) (this is the most time-consuming step of the algorithm):

$$a_7 = \frac{k\sum_{i=1}^{k} g_i h_i - \sum_{i=1}^{k} g_i \sum_{i=1}^{k} h_i}{k\sum_{i=1}^{k} g_i^2 - (\sum_{i=1}^{k} g_i)^2}, \tag{8}$$

$$a_8 = \frac{1}{k}\left[\sum_{i=1}^{k} h_i - s_n \sum_{i=1}^{k} g_i\right], \tag{9}$$

where $g_1, \ldots, g_k$ are the pixel intensities of the transformed and resized domain block, and $h_1, \ldots, h_k$ are the pixel intensities of the range block. If $k\sum_{i=1}^{k} g_i^2 - \left(\sum_{i=1}^{k} g_i\right)^2 = 0$, then $a_7 = 0$ and $a_8 = \frac{1}{k}\sum_{i=1}^{k} h_i$.

3   Mapping Vectors Similarity Method

In this section we introduce a new fractal recognition method which is based on the mapping vectors and numbers obtained from transformations of the PIFS. But first, we introduce some observation with the help of which we get a better results of the recognition.

Let us take a look at fractal compression from the point of view of domain block D. Further, assume that this block fits to several range blocks (Fig. 1a). Each of these fits correspond to one mapping in PIFS. Now let us suppose that block D was changed into block D' (e.g., the shape was cut or it was deformed). This situation is shown in Fig. 1b. In this case domain D' can fit to the same range blocks as D (to all or only to some), it can also fit to some other range blocks. This change of fitting causes a change of the mapping in PIFS. In the worst case all mappings can be changed.

Now we divide the image into several non-overlapping sub-images, e.g., into 4 sub-images (Fig. 2) and compress each of the sub-images independently. And again let us consider the same domain block D and the same range blocks. This time block D fits only to the range blocks from the same sub-image, the other range blocks from different sub-images fit to other domain blocks D_1, D_2 (Fig. 2a). Now suppose that block D was changed in the same way as earlier (Fig. 2b). The change of the block only has an influence on the sub-image in which the block D' is placed. The fitting in other sub-images does not change. This time in the worst case only mappings corresponding to this sub-image change, all the other mappings remain the same. So, locally change of the shape has only a local influence on the transformations of PIFS and not global one as in the previous case.
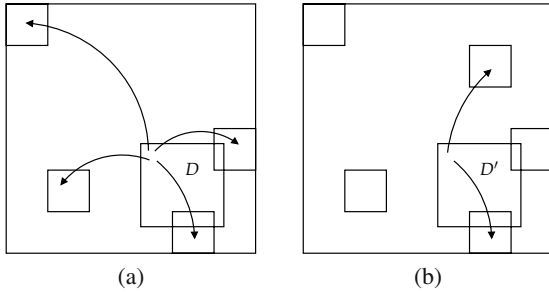
Fig. 1 Fractal image compression. **a** Starting situation. **b** Situation after the change of D into D'
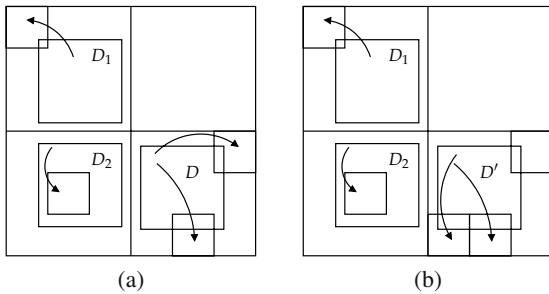

Fig. 2 Fractal image compression with division. **a** Starting situation. **b** Situation after the change of D into D'

Now we are ready to introduce the mapping vectors similarity method. First, we set the partition of the image into sub-images and the number of mappings of each PIFS. The method looks as follows:

1. binarize the image and extract the object,
2. give the object a correct orientation,
3. find a normalized PIFS W, i.e., for which the space is $[0, 1]^2$, for all sub-images,
4. for each PIFS V from the base:

   a. calculate mapping vectors similarity $\mathbf{s}=[s_1, \ldots, s_N]^T$ and vector $\mathbf{t}=[t_1, \ldots, t_N]^T$,
   b. calculate $e_V = \|[\begin{smallmatrix} s \\ t \end{smallmatrix}]\|$,

5. choose an object from the base for which the value e_V is the smallest.

Giving the object a correct orientation is a process in which we rotate the object so that it fulfills the following conditions: area of the bounding box is the smallest, height of this box is smaller than the width and left half of the object has at least as many pixels as the right. Figure 3 presents examples of giving an object a correct orientation. In the case of the triangle (Fig. 3(b)) we see three different orientations. If we add such an object to the base for each orientation we find the corresponding
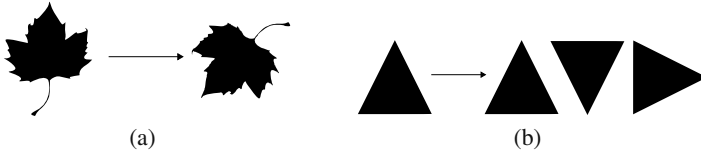
(a) (b)

Fig. 3 Examples of giving the object a correct orientation

PIFS and add it to the base. In the case of recognition we simply choose one of the orientations.

It remains to tell how to compute the vectors s and t. Firstly, for each of mapping w_n which belong to the PIFS W we take the corresponding mapping v_n from PIFS V. Next, we extract from them the mappings vectors (vector between the range block and corresponding domain block) and mappings used to transform the domain block onto range block (mappings given by (6) and (7)). Then

$$s_n = 100 - sim(p_n, q_n) = 100 - 50(1 + \cos\alpha)\frac{\min(\|p_n\|, \|q_n\|)}{\max(\|p_n\|, \|q_n\|)}, \qquad (10)$$

where p_n and q_n are the mapping vectors for w_n and v_n respectively, and α is the angle between vectors p_n and q_n. Next,

$$t_n = \eta(f_n, g_n) = \begin{cases} 0 & \text{, if } f_n = g_n \\ 1 & \text{, if } f_n \neq g_n \end{cases}, \qquad (11)$$

where f_n and g_n are mappings given by (6) and (7) corresponding to mappings w_n and v_n, respectively.

4 Experiments

Experiments were performed on two databases. The first database was created by the author and the second base was MPEG7 CE-Shape-1 database [6].

Our base consists of three datasets. In each of the datasets, we have 5 classes of objects, 20 images per class. In the first dataset we have base objects changed by elementary transformations (rotation, scaling, translation). In the second dataset we have objects changed by elementary transformations and we add small changes to the shape locally, e.g., shapes are cut and/or they have something added. In the last, the third set, similar to the other two sets the objects were modified by elementary transformations and we add to the shape large changes locally.

The MPEG7 CE-Shape-1 Part B database consists of 1400 silhouette images from 70 classes. Each class has 20 different shapes. Figure 4 presents some sample images from the authors base and the MPEG7 CE-Shape Part B base.
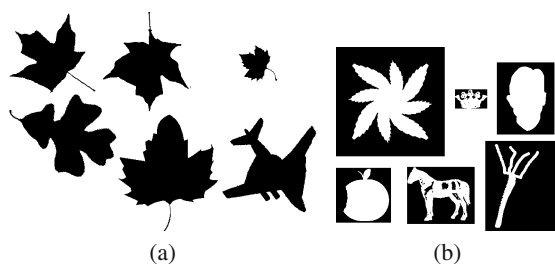
Fig. 4 Sample images: **a** authors base, **b** MPEG7 base

Table 1 Results of tests for the authors base

(a) elementary		(b) locally small		(c) locally large	
Partition	Error [%]	Partition	Error [%]	Partition	Error [%]
1×1	3.00	1×1	6.00	1×1	14.00
2×2	2.00	2×2	4.00	2×2	9.00
4×4	1.00	4×4	1.00	4×4	8.00

In the test we used several different partitions into independent sub-images. One was a partition into 1×1 sub-images which is the classical case. The other partitions were 2×2 and 4×4. The number of transformations used in the fractal compression of each sub-image depends on the partition. For partition 1×1 we used 256 transformations (16×16 range blocks division), for 2×2 – 64 transformations (8×8) and 16 transformations (4×4) for the 4×4 partition.

To estimate error rate of our method we used leave-one-out method for the three datasets created by the author and for the MPEG7 CE-Shape-1 Part B base we used stratified 10-fold cross validation.

Tables 1a–1c present the results of the tests for the authors base and Table 2 presents the results for the MPEG7 CE-Shape-1 Part B base.

Table 2 Results of tests for the MPEG7 CE-Shape-1 Part B base

Partition	Error [%]
1×1	30.45
2×2	18.58
4×4	16.37

5 Conclusions

A new method for recognition of shapes has been presented in this paper. The method was based on fractal description of the shape. Moreover a modification of

the compression scheme was proposed which led to a significant decrease of the recognition error. The division into several independent sub-images also brings an improvement in speed of achieving the fractal description of the shape. This is due to the fact that in the case of dividing the image into sub-images and then compressing them, the list of the domain blocks on which we are doing the search process is smaller than in the classical case.

In our further work we will concentrate on taking into account the number of matching sub-images in the similarity measure, which may bring a further decrease of the recognition error. Moreover, we will perform tests with other partitions of the image into independent sub-images to see the influence of different divisions on the recognition rate. Furthermore, we will search for the optimal division into sub-images. We will also try to bring the division into sub-images into other known fractal recognition methods.

References

1. Barnsley, M.: Fractals Everywhere. Academic Press, Boston (1988)
2. Chandran, S., Kar, S.: Retrieving faces by the PIFS fractal code. In: Proceedings of the 6th IEEE Workshop on Applications of Computer Vision, pp. 8–12 (2002)
3. Fisher, Y.: Fractal Image Compression: Theory and Application. Springer, New York (1995)
4. Huang, K., Yan, H.: Signature verification using fractal transformation. In: Proceedings of the International Conference on Pattern Recognition, vol. 2, pp. 855–858 (2000)
5. Kouzani, A.Z.: Classification of face images using local iterated function systems. Machine Vision and Applications 19(4), 223–248 (2008)
6. Latecki, L.J., Lakamper, R., Eckhardt, T.: Shape descriptors for non-rigid shapes with a single closed contour. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 424–429 (2000)
7. Mozaffari, S., Faez, K., Faradji, F.: One dimensional fractal coder for online signature recognition. In: Proceedings of the International Conference on Pattern Recognition, pp. 857–860 (2006)
8. Neil, G., Curtis, K.M.: Shape recognition using fractal geometry. Pattern Recognition 30(12), 1957–1969 (1997)
9. Yokoyama, T., Sugawara, K., Watanabe, T.: Similarity-based image retrieval system using partitioned iterated function system codes. Artifical Life and Robotics 8, 118–122 (2004)
10. Zhao, G., Cui, L., Li, H.: Gait recognition using fractal scale. Pattern Analysis & Applications 10(3), 235–246 (2007)

Computer Vision Support
for the Orthodontic Diagnosis

Agnieszka Tomaka and Agnieszka Pisulska-Otremba

Abstract. The following paper presents the achievement reached by our joined teams: Computer Vision System Group (ZKSW) in the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences and Department of Orthodontics, Silesian Medical University. The cooperation began from the inspiration of late Prof. A. Mrózek. Computer Vision in supporting orthodontic diagnosis means all the problems connected with proper acquisition, calibration and analysis of the diagnostic images of orthodontic patients. The aim of traditional cephalometric analysis is the quantitative confirmation of skeletal and/or soft tissue abnormalities on single images, assessment of the treatment plan, long term follow up of growth and treatment results. Beginning with the computerization of the methods used in traditional manual diagnosis in the simplest X-ray films of the patient's head we have developed our research towards engaging different methods of morphometrics, deformation analysis and using different imaging modalities: pairs of cephalograms (lateral an frontal), CT-scans, laser scans of dental models, laser scans of soft tissues, finally merging all the image information into patient's specific geometric and deformable model of the head. The model can be further exploited in the supporting of the surgical correction of jaw discrepancies. Our laboratory equipment allows us to design virtual operations, educational programs in a virtual reality with a CyberGlove device, and finally to verify the plan of intervention on stereolithographic solid models received from a 3D printer.

Keywords: computer vision, orthodontic diagnosis, image acquisition, calibration, merging information, virtual reality.

Agnieszka Tomaka
Institute of Theoretical and Applied Informatics, Polish Academy of Sciences
Bałtycka 5, 44-100 Gliwice, Poland
e-mail: ines@iitis.gliwice.pl

Agnieszka Pisulska-Otremba
Department of Orthodontics, Silesian Medical University,
Traugutta Sqare 2, 41-800 Zabrze, Poland
e-mail: orthapo@interia.pl

1 Introduction

In the beginning of the 90-ties we were inspired by late Prof. Adam Mrózek enthu-siasm to use computers to facilitate the work of medical specialists. He anticipated that the knowledge, obtained in the image processing, can successfully be used in medical diagnosis. This inspiration was so strong, that it determined our researches and now we feel obliged to present some results of our work. Prof. A. Mrózek de-veloped many fields of medical knowledge processing, but our cooperation arose around the supporting of the orthodontic diagnosis, using the X-ray images of the head – cephalograms. Since that time our cooperation is being developing.

2 Computerization of the Traditional Analysis

When we started, ZKSW had already had some experience in orthodontic appli-cations, but both teams were still learning each other, their jargons and the ways they understood the same problems. Some work already had been done to perform nasopharynx measurements [16]. The first attempt to measure deformation, using the Bookstein tensor method, was also implemented [15]. While the scanners for cephalograms were then unavailable, a special graphical card called MAVEC was designed in ZKSW [10]. It was successfully used for the acquisition of the X-ray images in angiography and ventriculography. Next we used it to acquire the im-age of cephalograms, which had to be done in standardized conditions, assuring the possibility to calibrate the system.

Our first project was concentrated around the computerization of the traditional orthodontic diagnosis (Fig. 1). The methods of Jarabak, Bjork, Sassouni cephalo-gram analyzes were implemented [12, 13]. The mechanism for defining new meth-
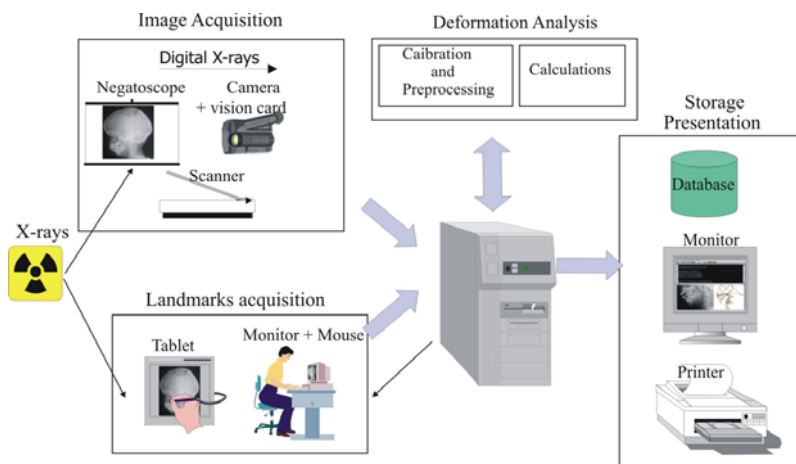

Fig. 1 Computerization of traditional orthodontic analysis

ods was also considered. Although the localization of landmarks had to be done by the user, the computerized analyzes did not require the laborious manual measurements, which were necessary during traditional analyzes, done with the use of tracing paper and calipers.

Traditional methods of orthodontic analysis, developed over decades of XX century, had introduced standard values for healthy individuals, as well as ranges for various syndromes. The most significant parameters were chosen and their proper values were established. The standard values included the gender and age of patients. It must be remembered, that the source of geometrical information for these tiresome and valuable works were only 2D projections, and therefore these standards can only be applied to 2D analysis.

3   From 2D to 3D Analysis

The subject of the orthodontic analysis is the interpretation of the craniofacial complex and particularly the skull. Traditional and most common diagnostic images, in early 90-ties of XX century, were cephalograms. X-rays are attenuated with a different degree by different body tissues, which enables the visualization of the inner layers, especially bones. But the image in mathematical sense is a projection. Therefore the enlargement of particular part of the object depends on its distance from the detector. The symmetric parts, i.e., of mandible in lateral (LL) projection are magnified by a different factor. For source-detector distance about 1.5 m these factors are about 1.034 for the nearest, 1.2 for the farthest layers [23]. The other inconvenience is that cephalograms are summation images. This superposition of shadows of particular parts of the head makes it difficult to reliably distinguish sides of the object. It had been proved that for simple diagnostic cases a 2D analysis is sufficient. But the cases of serious malformations like hemifacial microsomia, temporo-mandibular joint ankylosis need more detail 3D analysis.

3.1   3D Cephalogram

The idea of constructing a three-dimensional cephalogram was presented by Grayson in [6]. Its purpose was to reconstruct craniofacial geometry based on a pair of biorthogonal cephalograms (a lateral and a frontal films). If the geometry of X-ray imaging system is known, and if the landmarks can be reliably localized on both biorthogonal projections, then real 3D coordinates of these landmarks can be estimated [6, 23]. The problem with landmarks for the 3D cephalogram is connected with the fact, that the sets of landmarks used in 2D analyzes of both projections, overlap only partially. Some points, easily localized in one projection, cannot be localized in the other, some are known to lie on the edge, but their position on the edge is undetermined. However, the knowledge of geometry constraints, enables the construction of those points, as the intersection of the edge with the image of the ray from the second projection. This approach was successfully applied in Bolton Brush Growth
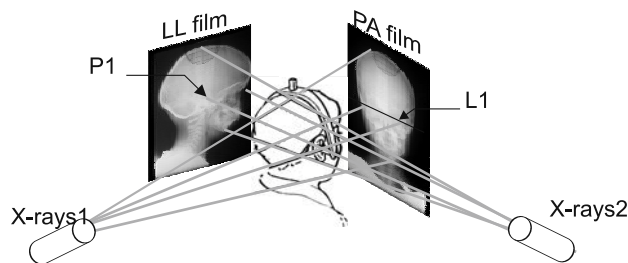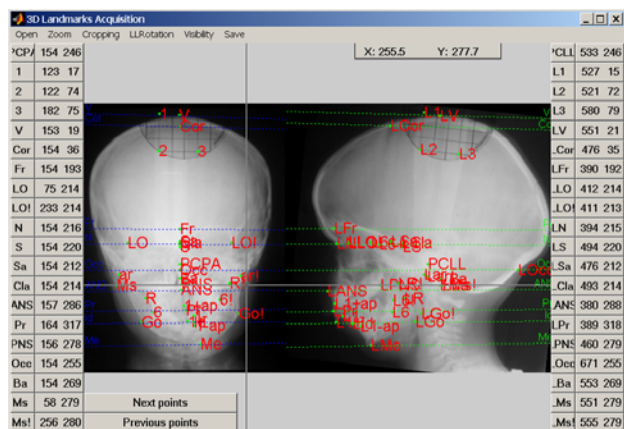
Fig. 2 The idea of a 3D cephalogram

Fig. 3 Software landmarks localizer

Study Center. They constructed a system, which was composed of a couple of bi-orthogonal, synchronized X-ray devices. The known parameters of X-ray imaging system geometry was fundamental for the construction of the hardware localizer of landmarks [3]. The software version of this localizer was implemented in ZKSW [23]. It had the possibility of landmarks localization also when each cephalogram of a pair was obtained by different X-ray devices, if only the geometrical parameters of both machines were known.

3.2 Stereovision Techniques to Reconstruct 3D Landmarks

Unfortunately, the cephalostat, a device for positioning of the patient's head, is not always used properly, which forces the need of the different corrections [23] and finally leads to the situation, that the mutual relations between two X-rays systems are unknown. In these situations stereovision techniques can be applied [26]. Eight landmarks reliably localized in both projections are sufficient to find a
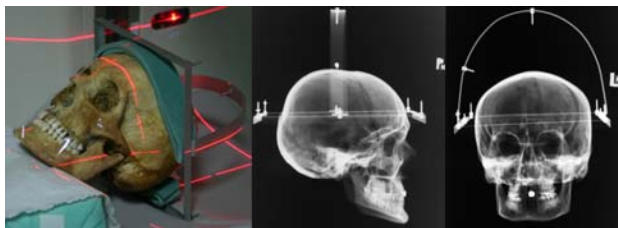
Fig. 4 The stereotactic frame and the X-rays of the skull

fundamental matrix, epipolar lines, next used by the localizer. Without the calibration the technique enables the 3D reconstruction up to the scale. The calibration requires an additional calibrating device – a stereotactic frame. Several prototype calibration devices were constructed and their application was tested both on dry skulls and on volunteers as well.

3.3 3D Reconstruction from CT Slices

Both stereovision techniques and 3D cephalogram methods yielded 3D coordinates of landmarks only. This was suitable for the analysis, but still the exact information about the shape of the skull and mandible was not accessible. The most complete information about the details of shapes of bones soft tissues and teeth can be obtained from a CT examination. This examination however is invasive and not necessary for each patient. Therefore only patients with considerable medical indications can be exposed to radiation, which is closely connected with CT scanning. From technical point of view, data from CT require additional processing like segmentation, isosurface construction, gantry correction and finally the acquisition of landmarks. The aim of the segmentation is to distinguish different tissues of the head (bones, soft tissues, teeth). In this process the values of attenuation coefficient, transformed to image intensities, are used assuming that the same tissue will have the same or similar attenuation coefficient in each point of the body. This assumption although leading to the simple thresholding, has the disadvantage that areas slightly differing from the threshold can be ignored, and the image has artificial holes. Algorithm of the isosurface reconstruction are developed by computer graphics. Most known are Marching Cube and Marching Tetrahedra algorithms [11], although there exist approaches combining segmentation and isosurface reconstruction. The segmentation algorithms and isosurface reconstruction act in the image coordinate system, therefore the results should be calibrated to match the patient's coordinate system. Especially it concerns the case of CT scanning with tilted gantry, which forces the analysis in the coordinate system with skewed basis. It has to be stressed that not all graphical applications (especially technical ones) are able to detect the necessity and perform appropriate correction [30].
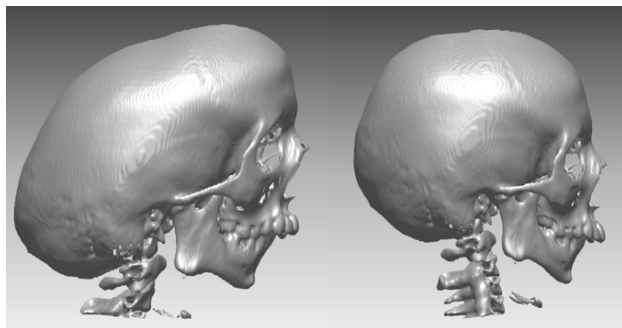
Fig. 5 Two individual CT imaging of the same patient (no.157),with and without gantry tilt, and 3D reconstructions without appropriate correction obtained in commercial application software

3.4 3D Analysis and Standard Values

Another problem with the analysis in 3D space is the choice of an appropriate set of landmarks and methods of landmarks positioning. The choice of landmarks from a technical point of view is of a secondary meaning, however it still remains the expectation that both sets of 2D and 3D analysis are related to each other. From the medical viewpoint it is very important which points are chosen for the analysis. As far as there are no established standards for the individual analysis all choices are equally valuable. The lack of standards for the 3D analysis results from the fact that this issue has developed for only several years. The approach in which the 3D landmarks are projected on the image plane in lateral or posterior-anterior projections then existing norms can be applied, is tiresome, but can be the first step to develop the 3D standards.

The standards are connected with the chosen method of analysis. Usually the distances between landmarks and inclinations of particular planes, created using
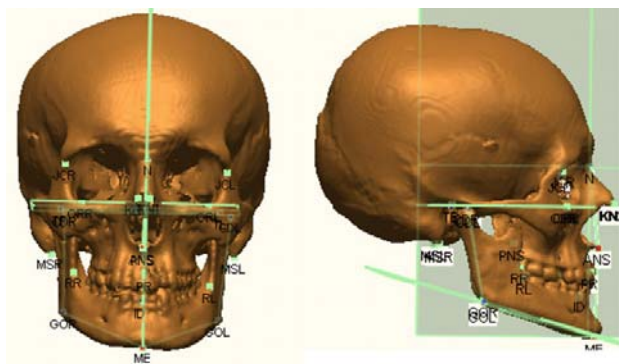

Fig. 6 The 3D analysis of the craniofacial complex

this landmarks, are analyzed. The mean values and variance are being estimated. The development of the methods of morphometrics allows to treat the analysis in a different way. Instead of measuring the 3D distances, the set of coordinates of landmarks can be treated as the configuration matrix and the methods of Procrustes superimpositions can be used [5]. This approach minimizes the distances between the corresponding landmarks for the sets of configuration matrices and after this minimization the mean shape can be established, as the mean location for each landmark. This approach, although described in the literature, is not used very often in everyday practice because of difficulty with medical interpretation.

4 Deformation Analysis

Single image is sufficient to perform measurements that described the shape of a skull in a single time moment. The comparison of two images taken in standardized conditions enables the measurement of shape change – deformation. From the medical point of view this deformation of the primal shape results from growth, malformation development, surgical interventions, or may be a compound of all these factors. It is difficult to anticipate the influence of growth on the results of the surgical treatment. Another difficulty is to distinguish the growth areas from those which remain unchanged, or are simply translated. Therefore the substance of analysis is the interpretation of the parameters of deformation, obtained from simple geometrical change of the shape, to conclude about the growth and treatment results in the craniofacial complex.

4.1 Bookstein Tensor Method

The first approach [15], implemented by our teams, was Bookstein tensor method [2]. The area of the image was divided into a triangle mesh and in each triangle the tensor of the dilatation was estimated. Bookstein described the graphical solution, acceptable to be performed manually. For the next implementation we developed the analytic algorithm for the measurement of the dilatation tensor [22]. The dilatation tensor is a basis for determination of the parameters of the deformation. We calculate the shape and size change coefficients and the directions of the greatest and the smallest change in the area of each triangle.

Although the graphical illustrations of the parameters of deformation seem to be convincing, the medical interpretation is not so straightforward. The parameters of the deformation highly depend on the assumed triangulation. Another problem is connected with the averaging of received parameters, and the method of determining the mean dilatation tensor for the corresponding triangles in analyzed groups of patients. The medical interpretation of the parameters of the deformation, determined using Bookstein tensor method, was the subject of publications and medical Ph.D. dissertations [14, 17, 21].
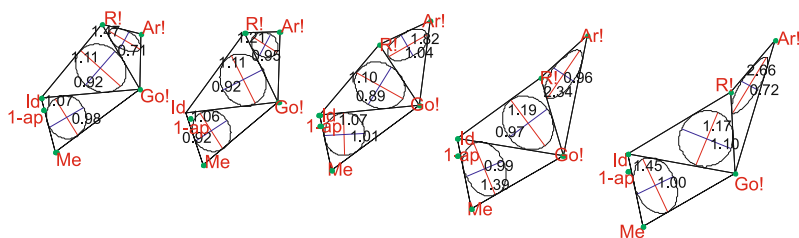
Fig. 7 Longitudinal measurements of parameters of the deformation based on the sequence of PA cephalograms of the same patient

4.2 Extension for 3D Analysis

A generalized form of the analytical solution of determining the tensor of dilatation, that we received [23], enabled the analysis to be extended to 3D. The idea still relied on the determining the affine transformation, but instead of triangles the tetrahedron simplex were used. In the 3D case there were three principal directions of dilatations and size change and shape change coefficients had to be redefined respectively. The similar issue of the medical interpretation and averaging were investigated for the set of patients with temporo-mandibular joint ankylosis [1]. The 3D landmarks coordinates were estimated using pairs of bi-orthogonal cephalograms, and methods of 3D cephalogram described above.

4.3 Multilevel Deformation Analysis

The dilatation tensor described above was the basis of the multilevel deformation analysis described in Ph.D. thesis [23] run primary under the supervision of late Prof. A. Mrózek, and after his early death of Prof. Konrad Wojciechowski. The idea of the work was the description of the deformation on different levels:

- global analysis,
- local analysis,
- interpolation of deformation parameters.

Global analysis searched for the global affine transformation, which would act on each analyzed set or subsets of landmarks. Therefore the parameters of such deformation could be treated as the overall characteristic of deformation. The parameters of deformation could be estimated for the whole analyzed object or for a particular part of it treated globally (Fig. 9).

The local analysis finds the affine transformation in each triangle and was described in see Sect. 4.1 and 4.3. The interpolation of deformation parameters is based on the estimation of the affine transformation in each point of the object. This affine transformation is calculated as the Jacoby matrix of interpolating function. The interpolation is performed using thin – plate spline function and its extension to
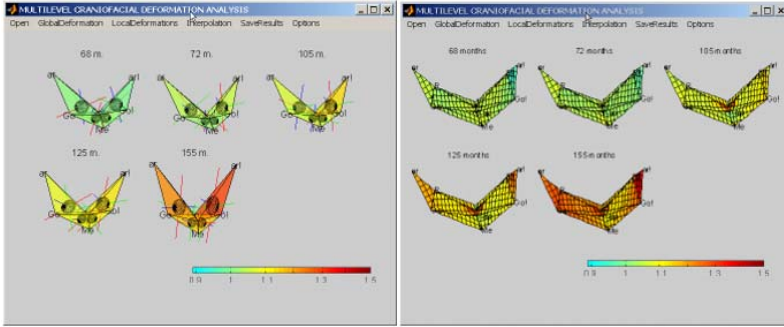
Fig. 8 Left: longitudinal study of dilatation tensor based on the sequence of pair of cephalograms. The size change coefficient for each tetrahedron is visualized by color, Right: longitudinal interpolation of size change coefficient
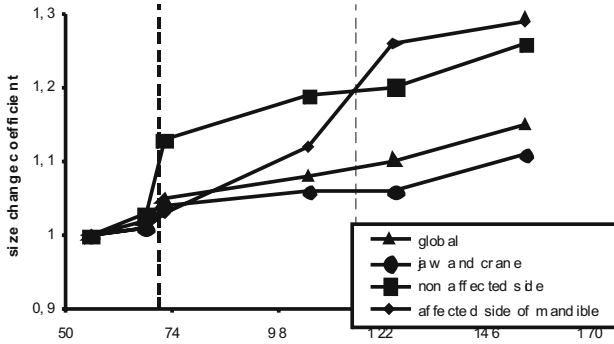
Fig. 9 Estimation of global size change coeff. for skull, upper jaw, and parts of mandible

3D. Therefore the field of the deformation tensor is estimated showing the local tendency of changes (Fig. 8). Described method is can be very useful for longitudinal studies [25].
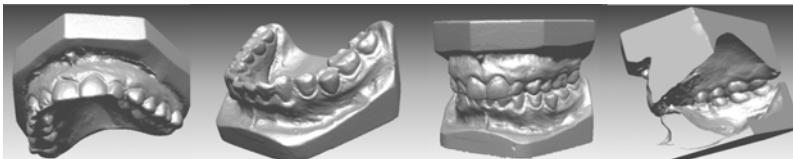

Fig. 10 Left to right: scan of upper model, of lower model, model in occlusion, the two separate scans combined and cut (virtually) to expose internal detail

5 New Imaging Techniques

The development of our Laboratory creates new possibilities addressing orthodontic
diagnosis. New, noninvasive techniques are able to digitize the information, avail-
able so far only in CT examination. Since 2003 we have used a Konica Minolta
VI-9i laser scanner, and since 2008 the 3dMD face scanner. We are able to perform
scanning of different objects, but for orthodontic purposes the most important is
scanning of the dental models and human faces.

5.1 Digital Dental Models

An additional equipment of Konica Minolta scanner – a rotary table facilitates the
scanning of small objects. The scans, acquired from many viewpoints are initially
aligned (registered) in a common coordinate system. Using this instruments we de-
veloped the technique of digitization of dental models. Each, the upper and the lower
model are first scanned separately, and then in occlusion, which is ensured by the
usage of a wafer. From the partial scans, after some registration and merging pro-
cedures the digital model with occlusion is obtained. Such models can be used both
for analysis and documentation purposes.

5.2 3D Photography

Our system for 3D head scanning consists of a Minolta Vivid VI-9i scanner moving
around the patient and a special chair with a headrest, that preserves breathing-
related movements of patient's head and assures unchanged head position during
the examination. The scanner can only scan those surfaces which are visible from a
given viewpoint. In order to acquire entire relevant surfaces of the head it must be
moved around the patient. The scans should be merged together after being brought
into register (transform into common coordinate system), The automation of this
process is described in [18, 28]. Konica Minolta scanner has an accuracy about
0.008 m, but the scanning process last about 20 s for each viewpoint, which is quite
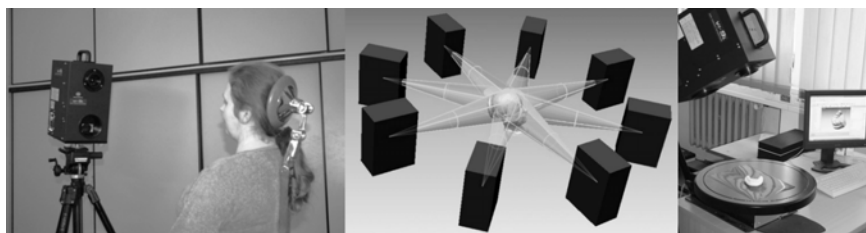long for scanning a patient's especially children's face. It is then highly probable

Fig. 11 Left to right: Scanning a person with the VI-9i; eight viewpoints for an all-round
image; scanning a dental model

Fig. 12 Qualitative evaluating facial symmetry. Various aspects of the patient's malformations become visible in different positions
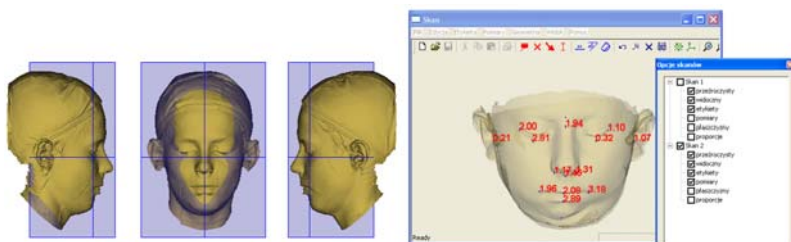

Fig. 13 Patient's reference plane. The distances between corresponding landmarks for two scans aligned in baseline superimposition

that the patient will change the appearance of the face between two scans performed from different viewpoints. Another disadvantage of this method is also the texture of soft tissues, which highly depends on the illumination. Therefore, after the procedure of merging the texture can be disturbed in such a degree, which makes it useless in the landmark identification process.

The better solution is offered in the 3dMD system. Two pairs of stereo cameras, additionally equipped with the color cameras for texture and projectors enables the capturing of a 3D photo in a very short time. The overall accuracy of the system is not so high as in Minolta scanner, the resolution is also lower but the short acquisition time and good quality of texture makes it a good alternative to laser scanners.

5.3 Soft Tissue Analysis

The laser scanner only scans the external surfaces of a head – the face. It is the shape, which the patient sees in the mirror and what often makes him seek medical help. Therefore scans provide the information about facial esthetics and can serve as the information for facial feature analysis. This analysis is performed intuitively by an expert during every patient examination. Scanning enables the quantitative analysis which does not require the presence of the patient and can be performed in any moment. One can measure the face morphology, its proportions, asymmetry and degree to which the face departs from the standard [24]. Additionally scans can be used to evaluate the results of treatment and growth changes, by comparing two scans, and to perform longitudinal and statistical studies. The special system for

the landmarks localization and analysis of facial features was implemented in our laboratory [19] and described in [20].

6 Integration of Different Imaging Modalities

Each imaging modality obtains the information about different parts of the skull, face and oral area. The reconstruction of skeletal surfaces can be done using data acquired from CT, while images of soft tissues are obtained by MRI [4], external surfaces of face and dental models can be acquired using laser scanner. The diagnostic value of these types of images can be dramatically increased if they are brought into register, i.e., put together in a common coordinate system [7].

6.1 Registration

Registration, understood as a process of aligning data from different imaging modalities and putting them together in a common coordinate system, is quite a large challenge. There exist many different approaches to register medical images and they use different technical and medical premises.

One, the most popular medical registration is establishing the same patient's specific coordinate systems for each image and aligning those coordinate systems. For orthodontic purposes usually the horizontal Frankfort, and midsagital planes are used [4]. This type of registration can be found in morphometric literature as baseline superimposition [5]. The locations of particular landmarks in both images then are interpreted as the movement in the established coordinate system. As such they allow the movement to be classified as forwards or backwards, to the left or to the right, upwards or downwards, which is very useful in medical interpretations (Figs. 6, 13). The drawback of this method is that the results are highly dependent on the accuracy of landmark identification, especially of those set of landmarks which are used to built the coordinate system. Much more error prone, but more difficult in medical interpretations, are registrations which use the geometrical constrains. The optimal registration, called Procrustes superimposition [5] searches the transformation composed of a rotation, translation and sometimes scaling in order to minimize the distances between corresponding landmarks. This approach mentioned in Sect. 3.4 is used to find mean shape. Other methods of registration use the information enclosed in the graphical representation of the image. The ICP matching uses the coordinates of points, creating the triangle mesh, in order to find such a transformation, which minimizes the distances between the meshes representing different images. Depending on the constrains, determining which set of points is taken into account, the registration can be global or regional. Different choices of matching points results in different registrations. Therefore this registration must be interpreted with care. The regional registration can be very useful in cases, where the imaged subject changes position. Registering CT images with laser scans it must be remembered that during CT examination the patient is lying while during 3D photo

he is sitting. Therefore the soft tissues of the face are acquired in different positions. In order to register those two images properly the areas, which show minimal changes, should be chosen for the registration procedures. The mentioned method of registration were implemented in our laboratory [8, 9].

6.2 Non-invasive Techniques

The special case of the registration appears when two images being registered do not have common areas. There is a need of adding additional surfaces which can be scanned with both images. This kind of registration turn to be very useful in combining the scans of faces and digital dental models [29]. The additional surfaces come from the reference object – an equivalent of the traditional facial arch, which can be scanned both with faces and dental models (Fig. 14). The resulting image is a new quality in orthodontic imaging. It contains the information, so far available only in CT, namely the relationship between facial features, teeth and occlusal plane. Requiring no radiation, the imaging can be repeated freely to continuously monitor changes. [29] shows the application of this method in orthodontic diagnosis.

7 Patient's Specific Model

Depending on the available image information it is possible to built the individual patient's model. We assume this model as the sets of different surfaces representing the maxillofacial skeleton, teeth and skin of a patient together with localized, skeletal and skin landmarks.

This model can be used for different purposes – to anticipate the future or past patient's appearance and for planning surgical corrections. One of the major advantage of the model is the fact that full information of the model in a particular
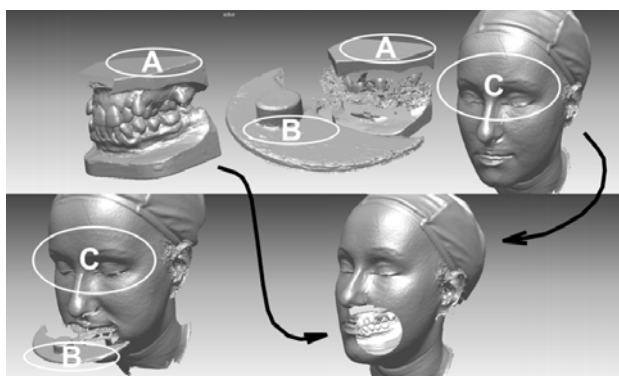
Fig. 14 Scan of a cast in occlusion, first without, then with a facial arch, a scan of the patient's face first without, then with the same arch, and the final 3D image of the face with teeth inside and in register. White letters indicate matching areas

Fig. 15 On the left: layers of the model from 3D reconstruction together with landmarks, on the right: the same information from less invasive imagery sources
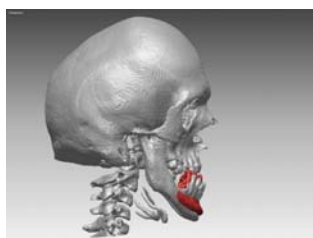
Fig. 16 Result of the geometric optimization – rejected by medical specialist

Fig. 17 The 3D printer and the solid model of the mandible

time moment can be used to interpolate the information in the next time moment in which only partial information is available. When data from one pre treatment CT are available, this information is sufficient to interpolate the model which can be acquired from non invasive (scanning) or less invasive (pair of cephalograms) sources of images. Such a model was the subject of our research project 3 T11F 004 27 supported by the Ministry of Science and Higher Education of Polish Government.

8   Support for Surgical Planning

The obtained data can serve as the information for the planning of the surgical interventions. The aim of such intervention is to correct spatial, facial, maxillary deformities as well as restore function of the masticatory system. At first careful measurements must be performed. Planning a treatment needs careful quantitative evaluation of facial and skeletal deformities. The plan of the intervention usually is very individual for each patient. Here the possibilities of computers can be of great value to find the optimal vector of translation of tissues, positioning of implants, trajectory of the distraction screw. Unfortunately, the geometrical information is not sufficient. The medical constrains must be also taken into consideration. Figure 16 shows the result of geometric optimization, rejected by a specialist, who wanted to avoid mandibular nerve injury during the osteotomy of the mandible. The planning of the interventions is still our aim, and the only way is interdisciplinary cooperation.

Much easier is the implementation of a system giving the possibility to perform virtual interventions depending on the expert. Such system can use the advanteges of new technologies such 3D projectors and haptic devices like CyberGlove. Additionally, there exists the possibility to perform the intervention on stereolitographic solid models, obtained from 3D printer.

9   Conclusions

The aim of the article was not to discus in details any of the mentioned ideas, but to show the immensity of the research problems and the applications which emerged from the computerization of traditional methods of analysis. It is rather paying homage to late Prof. A. Mrózek who was the inspirer of all those coverage and the statement that his ideas and work is being continued now in order to help children with maxillofacial malformations.

References

1. Bargieł, J.: Trójwymiarowa analiza metodą tensorów zniekształsceń twarzowej części czaszki u pacjentów z zesztywnieniem stawów skroniowo-żuchwowych. Ph.D. thesis, Silesian Medical University, Zabrze, Poland (2003)
2. Bookstein, F.: On cephalometrics of skeletal change. American Journal of Orthodontics 82, 177–198 (1982)
3. Dean, D., Hans, M., Boostein, F., Subramanyan, K.: Three-dimensional Bolton-Brush growth study landmark data: ontogeny and sexual dimorphism of the Bolton standards cohort. Cleft Palate Craniofac Journal 37, 145–156 (2000)
4. Dietrich, P.: Ortodoncja tom 1 - rozwój struktur ustno-twarzowych i diagnostyka. Wydawnictwo Medyczne Urban and Partner, Wrocław (2004)
5. Dryden, L., Mardia, K.: Statistical Shape Analysis. John Wiley & Sons, London (1998)
6. Grayson, B., Cutting, C., Bookstein, F., Kim, H., McCarthy, J.: The three-dimensional cephalogram: theory, technique, and clinical application. American Journal of Orthodontics and Dentofacial Orthopedics 94, 327–337 (1988)

7. Hajnal, J., Hill, D., Hawkes, D.: Medical image registration. CRC Press, Boca Raton (2001)

8. Kadłubek, S.: Opracowanie systemu rejestracji obrazów wielomodalnych. Master's thesis, Silesian University of Technology, Gliwice, Poland (2006)

9. Kadłubek, S., Tomaka, A., Winiarczyk, R.: Computer system for the registration and fusion of multimodal images for the purposes of orthodontics. Journal of Medical Informatics & Technologies 10, 65–71 (2006)

10. Karaczyn, M., Pucher, K.: The MAVEC FPGA-based image acquisition and processing board. In: Proceedings of the International Conference on Programmable Devices and Systems, Gliwice, Poland (1995)

11. Lorenson, W., Cline, H.: Marching cubes: A high resolution 3d surface construction algorithm. ACM SIGGRAPH Computer Graphics 21, 163–169 (1987)

12. Łabiszewska Jaruzelska, F.: Ortopedia szczękowa, zasady i praktyka. Portal Wydawnictwa Lekarskiego, Warsaw (1995) Handbook for stomatology students

13. Łabiszewska Jaruzelska, F., Grzesiewska, K., Pisulska-Otremba, A.: Materiały do ćwiczeń z zakresu teleradiografii

14. Michalik, A.: Rozwój twarzowej części czaszki u dzieci z całkowitym obustronnym rozszczepem podniebienia pierwotnego i wtórnego na podstawie komputerowej oceny telertg głowy w projekcji L-L i A-P. Ph.D. thesis, Silesian Medical University, Zabrze, Poland (1994)

15. Otremba, J.: Komputerowa analiza teleradiogramów bocznych głowy dziecka metodą tensorów. Master's thesis, Silesian University of Technology, Gliwice, Poland (1992)

16. Pawełczyk, M., Luchowski, L., Pisulska-Otremba, A.: Porównanie metody półautomatycznej i automatycznej komputerowej analizy teleradiogramów w zastosowaniu do pomiarów nosogardła u dzieci z rozszczepami podniebienia. In: Materiały Ogólnopolskiej Konferencji–Współczesna Diagnostyka Radiologiczna Części Twarzowej Czaszki, Opole, Poland (1991)

17. Pisulska-Otremba, A., Michalik, A., Tarnawski, M., Bargieł, J.: Analiza morfologii twarzowej części czaszki u pacjentów z zesztywnieniem stawów skroniowo-żuchwowych metodą tensorów. In: Materiały Konferencji Technki Informatyczne w Medycynie, pp. BP93–B98 (1998)

18. Skabek, K., Tomaka, A.: Automatic merging of 3d attribute meshes. In: Kurzyński, M., et al. (eds.) Computer Recognition Systems 2. Advances in Soft Computing, vol. 45, pp. 196–202 (2007)

19. Sornek, A.: Opracowanie systemu pomiarowego operującego na skanach głowy dla potrzeb diagnostyki medycznej. Master's thesis, Silesian University of technology, Gliwice, Poland (2006)

20. Sornek, A., Tomaka, A.: Computer system for the analysis of facial features based on 3d surface scans. In: Proceedings of the 11th Conference on Medical Informatics & Technology, pp. 377–382 (2006)

21. Tarnawski, M.: Metoda tensorów w ocenie zmian morfologii twarzowej części czaszki i planowaniu leczenia połowicznego niedorozwoju twarzy. Ph.D. thesis, Silesian Medical University, Zabrze, Poland (2002)

22. Tomaka, A.: Wybrane aspekty implementacji analizy wzrostu twarzo-czaszki metodą tensorów. In: Materiały Konferencji Techniki Informatyczne w Medycynie, pp. BP21–B32 (1998)

23. Tomaka, A.: Detekcja i opis deformacji struktur na podstawie sekwencji obrazów cyfrowych. Ph.D. thesis, Silesian University of Technology, Gliwice, Poland (2001)

24. Tomaka, A., Liśniewska-Machorowska, B.: The application of 3d surfaces scanning in the facial features analysis. Journal of Medical Informatics & Technologies 9, 233–240 (2005)

25. Tomaka, A., Luchowski, L.: Computer-aided longitudinal description of deformation and its application to craniofacial growth in humans. Archiwum Informatyki Teoretycznej i Stosowanej 4, 323–342 (2003)
26. Tomaka, A., Luchowski, L.: Stereovision techniques for the three-dimensional cephalo-gram. In: Proceedings of E-helth in Common Europe. Cracow, Poland (2003)
27. Tomaka, A., Mrózek, A.: Komputerowe metody pomiaru deformacji twarzo-czaszki na podstawie sekwencji teleradiogramów. In: Materiały II Krajowej Konferencji: Metody i systemy komputerowe w badaniach naukowych i projektowaniu inżynierskim, pp. 109–114. Cracow, Poland (1999)
28. Tomaka, A., Skabek, K.: Automatic registration and merging of range images of human head. In: Proceedings of the 11th Conference on Medical Informatics & Technology, pp. 75–81 (2006)
29. Tomaka, A., Tarnawski, M., Luchowski, L., Liśniewska-Machorowska, B.: Digital den-tal models and 3d patient photographs registration for orthodontic documentation and diagnostic purposes. In: Kurzyński, M., et al. (eds.) Computer Recognition Systems 2. Advances in Soft Computing, vol. 45, pp. 645–652 (2007)
30. Tomaka, A., Trzeszkowska-Rotkegel, S., Tarnawski, M.: Correction of error in 3d recon-struction induced by CT gantry tilt. Journal of Medical Informatics & Technologies 12, 169–175 (2008)

From Museum Exhibits to 3D Models

Agnieszka Tomaka, Leszek Luchowski, and Krzysztof Skabek

Abstract. The article presents the use of 3D scanners and printers for the digitization of 3D objects of cultural heritage. The work describes the full processing chain from the scanning of an exhibit to the creation of a tangible copy using a 3D printer. Problems caused by imperfect data sets are discussed and solutions proposed.

Keywords: 3D scan, digitization of objects, models of exhibits.

1 Introduction

A *virtual museum* can be defined as a collection of digital models of exhibits. The notion of exhibits, in this context, is extremely comprehensive. However, from a technical point of view, it is the strictly physical characteristics of each item that determines the way the models are created. The 3D acquisition problem was also discussed in [1].

The Virtual Museum project aims to create tools for the digitization of 3D objects of cultural heritage. The project tools and methods are developed at the 3D Exploration Laboratory (*LEP3D*). In this article we focus on museum exhibits presented in Fig. 1b–c.

2 The Devices

Optical 3D scanners (Fig. 1a) and so-called *3D printers* (Fig. 1b) are two classes of instruments that ensure a direct transition between real physical shapes and their 3D mesh models. They are obviously complementary to each other.

Agnieszka Tomaka · Leszek Luchowski · Krzysztof Skabek
Institute of Theoretical and Applied Informatics,
Bałtycka 5, 44-100 Gliwice, Poland
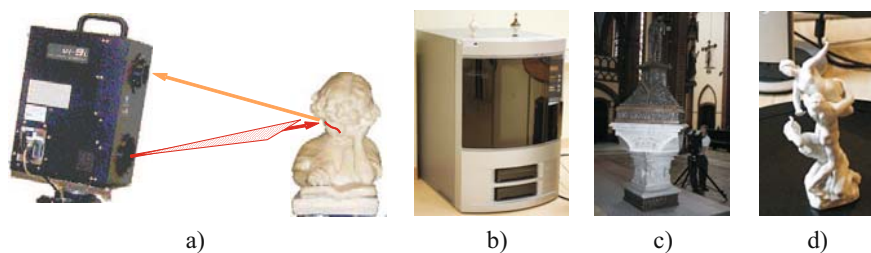e-mail:{ines,leszek.luchowski,kskabek}@iitis.pl

a)   b)   c)   d)

Fig. 1 (a) Konica-Minolta VI-9i – principle of a laser triangulation system. **(b)** Dimension Elite – 3D printer. **(c)** Baptismal font from Cathedral in Gliwice. **(d)** Figure of *The Rape of the Sabines* from Museum in Gliwice

Optical 3D scanners analyze the shape of an object and create its detailed digital model. The initial representation is just a set of points (x, y, z) densely and more or less uniformly distributed over the surface of the object, where the coordinates of each point are known. The set may be an unstructured *point cloud* or it may be organized into a polygonal *mesh* where neighboring points are connected by edges.

While the scanned image as a whole can be quite accurate, individual points can be missing. Moreover, if the points are organized into a mesh, its topology does not always reflect that of the object, and in most cases it does not define any solid at all, i.e., the polygons do not form a closed surface.

3D printers, on the other hand, were originally intended as *rapid prototyping devices*, i.e., tools used to quickly produce a physical rendition of a shape designed under CAD. They expect to receive a model of a well-defined 3D solid shape – which, in the case of a mesh format, means a closed manifold with no open ends, no loose faces, no self-intersections, and no zero-thickness parts.

There are various types of 3D scanners, using various techniques to acquire the coordinates of object points. In the present work we concentrate on using a triangulating 3D laser scanner (Fig. 1a).

A triangulating laser scanner sends a beam of light from one *projection center*, and uses a camera positioned at a different point to observe the resulting spot or streak on the object. The 3D point is identified at the intersection of the (known) trajectory of the laser beam with the visual ray connecting the spot to the camera center. This process yields very accurate short-range measurements, but can only be applied at a limited range; at greater distances, the two lines become nearly parallel and their intersection cannot be computed reliably. Another setback is that a point can only be scanned if it is in line-of-sight view of both the laser and the camera. For this reason, a triangulating scanner cannot explore narrow deep recesses in objects.

Our laboratory now has several different 3D scanners. The choice of device with which to scan each object depends on its size and on the environment in which the scanning will be done. The Konica Minolta Vi-9i is the scanner we now have most experience with. It is of the triangulating laser variety.

3 Scanning Conditions

The Minolta VI-9i laser scanner has a number of options which have to be selected before scanning [3]. The user has a choice of three lenses (known as wide, middle, and tele), and several software settings. The object may also require some treatment before it can be reliably scanned.

3.1 Choice of Lens

The scanner lenses differ in the angular width of their field of view, and in this respect choosing a lens is similar to setting the zoom in an ordinary camera. However, the lenses also have different ranges of distances at which they can focus, and they yield a different level of detail.

The lens must therefore be chosen in consideration of two factors: the size of the object and the required precision. The wide lens allows larger objects to be scanned in fewer takes, which saves scanning time and, more importantly, reduces the workload of data integration. It is, however, less precise and yields 3D images with a poorer resolution. The tele lens should be used whenever the object is small enough to fit into its field of view, and for larger objects if precision and high resolution are important. A compromise is often necessary between speed, scope and convenience on one side, and image quality on the other.

A combination of scans using different lenses can also be used. In this case the wide lens provides a general outline of the object. The more precise but fragmented imagery from the tele lens is then brought into register using this outline as a reference.

3.2 Parameter Settings

The Konica Minolta VI-9i has a very sensitive focus system. In fact, the manual focus is really a powerful tool to acquire many depth planes of complicated objects. The focus distance can also be set automatically by the AF (AutoFocus) function.

The next important parameter is laser power. This can also can be set automatically, but there are cases in which it is useful to adjust it manually. The laser power influences the width of the stripe of light seen by the camera of the scanner, which affects the final accuracy of the scanned surface. As not all colors of surfaces have the same reflectance for red light, laser power has to be increased when scanning black or blue surfaces.

The influence of object color on scan precision, and the choice of laser power for the color or colors of a given object, were discussed in detail in an M.Sc. dissertation [7].

The next choice concerns the mode of scanning. The standard mode enables an object to be scanned at a distance from 600 to 1000 mm. In extended mode, the range is from 500 to 2500 mm, but with a significant loss of accuracy at boundaries.

The next parameter to be set is the number of scans, i.e., the number of consecutive passes with the laser beam at different power, applied in one take in order to

reduce data loss. The scanner also has the additional possibility to reduce the data (1/1, 1/4, 1/9, 1/16, and adaptative mode), and to filter the data – the noise filter and high quality filter can be applied, to automatically fill holes, and also to remove boundary points and polygons that are scanned at a grazing angle less than 5, 10, 15, or 20 degrees (measured between the surface and the line of view).

Applying all those filtering operations increases the accuracy of the scan, so that the less accurate data is filtered out, but it also reduces the number of polygons. Even though the object is usually scanned from many viewpoints, some parts of the surface may not be represented at all. The user can choose the level of data filtration, i.e., removing boundaries and sharply sloping polygons. In our practice we very often decide to apply only a slight filtering so as to cover a greater surface.

3.3   Preparing an Object Whose Optical or Shape Properties Are Difficult

For the beam to be visible on the surface of the object, the surface should display isotropic reflectance, i.e., have approximately Lambertian optical properties. This condition is not met by specular (mirror-like) surfaces, which reflect the laser beam in one direction, and transparent objects like glass where the beam passes through matter. The most popular solution to these problems is to cover the surface with a powder, e.g., talcum. It should, however, be kept in mind that a layer of powder is never infinitely thin or perfectly uniform, so it will have some adverse effect on precision. The powder can also fill cracks and narrow recesses in the object, thus eliminating some details from the scan and possibly making the physical object difficult to clean.

Another issue is ensuring that there are enough landmarks (surface features) in every scan. When individual scans are put together to create a full model, uniform surfaces (especially flat, cylindrical and spherical) may lead to ambiguous registration with one, two, or three degrees of freedom, as the individual scans can be shifted and rotated with respect to each other with no clue to define one correct position. If it is not possible to change the field of view to include some less uniform parts of the object, markers (artificial landmarks) may have to be added. The markers are then removed from the scan, and the resulting holes filled by copying neighboring parts of the uniform surface.

4   Scanning an Object

Most exhibits are 3D solids whose entire shape (all-round surface) is of interest, and which therefore cannot be scanned from one point of view. Exceptions include works sculpted in bas-relief or intaglio, which can, in principle, be sufficiently represented by one depth map. As a general rule, however, an exhibit has to be scanned many times, from many juditiously selected points of view, and the partial images have to be brought into register and merged using 3D editing software.

This process, described below, is complicated and labor-intensive. To keep the workload within reasonable bounds, viewpoints must be chosen in such a way as to cover the whole surface of the object, with no gaps, in the smallest possible number of takes.

A certain viewpoint may yield less information than expected. The cause of the problem then has to be identified and the scan repeated with a slightly modified scanner position or different scanning parameters. Some of the possible causes are: bad focus caused by the central beam passing through a gap in the object, wrong laser power for a particular object reflectivity, excessive angle between the central beam and the surface normal (the surface should be approximately perpendicular to the beam), scanned area too narrow for reliable registration.

The position of the scanner relative to the object can be changed by moving either the scanner or the object. The most convenient solution is to put the object on a turntable, which is controlled by computer and so a coarse preliminary registration can be performed automatically.

Some objects cannot be moved because they are too fragile, too heavy, or permanently attached to the ground. Others (such as textiles) might change shape when moved around, leading to unsolvable differences between successive scans. In such cases, the scanner has to be moved around. Unless its motion can somehow be digitally tracked, the successive viewpoints are not precisely controlled or even known, and both the positioning and subsequent scan registration then has to rely heavily on human judgment.

5   Registration

Scanner data from a single observation point are recorded in the local coordinate system associated with the center position of the scanner (Fig. 2a). *Registration* means the transfer of the scans taken from different viewpoints into a common coordinate system associated with the object. In this work we used the registration procedures implemented in the RapidForm environment. These methods are mostly based on techniques of minimizing the distance between the two registered surfaces [2]. Depending on the type of registration we minimize the point-to-point distance of the whole surfaces (*global registration*), or only selected parts of the surfaces (*regional registration*). There is another method (*manual registration*) where a user performs the registration manually by pointing to at least three corresponding points on each surface and the procedure optimizes slightly the resultant position. This procedure is really useful when the scans being registered differ greatly, but still there must be an overlapping area between them.

5.1   Registration of Large Objects

In practice, the main disadvantage that appears during registration is the large volume of data. This makes the available registration procedures inadequate, especially for large objects and for ones that were scanned at a higher resolution.

Such is the case of the baptismal font (Fig. 2). The registration of this object required some special treatment of the data: simplifying and reducing the volume of data from individual scans and also creating groups of scans. For registration within a group to be possible, the individual scans must have a common part. Registration is performed for scans in each group (Fig. 2b) and after that scans are merged in groups and then overall registration is performed for the merged groups (Fig. 2c). In this way the redundant data in scans is removed and thus the procedure requires less memory.

This registration method has certain disadvantages. The simplification of scans makes it relatively easy to obtain the final structure, but it loses some information and decreases the accuracy of the final representation.

In some cases it can also appear that the initial registration in groups leads to wrong results. For example, at a certain stage of registration of the merged groups it appeared that the resulting structure has incorrectly matched parts of surfaces (e.g., two heads side by side in Fig. 3) and thus correct registration was impossible.

Our solution to these problems is to use information about the location of each scan. This information is only available for the source scans. Then the center of the scanner is annotated as $(0,0,0)$. Each registration procedure looses the position of the scanner and, to prevent this, we use the RapidForm tool that enables us to assign an *orientation marker* to each scan. Such assignment makes it possible to automatically change the position of the marker during transformations (e.g., registration) of the corresponding scan. As the orientation marker we mean three vectors denoting the position of the axis system for coordinates associated with the scanner. Knowing the location of the scanner we can easily place the source scans into already registered group of scans. This property makes it possible to propagate the information to the registration in groups, as well as in the case of registration of merged groups
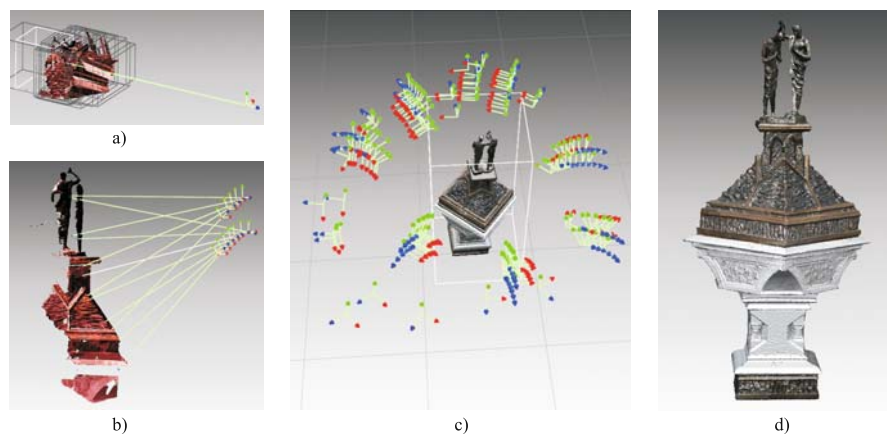

Fig. 2 Registration of large objects: **(a)** scans in the local coordinate system, **(b)** one group of scans registered, **(c)** the whole object registered, **(d)** the merged and textured object

Fig. 3 Wrong registration of merged scan groups

it is possible to reload the source data using the corresponding markers in order to reduce matching errors.

5.2 Registration Using the Rotary Stage

If only it is possible, the scanning process is supported by the calibrated rotary stage. The calibration makes it possible to identify the axis of the rotation table and the known rotation angle allows the subsequent scan to be preliminarily registered to the first position (Fig. 4a–b). In this way, a sequence of scans taken with the table is stored in the coordinate system associated with the table. As a single sequence is usually not sufficient to cover the whole object, it is repeated for another position of the object on the table. Registration between the sequences of scans has to be performed manually.
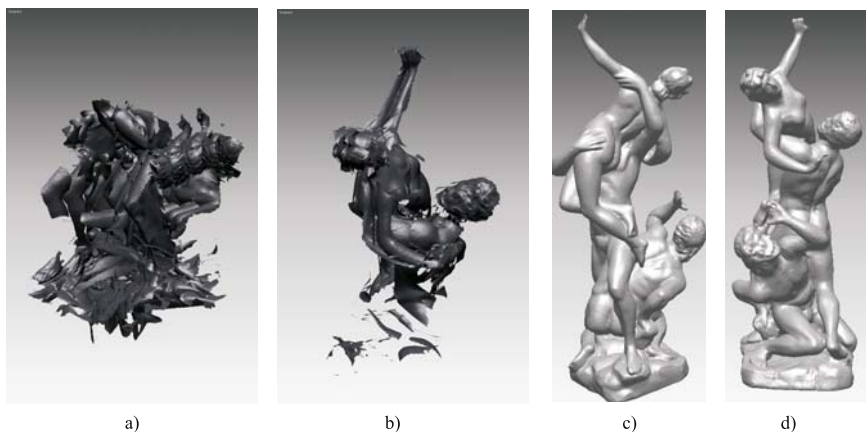

a)   b)   c)   d)

Fig. 4 (a) Scans from uncalibrated rotary table. **(b)** Scans from calibrated rotary table. **(c)**–**(d)** The merged model of Sabines

6 Merging

Merging is a process that combines many registered scans into one surface model. The RapidForm environment offers two techniques of merging: *surface* and *volumetric*. Surface merging procedure reconstructs the model from partial surfaces joining them in the smoothest transition area (if such can be found). Volumetric merging procedure takes into account all the surfaces and the final result is an average of all partial surfaces. There is a parameter that limits the distance between the surfaces for a given direction and this is a way to do the averaging with a lesser or greater tolerance.

The quality of merging directly depends on the reliability of the processing prior to registration. The merging may be also wrong if the surfaces in some parts that should be common do not overlap because of incorrect registration or because of artifacts found during the scanning process (large slope of the surface, flashes as a result of a reflecting surface). The local disruption of surface continuity as well as intersecting surfaces or holes in the surface may occur in such cases.

7 Printing a 3D Model

There is a number of technologies (powder-based, stereophotolitographic and others) that allow a digital 3D model to be *printed* as a physical solid body. The Dimension Elite printer at LEP3D uses a moving head with twin heated nozzles to deposit molten plastic or *model material* on a tray.

The head moves in a horizontal X, Y plane as in an ordinary 2D plotter, forming one layer of the model at a time. When a layer is finished, the tray is moved downwards by a small increment (of the order of 100 microns), and the head proceeds to form the next layer.

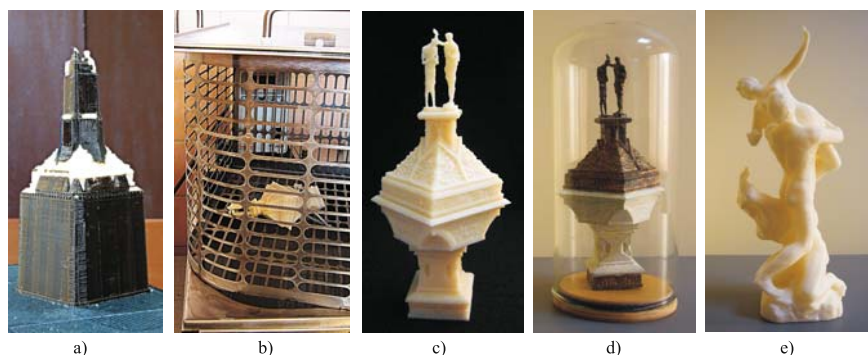a)   b)   c)   d)   e)

Fig. 5 Printing a 3D model. **(a)** Model covered with supporting material. **(b)** Dissolving the supporting material. **(c)** The printed model of baptismal font. **(d)** Painted model. **(e)** The printed model of Sabines

Table 1 Simulation results

Figure	Layer res. [mm]	Model interior	Support fill	Model mat. [cm^3]	Support mat. [cm^3]	Time [hh:mm]
Sabines	0.1778	Solid	Sparse	204.26	100.28	50:44
Sabines	0.1778	Sparse	Sparse	92.18	100.28	41:35
Baptismal font	0.2540	Solid	Sparse	109.82	33.24	10:52
Baptismal font	0.1778	Solid	Sparse	110.00	34.05	19:11
Baptismal font	0.2540	Sparse	Sparse	39.19	36.85	10:40
Baptismal font	0.1778	Sparse	Sparse	33.05	36.22	16:17

It occurs quite often that some part of a layer extends beyond the preceding layer (e.g., when the model represents a person and the head is normally broader than the neck). In such cases the protruding part would have to be formed by the head depositing material on top of empty space (or air). This problem is solved by the second of the twin nozzles, which is fed *support material* and builds a kind of lattice to support the protruding parts of the model. After printing, much of the support material can be removed mechanically, and the rest is dissolved in a bath of hot lye.

Comparing the simulation results shown in Table 1 for the ready-to-print models it can be concluded that layer thickness only slightly affects the amount of material that must be used to build the model and significantly increases the printing time. As the models printed using the thinner layer (0.1778) are much more accurately reproduced and time is not a critical parameter, we often choose printing with the thinner layer.

Parameters that significantly affects the amount of material to print are the size of the object and the type of filling. When selecting a solid fill the printer consumes approximately twice as much material as in the case of sparse filling. The saving is obvious, although in some cases, for example, when we model large flat areas the usage of sparse fill may result in distortion of the model during the cooling process.

References

1. Bernardini, F., Rushmeier, H.: The 3D model acquisition pipeline. Computer Graphics Forum 21(2), 149–172 (2002)
2. Gruen, A., Akca, D.: Least squares 3D surface matching. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 34 (2003)
3. Konica-Minolta Sensing Incorporated: Non-contact 3D digitiser Vivid 9i. Instruction Manual Hardware (2004)
4. Skabek, K., Tomaka, A.: Automatic registration and merging of 3D surface scans of human head. In: Proceedings of Medical Informatics & Technology Conference, pp. 75–81 (2006)

5. Skabek, K., Tomaka, A.: Automatic mering of 3D atribute meshes. In: Kurzydlowski, M., et al. (eds.) Computer Recognition Systems 2. Advances in Soft Computing, vol. 45, pp. 645–652. Springer, Heidelberg (2007)
6. Skabek, K., Tworzydło, A., Luchowski, L., Winiarczyk, R.: Bringing into register incomplete range images of cultural artifacts. Machine Graphics and Vision 15(3-4) (2006)
7. Spytkowska, A.: The exploitation of Konica Minolta Vivid 9i scanner for creating a virtual copy of small museum objects. Master's thesis, AGH University of Science and Technology (2008)

Part IX
Advances in Algorithmics

A Method for Automatic Standardization of Text Attributes without Reference Data Sets

Łukasz Ciszak

Abstract. Data cleaning is an important step in information systems as low quality of data may seriously impact business processes that utilize the given system. Textual attributes are most prone to errors, especially during the data input stage. In this article, we propose a novel approach to automatic correction values of text attributes. the method combines approaches based on textual similarity with those using data distribution features. Contrary to all the methods in the area, our approach does not require third-party reference data. Experiments performed on real-world address data prove that the method may effectively clean the data with high accuracy.

Keywords: data quality, data cleaning, attribute standardization.

1 Introduction

Data cleaning is the process of improving the quality of data by changing the values assumed to be incorrect. It is an important step in the data mining applications, as low quality of data leads to low quality of knowledge inferred from it.

In this article, we concentrate on improving the quality of data through automatic correction of text attribute values. We propose a method that combines the techniques of textual similarity with the approach utilizing the features of attribute value distribution. The method does not need external reference data as it infers it directly from the examined data set.

The structure of this paper is as follows: the first section presents data quality issues and related research in the field of attribute standardization. the second section presents the method of automatic discovery of the reference data and data cleaning

Łukasz Ciszak
Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: L.Ciszak@ii.pw.edu.pl

rules based on the properties of personal and address data in contemporary enterprise databases. the third section presents the results of the experiments on real and artificial data sets performed using the method. It also contains propositions of the algorithm enhancements and a discussion of results.

1.1 Data Quality Issues

Low quality of data may have great impact on the quality of knowledge inferred from it during the data mining process. the most error-prone are the text attributes. Errors in text attributes include syntactical errors such as mistypings(pressing incorrect keys, multiple key pressing, character omission/exchange, and so forth), misspellings (orthographic errors), and vocabulary incompetence errors. Another class of errors are semantic errors where a value may be correct from the syntactic point of view, but incorrect in terms of a given record attribute. Finally, there are random, irrelevant values, for example *first_name*='xxe3', are described as noise.

Attribute standardization, one of the areas of data cleaning, is the process of discovering incorrect or non-standard values of a given parameter and mapping them into the values of the reference data set.

1.2 Related Work

Attribute standardization methods in the data warehouse area [9, 11] involve manual definition of cleaning rules for handling incorrect values. the rules are based on the reference data and the results of the data analysis step. Hérnandez [8] proposes a data cleaning/duplicate detection approach based on textual similarity of values to the reference data set elements. Bilenko and Mooney [2] also use textual similarity approach extended with machine learning methods which are used to determine the accepted distance threshold between values. Lee, Ling, et al. [10] propose a method that combines textual similarity approach extended with user-defined rules. Dardzinska-Glebocka [5] proposes a rough set approach to imputation of missing values. This approach may also be used for attribute standardization, as incorrect values that are not included in the reference data may be treated as missing.

The area related to attribute correction is the area of spell-checking. Damerau [4] proposed a method that involves the usage of a lexicon. In this approach, each word in the document is checked against the lexicon. the lexicon approach was extended by Deorowicz and Ciura [6]. In their method, they represent the lexicon as a finite automaton, which leverages search time efficiency. Churches [3] extends the lexicon method with the probability scores of textual errors.

All the aforementioned approaches require the existence of reference data or lexicons of correct words.

2   The Proposed Approach

In this section, we present our approach to automatic correction of text attributes. Our main contribution and the main difference between our approach and the methods described in the previous section is that no reference data is needed as it is inferred directly from the data set. The discovered reference data are then used to create a set of data cleaning/substitution rules. the substitution rules define the mapping between an incorrect value and a correct value describing the same object from the real world.

The proposed method is useful in situations where the reference data set does not exist or is not up to date. Using the method one can confront the existing reference data with the data generated in the cleaning process hence obtaining the most up-to-date set of reference data.

2.1   Method Description

Let $S = \{s_1, s_2, \ldots, s_n\}$ denote the set of values of a given attribute. Each element s_n of the set S is described by a pair of values (v_n, o_n), where v_n is the text string representing the value and o_n is the number of occurrences of a given value within the data set. Let R denote the reference data set. We assume that $R \subset S$.

Definition 1. *Distance measure is a function $M : S \times S \rightarrow \mathbb{R}^2$ such that:*

$$M(s_m, s_n) = (d_{mn}, r_{mn}) \,,$$

where:

$$d_{mn} = D_w(v_m, v_n), \, d_{mn} \in [0, 1] \,,$$

$$r_{mn} = \max(\frac{o_m}{o_n}, \frac{o_n}{o_m}), \, r_{mn} \in [1, \infty)$$

D_w is the Jaro-Winkler measure of similarity between two text strings [13] based on the number of common characters within a limited distance. It varies from 0 to 1 and is equal 0 for two completely different strings, whereas for two identical strings it is equal 1. For instance, $D_w(\text{'Warsaw'}, \text{'Warswa'}) = 0.97$.

Definition 2. *Similarity relation '$\backsim$' over the set S is a relation defined in that way that:*

$$s_m \backsim s_n \Leftrightarrow (d_{mn} \geq \varepsilon) \wedge (r_{mn} \geq \alpha) \wedge (o_m < o_n),$$

where ε is a textual similarity threshold parameter and α is an occurrence ratio threshold parameter.

The similarity relation over the set S may be represented as a directed weighted graph [12]. The elements of the set are represented as the nodes of the graph, whereas the arcs of the graph represent the similarity relation. the weight of the arc between elements s_m and s_n is equal $M(s_m, s_n)$.

Definition 3. *The indegree of a node, denoted by* $\deg^-(s)$*, is the number of incoming arcs.*

Definition 4. *The outdegree of a node, denoted by* $\deg^+(s)$*, is the number of outgoing arcs.*

Definition 5. *A sink is a node for which* $\deg^-(s) > 0$ *and* $\deg^+(s) = 0$*.*

Definition 6. *An isolated node is a node for which* $\deg^+(s) = 0 \wedge \deg^-(s) = 0$*.*

We have observed that all the examined datasets displayed the features of scale-free networks [1] as there were hub nodes – sink nodes with significantly high indegree.

We assume that the reference dataset consists of the values stored in hub nodes and isolated nodes with a large number of occurrences. Remaining nodes represent incorrect (misspelled, mistyped, etc.) versions of values from the reference data set. Isolated nodes with low number of occurrences of value are assumed to be noise.

2.2   Algorithm Definition

The algorithm discovers the reference data set by analyzing the structure of the graph. It also proposes a set of correction/substitution rules that map the incorrect values into the reference data set elements.

The algorithm has the following parameters:

1. ε – textual similarity threshold, that is, the minimum similarity between two strings allowing to treat them as identical;
2. α – occurrence threshold, that is, the minimum ratio of occurrences of two attributes allowing to determine the direction of the similarity relation.

The algorithm consists of the following steps:

1. Distance matrix calculation: in this steps we calculate the similarity matrix $\mathbf{A} = (a_{ij})$ where $a_{ij} = M(s_i, s_j)$ using the metric M.
2. Graph construction: using the similarity matrix we construct the graph satisfying algorithm parameters ε and α.
3. Graph reduction: for each node we remove all the arcs except the one having the maximum weight (the measure between the nodes it connects is maximum). Also, in this step we remove the out arcs for nodes having the indegree greater or equal than the defined limit for sink nodes.
4. Graph merge: for each sink node we traverse the graph and connect indirect neighbors(nodes connected to direct predecessors of the considered node) directly to the considered node
5. Incorrect value cleaning: in this step, the mapping cleaning rules are generated. All the values in the non-sink nodes are mapped into the values in the sink nodes they are connected to.

The computational complexity of the described method is $O(n^2)$, as we need to calculate the distance between each pair of the attributes from the input set. To

minimize the number of comparisons and the complexity, the method may utilize the 'sliding window' technique as described in [8]. In this technique, the values are sorted alphabetically and each attribute is compared only with k other attributes, where k is the size of the window. This reduces the complexity to $O(n)$.

3 Experimental Results

We have tested the proposed approach experimentally. the aim of the experiments was to test if the method can effectively discover reference data and use it to generate the correction rules. We have defined the following algorithm effectiveness measures:

1. p_c – the percentage of correctly altered values (correct ratio). This measure describes the capability of the algorithm to map the incorrect values into the reference data set elements;
2. p_f – the percentage of values that should have been altered but were not ('false negatives'). This measure describes the capability of discovering the errors in the data and the reference data set values.

The measures are defined as follows:

$$p_c = \frac{n_c}{n_a} \times 100, \qquad p_f = \frac{n_0}{n_{00}} \times 100, \qquad (1)$$

where n_c is the number of correctly altered values, n_a is the number of altered values, n_0 is the number of values initially identified as incorrect and not altered during the cleaning process, and n_{00} is the number of values initially identified as incorrect.

The result of each algorithm run for a pair input parameters were the values of the aforementioned measures.

The tests were performed on three data sets. For the first data set we used the names of the fifty largest Americancities as the reference data. The numbers of occurrences were generated according to the Zipf law. For this reference data set we have generated a set of incorrect values in accordance with the error statistics published by Giebultowicz [7]. The percentage of incorrect values reflected that typical for the real world data from and was equal 10%. The second data set was generated in the same way as the first one; the difference was that in this case the reference data set was the set of fifty largest Polish cities.

The third data set is a real data set of Polish cities taken from an Internet survey. The set was inspected manually and all the values identified as incorrect were assigned a correct value in order to enable calculating the aforementioned measures after the cleaning.

For the first two datasets we have obtained very good results – the p_c measure reached 100% with the value of p_f measure equal 6% for the algorithm parameters $\varepsilon = 0.7$ and $\alpha = 4$. This means that the algorithm can effectively correct data when the reference set elements are far from each other in terms of textual similarity.

The results for the real-world data set are contained in Tables 1 and 2.

Table 1 The dependency between the algorithm parameters and the p_c measure

p_c					α					
ε	1	2	3	4	5	6	7	8	9	10
0.0	10	12	11	9	9	10	8	7	7	7
0.1	10	12	11	10	9	10	8	8	7	7
0.2	10	11	12	10	9	10	8	8	7	7
0.3	10	11	12	11	11	11	9	9	9	9
0.4	13	16	17	15	16	16	16	16	16	16
0.5	16	20	20	22	23	24	24	25	25	25
0.6	31	38	43	44	45	49	50	52	53	54
0.7	54	64	67	69	69	73	74	74	75	76
0.8	81	86	87	87	87	87	89	89	89	89
0.9	92	97	97	97	97	97	96	96	97	99
1.0	0	0	0	0	0	0	0	0	0	0

Table 2 The dependency between the algorithm parameters and the p_f measure

p_f					α					
ε	1	2	3	4	5	6	7	8	9	10
0.0	0	0	12	22	26	24	34	34	37	37
0.1	0	0	12	22	26	24	34	34	37	37
0.2	0	0	12	22	26	24	34	34	39	39
0.3	0	4	16	28	28	30	40	40	41	41
0.4	5	5	18	32	34	36	37	39	39	39
0.5	8	8	21	26	27	30	31	32	33	33
0.6	10	11	13	19	21	22	23	24	25	25
0.7	16	16	19	22	25	26	27	27	28	29
0.8	24	25	27	30	32	33	34	34	35	36
0.9	89	90	91	92	92	92	93	93	93	94
1.0	100	100	100	100	100	100	100	100	100	100

The generated data cleaning rules are optimal when the p_c measure is as high as possible and the p_f is at the same time as low as possible; the interpretation is that the algorithm managed to discover large number of incorrect values and successfully mapped them into the elements of the reference data set. For the examined data set the optimal results were obtained for the algorithm parameters $\varepsilon = 0.8$ and $\alpha = 2$. For this set of parameters the measures were equal $p_c = 86\%$ and $p_f = 25\%$.

The ε parameter has the largest impact on the algorithm effectiveness. Figure 1 shows the dependency between the measures p_c and p_f, and the parameter ε for $\alpha = 2$. It shows that with the growth of the ε parameter the percentage of correctly altered attributes p_c is rising; also, the percentage of 'false positives' p_f is rising. Therefore, the larger the ε parameter is, the more 'restrictive' is the
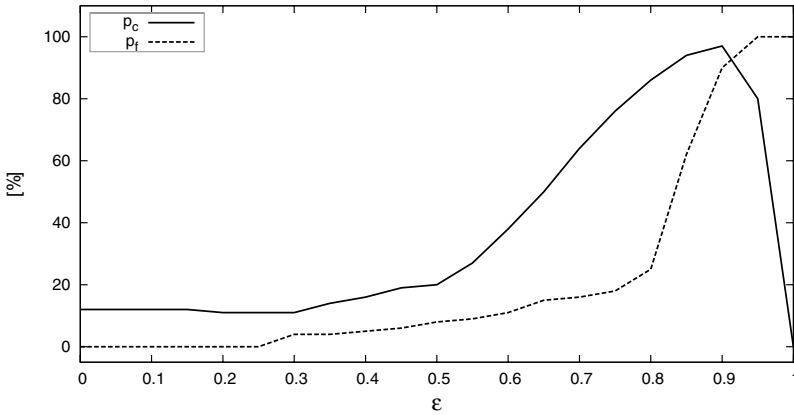
Fig. 1 The dependency between the algorithm parameter ε and the measures p_c and p_f for a fixed value of parameter $\alpha = 2$

algorithm. Also, the number of altered values is decreasing, but the ratio of elements correctly mapped into the values of the reference data set is higher.

The algorithm displays better performance for long strings as short strings would require lower value of the ε parameter to discover the errors. However, low values of this parameter result in large number of incorrectly altered elements. Also, this method applies to large datasets, as small sets could lead to inferring incorrect reference data.

The range of the applications of this method is limited to attributes that can be standardized,that is, for which reference data may exist. Conversely, using this method for cleaning last names could end with a failure. The major drawback of this method is that it may classify as incorrect a value that is correct in context of other attributes of this record, but does not have enough occurrences within the cleaned data set. Therefore, method improvement should involve examining the context of the cleaned value.

4   Conclusion

In this paper, we have presented a method for automatic correction of textual attributes. Contrary to other methods and approaches in this area, the proposed method does not need third-party reference data as it generates it directly from the data. Experimental results show that the method can effectively discover reference data set from the examined data, discover erroneous values and match them with correct values of the reference data set. The method gives best results when the examined data contain a low number reference elements that are distant in terms of the textual similarity. These conditions apply to attributes such as car brands, cities, and so forth.

Further research is planned in order to achieve better results using the method; we plan to modify the approach so that also the attribute context is taken into account when the correction is performed. We also plan to extend the method with a rough-set approach.

References

1. Barabási, A.L., Bonabeau, E.: Scale-free networks. Scientific American 288(5), 50–59 (2003)
2. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining (2003)
3. Church, K.W., Gale, W.A.: Probability scoring for spelling correction. Statistics and Computing 1(2), 93–103 (1991)
4. Damerau, F.J.: A technique for computer detection and correction of spelling errors. Communications of the ACM 7, 171–176 (1964)
5. Dardzinska-Glebocka, A.: Chase method based on dynamic knowledge discovery for predicting values in incomplete information systems. Ph.D. thesis. Polish Academy of Sciences, Warsaw (2004)
6. Deorowicz, S., Ciura, M.G.: Correcting spelling errors by modeling their causes. International Journal of Applied Mathematics and Computer Science 15, 275–285 (2005)
7. Giebultowicz, M.: Polish spelling errors categorization. In: Proceedings of the 1st International Interdisciplinary Technical Conference of Young Scientists (2008)
8. Hernández, M.A., Stolfo, S.J.: Real-world data is dirty: Data cleansing and the merge/purge problem. Data Mining and Knowledge Discovery 2(1), 9–37 (1998)
9. Kimball, R., Caserta, J.: The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. John Wiley & Sons, Chichester (2004)
10. Lee, M.L., Ling, T.W., Low, W.L.: IntelliClean: a knowledge-based intelligent data cleaner. In: Proceedings of the 6th ACM International Conference on Knowledge Discovery and Data Mining, New York, US, pp. 290–294 (2000)
11. Maydanchik, A.: Data Quality Assessment. Technics Publications, LLC (2007)
12. Monge, A.: Adaptive detection of approximately duplicate database records and the database integration approach to information discovery. Ph.D. thesis, University of California, San Diego, US (1997)
13. Winkler, W.E.: The state of record linkage and current research problems. Tech. rep., Statistical Research Division, U.S. Census Bureau (1999)

Internal Conflict-Free Projection Sets

Łukasz Mikulski

Abstract. Projections into free monoids are one of possible ways of describing finite traces. That approach is used to expand them to infinite traces, which are defined as members of a proper subfamily of cartesian product of mentioned free monoids, completed with least upper bounds in the Scott's sense. Elements of the cartesian product are called projection sets and elements of the proper subfamily are called reconstructible projection sets. The proposed description gives useful methods of investigation. Especially, it contains a constructive operation that turns an arbitrary element of cartesian product to the closest trace. Then, we define constructively a proper subclass of projection sets, called internal conflict-free projection sets, and show using structural induction that it is precisely equal to subfamily of reconstructible projection sets. This is the main result of the paper. The proofs are ommited becouse of the page limit.

Keywords: projection sets, free monoids, structural induction, finite traces.

1 Introduction

The idea of traces was firstly introduced in [1] and [7]. The first paper focuses on the combinatorial properties of traces, the second one treats traces as a model for concurrent computations. This practical application caused sudden development of the trace theory. During past three decades, there were plenty papers concerning finite traces. The most comprehensive monographs are [2] and [3].

The classical theory of traces concentrates on the model for finite computations. However, nowadays models for infinite computations are important as well. This need is caused by investigating behaviour of the concurrent systems that, in

Łukasz Mikulski

Faculty of Mathematics and Computer Science, Nicolaus Copernicus University,
Chopina 12/18, 87-100 Toruń, Poland
e-mail: frodo@mat.umk.pl

theory, works without breaks. One of possible solution is extending the trace theory to infinite traces. There were some approaches [4, 5, 6]. They start from dependence graphs of the traces or prefix partial order on the traces. The second approach gives more intuitive and promising definition, but is complex. It was a motivation of study infinite traces and reformulate their definition in the equivalent manner.

An interesting view on traces was introduced in [9]. It is noticed there that a set of specially chosen morphisms to free monoids completely describes traces. It is a starting point of constructing infinite traces in [4] and [8]. The present paper extends the approach of [8] mainly concerning the family of projection sets.

In the second section some basic notions of formal languages and domain theory is given. In the third, the definition of finite traces is recalled. Moreover, another way of thinking about dependence relation is proposed. It focuses on set of pairs of dependent letters. Finally, the definition of infinite traces using projections is given. The following two sections introduce the family of projection sets and its subfamily, called internal conflict-free projection sets. The latter are defined inductively and such definition allows to use structural induction in the proof of the main result – internal conflict free projection sets precisely correspond to reconstructible projection sets. In the sixth section the definition of reconstruction operation is recalled and the comprehensive catenation of traces is proposed.

2 Basic Notions

We will use some basic notions of formal languages theory and domain theory. By Σ we denote an arbitrary finite set, called *alphabet*. Elements of the alphabet are called *letters*. *Words* are arbitrary sequences over the alphabet Σ. The sets of all finite and infinite words will be denoted by Σ^* and Σ^ω, respectively. Their union, which is the set of all words, will be denoted by Σ^∞.

The main operation on words is a *catenation* $\cdot : \Sigma^\infty \times \Sigma^\infty \to \Sigma^\infty$. The catenation of two words is a second argument written after the first, if the first argument is finite, and just the first argument, if it is infinite. The set Σ^∞ equipped with the catenation forms a monoid. Another useful operation is a *projection* $\Pi : \Sigma^\infty \times 2^\Sigma \to \Sigma^\infty$. The projection of a word σ on subalphabet $S \subseteq \Sigma$ is the word received by erasing from σ all letters from $\Sigma \setminus S$. Instead of $\Pi(\sigma, S)$ we write $\Pi_S(\sigma)$.

The *partial order* in a set X is a relation $\leq \subseteq X \times X$, such that $\leq$ is reflexive, antisymmetric and transitive. Pair $(X, \leq)$ is called *partially ordered set*, or shortly *poset*. A nonempty subset of X, in which for arbitrary two elements exists an element greater than each of them, is called *directed set*. If in the poset $(X, \leq)$ every directed subset has least upper bound, then $(X, \leq)$ is said to be *directed-complete*. If, moreover, a poset $(X, \leq)$ possesses a least element, then $(X, \leq)$ is said to be *complete* in Scott's sense.

An *ideal* in a poset $(P, \leq)$ is a directed down-set $I \subseteq P$. It means that I is directed and for every element $a \in I$ all elements lower than a also belong to I. The *ideal completion* $\overline{P}$ of P is a poset of all ideals of P, ordered by inclusion. The ideal completion is always directed-complete.

In Σ^∞ we have a natural partial order, denoted by $\sqsubseteq$, called *prefix partial order*. Word σ is a *prefix* of word τ if and only if there exists a word υ, such that $\tau = \sigma\upsilon$. Poset $(\Sigma^\infty, \sqsubseteq)$ is complete.

3   Traces

Words, described in the first section, are used to model sequential computations and their behaviour. Letters of the alphabet Σ are treated as single, atomic actions. However, this is not enough when we study concurrent systems. A single word represents only one of the possible sequence of actions, the parallel behaviour of such systems are not caught. As a generalisation of previous model, in 1970's, was introduced concurrent words, named *traces* [7].

First step of defining traces is to establish *concurrent alphabet* (Σ, D). In this pair, Σ is a common alphabet (finite set) and $D \subseteq \Sigma \times \Sigma$ is an arbitrary reflexive and symmetric relation, called *dependency relation*. With dependency we associate, as another relation, an *independency relation* $I = \Sigma \times \Sigma \setminus D$. From other point of view, we can consider the dependency relation as a set of pairs of dependent letters. More precisely, the set of all *dependent pairs* $\mathscr{DP}$ is $\{\{a, b\} \mid aDb\}$.

Having concurrent alphabet we define a relation, that identifies similar computations. We say that word $\sigma \in \Sigma^*$ is in relation $\equiv_D$ with word $\tau \in \Sigma^*$ if and only if there exists a finite sequence of commutation of subsequent, independent letters that leads from σ to τ. Relation $\equiv_D \subseteq \Sigma^* \times \Sigma^*$ is a congruence relation (whenever it will not be confusing, relation symbol D would be omitted).

After dividing set Σ^* by the relation $\equiv$ we get a quotient monoid of all *finite traces* [3]. A single *trace* α is a member of $\Sigma^*/_\equiv$. This way, every word σ is related to a trace $\alpha = [\sigma]$, containing this word. This *canonical morphism* $[] : \Sigma^* \to \Sigma^*/_\equiv$ preserves length, catenation and projections to some subalphabets. Important examples of such subalphabets are all singletons and binary sets of dependent letters. In other words, for every $S \in \mathscr{DP}$, the operation Π_S is representative independent. Moreover, projections of a trace, as its arbitrary representative, to all prescribed subalphabets point interchangeably this trace.

This observation leads to another formulation of word equivalence. We can say, that two words σ and τ are equivalent if and only if for every set $S \in \mathscr{DP}$ their projections to S are equal. More precisely, $\sigma \equiv \tau \Leftrightarrow \forall_{S \in \mathscr{DP}} \Pi_S(\sigma) = \Pi_S(\tau)$. Very similar representation of Mazurkiewicz traces was given in [9].

Presented formulation of finite traces is well known and studied. However, problems are appearing when we try to get *infinite traces*. One of proposed definition is given in [6]. This construction starts from finite traces and goes through set of ideals of poset $(\Sigma^*/_\equiv, \sqsubseteq^\equiv)$. Simpler, but equivalent, definition could be found in [8]. It is based on projection representation of traces. Once more, two traces, finite or not, are equivalent if and only if their projections to every set from $\mathscr{DP}$ are equal.

Similarly, like in the sequential case, we can define prefix partial order on traces. It will be denoted by $\sqsubseteq^\equiv$. A trace α is prefix of trace β if and only if their projections

to every set from $\mathscr{DP}$ are in prefix relation as words. More precisely, $\alpha \sqsubseteq^{\equiv} \beta \Leftrightarrow \forall_{S \in \mathscr{DP}} \Pi_S(\alpha) \sqsubseteq \Pi_S(\beta)$.

Proposition 1. *Poset* $(\Sigma^{\infty}/_{\equiv}, \sqsubseteq^{\equiv})$ *is complete in Scott's sense.*

It is easy to see that definition of prefix order is not prescribed from words. The main reason is appearance of catenation in sequential definition. In contrast to words and finite traces, catenation in $\Sigma^{\infty}/_{\equiv}$ is not simple. Formulation that preserves representatives is at conflict with the intuition. Another idea preserves definition of prefix order and allows to add something to infinite traces (under specified circumstances). Two equivalent formulation of this idea can be found in [6, 8]. In second article, definition of catenation uses an reconstruction operation on projection sets.

4   Projection Sets

As we have claimed, trace is interchangeably circumscribed by a set of all its projections to the sets from $\mathscr{DP}$. It is natural to consider cartesian product of codomains of such projections. Elements of such cartesian product are called *projection sets*. To simplify the notation of discussion, we define the projection set as an arbitrary function $P : \mathscr{DP} \to \Sigma^{\infty}$, such that $P(\{a,b\}) \in \{a,b\}^{\omega}$. Every function of such property interchangeably circumscribe one element of previously described cartesian product. The family of all projection sets over the concurrent alphabet (Σ^{∞}, D) is denoted by $\mathscr{PS}(\Sigma^{\infty}, D)$ or, whenever it would not be confusing, by $\mathscr{PS}$.

Certainly, not every projection set is a set of projections for a trace. Those ones that are projections are called *reconstructible*. We assign the relationship between projection representation and trace by the projection function $\Pi_D : \Sigma^{\infty}/_{\equiv} \to \mathscr{PS}$. This function transforms a trace α to projection set $\Pi_D(\alpha)$, such that for arbitrary set $S \in \mathscr{DP}$ we have $\Pi_D(\alpha)(S) = \Pi_S(\alpha)$. The family of all reconstructible projection sets over the concurrent alphabet (Σ^{∞}, D) is denoted by $\mathscr{PR}(\Sigma^{\infty}, D) \subseteq \mathscr{PS}(\Sigma^{\infty}, D)$ or, if no confusion is caused, by $\mathscr{PR}$.

Example 1. Let $\Sigma = \{a, b, c\}$, where all letters are pairwise dependent. Let usconsider projection sets:

$$P_1 = \left\{ \begin{array}{l} (\{a,a\}, aa) \\ (\{a,b\}, abab) \\ (\{a,c\}, aca) \\ (\{b,b\}, bb) \\ (\{b,c\}, cbb) \\ (\{c,c\}, c) \end{array} \right\}, P_2 = \left\{ \begin{array}{l} (\{a,a\}, aa) \\ (\{a,b\}, abab) \\ (\{a,c\}, aca) \\ (\{b,b\}, bb) \\ (\{b,c\}, bbc) \\ (\{c,c\}, c) \end{array} \right\}, P_3 = \left\{ \begin{array}{l} (\{a,a\}, aa) \\ (\{a,b\}, aba) \\ (\{a,c\}, aca) \\ (\{b,b\}, bb) \\ (\{b,c\}, cbb) \\ (\{c,c\}, c) \end{array} \right\}.$$

Projection set P_1 is reconstructible (it is a projection representation of trace $[acbab]$), but projection sets P_2 and P_3 are not.

In the family $\mathscr{PS}$ we define a relation $\sqsubseteq^{\Pi} \subseteq \mathscr{PS} \times \mathscr{PS}$, that is a product relation of prefix orders in sets from $\mathscr{DP}$ given by the formula:

$$P_1 \sqsubseteq^{\Pi} P_2 \Leftrightarrow \forall_{S \in \mathscr{DP}} P_1(S) \sqsubseteq P_2(S).$$

Proposition 2. *The relation $\sqsubseteq^{\Pi}$ is a partial order. The poset $(\mathscr{PS}, \sqsubseteq^{\Pi})$ is complete in Scott's sense. Suprema and infima are calculated componentwise.*

5 Internal Conflict Free Projection Sets

We have noticed that reconstructible projection sets are properly contained in family of projection sets. In the difference of these two families are projection sets that can not be sets of appropriate projections of any trace. It involves that they can not be sets of appropriate projections of any word as well. There are two main reasons for such effect. Some projection sets can have different number of specified letter in values of subalphabets containing that letter, like P_3 in example 1. Other bad property is having some deadlocks, like P_2 in example 1. One can say that such projection sets have some internal conflicts. Let us define a family of projection sets that are free from such conflicts:

Definition 1. *The family of internal conflict-free projection sets is the least family $\mathscr{F}$ that fulfils the conditions:*

- $\forall_{S \in \mathscr{DP}} P(S) = \varepsilon \Rightarrow P \in \mathscr{F}$
- $P \in \mathscr{F} \wedge P$ *is finite* $\wedge (\exists_{c \in \Sigma} \forall_{S \in \mathscr{DP}} P'(S) = P(S) \Pi_S(c)) \Rightarrow P' \in \mathscr{F}$
- $(\forall_i P_i$ *is finite* $\wedge P_i \sqsubseteq^{\Pi} P_{i+1} \wedge P_i \in \mathscr{F}\}) \Rightarrow \sup P_i \in \mathscr{F}.$

First useful observation about family of all internal conflict-free projection sets is their closedness w.r.t. suprema of arbitrary upward subfamilies, also these containing infinite elements. Moreover, in some cases we can expand internal conflict-free projection set in same way, as in second condition of definition, even if it is not finite. The sufficient condition is finiteness of all values of subalphabets containing expanding letter. Another fact concerns relationship between partial order on internal conflict-free projection sets and their construction. If a finite conflict-free projection set P is a prefix of another finite conflict-free projection set P' then there exists a sequence of internal conflict-free projection sets $(P_i)_{i=1}^{k}$, such that $P = P_1 \wedge P' = P_k$ and

$$\forall_{i<k} \forall_{S \in \mathscr{DP}} \exists_{c_i \in \Sigma} P_{i+1}(S) = P_i(S) \Pi_S(c_i).$$

We show that internal conflict-free projection sets correspond to traces. Moreover, the correspondence is one-to-one.

Theorem 1. *The projection set P is an element of family $\mathscr{F}$ if and only if P is a representation of a trace.*

Now the Proposition 1 is a simple conclusion from the Theorem 1.

6   Reconstruction Operation and Catenation of Traces

In the end of Sect. 3 we have noticed that catenation of arbitrary traces can be defined using a special operation named reconstruction. Let us recall it very briefly.

The reconstruction takes an arbitrary projection set and transform it, in a fair way, to a trace. During the reconstruction, we start from a projection set P_0 and empty word υ_0. In the subsequent steps we try to increase the length of word υ_i by writing single letter and erasing it from the projection set P_i. To assure the fairness of this construction we use a special sequence seq_Σ that contains every letter from the alphabet Σ infinitely often. In ith step we use ith letter a of the sequence seq_Σ. If for every dependent pair $S \in \mathscr{DP}$ containing letter a value of the current projection set $P_i(S)$ starts with a, we write the letter a as the last letter of the new approximation $\upsilon_{i+1} = \upsilon_i a$ and erase it from every word $P_i(S)$ to get P_{i+1}. Otherwise the ith step is idle. At the end, as a result, we get the trace α containing limit of constructed sequence (υ_n). We will denote it by the function $R_D : \mathscr{PS} \to \Sigma^\infty/_\equiv$, such that $R_D(P) = \alpha$.

Let us cite a few useful properties of reconstruction function:

Proposition 3. *For arbitrary $\alpha \in \Sigma^\infty/_\equiv$ we have $R_D(\Pi_D(\alpha)) = \alpha$.*

Proposition 4. *For arbitrary projection sets P_1, P_2, such that $P_1 \sqsubseteq^\Pi P_2$, we have $R_D(P_1) \sqsubseteq^\equiv R_D(P_2)$.*

In the family of projection sets $\mathscr{PS}$, which is the cartesian product of some monoids $\{a,b\}^\infty$, we can define a natural, componentwise catenation $\cdot : \mathscr{PS} \times \mathscr{PS} \to \mathscr{PS}$.

Definition 2. *For arbitrary projection sets P_1, P_2 we define their catenation $P_3 = P_1 \cdot P_2$, such that $\forall_{S \in \mathscr{DP}} P_3(S) = P_1(S) \cdot P_2(S)$.*

As we have claimed, such definition is very natural. However, from our point of view, it also has very unpleasant property. It leads out of internal conflict-free projection sets.

Example 2. Let $\Sigma = \{a,b,c\}$, where only letters a and c are independent. Let us consider three projection sets:

$$
P_1 = \left\{
\begin{array}{l}
(\{a,a\},a^\omega) \\
(\{a,b\},a^\omega) \\
(\{b,b\},\varepsilon) \\
(\{b,c\},\varepsilon) \\
(\{c,c\},\varepsilon)
\end{array}
\right\}, \;
P_2 = \left\{
\begin{array}{l}
(\{a,a\},\varepsilon) \\
(\{a,b\},b^\omega) \\
(\{b,b\},b^\omega) \\
(\{b,c\},(cb)^\omega) \\
(\{c,c\},c^\omega)
\end{array}
\right\}, \;
P_3 = \left\{
\begin{array}{l}
(\{a,a\},a^\omega) \\
(\{a,b\},a^\omega) \\
(\{b,b\},b^\omega) \\
(\{b,c\},(cb)^\omega) \\
(\{c,c\},c^\omega)
\end{array}
\right\}.
$$

The sets P_1 and P_2 are internal conflict-free, but P_3 is the catenation of P_1 and P_2 which is not internal conflict-free from obvious reason (values of $\{a,b\}$ and $\{b,b\}$ are in internal conflict). The second kind of conflict, called deadlock, do not appear in catenation of two internal conflict-free projection sets.

Despite the fact that catenation of projection sets has described bad behaviour, we can use it to define catenation on traces. We can repair the result by calculating the value of reconstruction function of it.

Definition 3. *Let us consider two arbitrary traces* $\alpha, \beta \in \Sigma^{\infty}/_{\equiv}$.
Their catenation is a function $\cdot : \Sigma^{\infty}/_{\equiv} \times \Sigma^{\infty}/_{\equiv} \to \Sigma^{\infty}/_{\equiv}$ *given by formula*

$$\alpha \cdot \beta = R_D(\Pi_D(\alpha) \cdot \Pi_D(\beta)).$$

This definition preserves prefix order and is equivalent to that of [6]. Another way of defining catenation is to permit catenation of two reconstructible projection sets only if the result is also reconstructible or introduce new *zero-element* that is the result when catenation leads out of reconstructible projection sets [4]. The first approach gives an operation that is not associative, the second gives only partially defined operation and the third causes problems while defining partial order. We stay with the first definition.

Example 3. Let $\Sigma = \{a, b, c\}$, where only letters a and c are independent. Let us consider three traces $\alpha = [a^{\omega}]$, $\beta = [b]$ and $\gamma = [c]$. Their projection representations are:

$$\Pi_D(\alpha) = \left\{ \begin{array}{l} (\{a,a\}, a^{\omega}) \\ (\{a,b\}, a^{\omega}) \\ (\{b,b\}, \varepsilon) \\ (\{b,c\}, \varepsilon) \\ (\{c,c\}, \varepsilon) \end{array} \right\}, \ \Pi_D(\beta) = \left\{ \begin{array}{l} (\{a,a\}, \varepsilon) \\ (\{a,b\}, b) \\ (\{b,b\}, b) \\ (\{b,c\}, b) \\ (\{c,c\}, \varepsilon) \end{array} \right\}, \ \Pi_D(\gamma) = \left\{ \begin{array}{l} (\{a,a\}, \varepsilon) \\ (\{a,b\}, \varepsilon) \\ (\{b,b\}, \varepsilon) \\ (\{b,c\}, c) \\ (\{c,c\}, c) \end{array} \right\}.$$

Then we have $\alpha(\beta\gamma) = [a^{\omega}]$ and $(\alpha\beta)\gamma = [ca^{\omega}]$. It shows that catenation of traces is not associative.

Let us look more discerning into reconstruction operation and the essence of described reparation. Given in Propositions 3 and 4 properties of reconstruction suggest the following observation. Such reparation gives the greatest, in the meaning of prefix partial order on projection sets, internal conflict-free projection set that is not greater than starting projection set. This observation can be used to give another, equivalent definition.

For an arbitrary projection set P, let us consider a set of all its internal conflict-free prefixes, which will be denoted by *CFP*,

$$CFP(P) = \{P' \mid P' \text{ is internal conflict-free} \wedge P' \sqsubseteq^{\Pi} P\}.$$

First, we show that the set $CFP(P)$ is directed and its supremum is an internal conflict-free, not greater than P, projection set.

Proposition 5. *For arbitrary projection set P and projection sets $P_0, P_1 \in CFP(P)$, there exists a projection set $P_2 \in CFP(P)$, such that $P_0 \sqsubseteq^{\Pi} P_2$ and $P_1 \sqsubseteq^{\Pi} P_2$. Moreover, $supCFP(P) \in CFP(P)$.*

Now we can use *CFP*-sets to repair catenation. Trace catenation of two arbitrary, internal conflict-free projection sets P_1, P_2 can be given by formula:

$$P_1 \cdot_{\mathrm{tr}} P_2 = supCFP(P_1 \cdot P_2).$$

Theorem 2. *For arbitrary traces* $\alpha, \beta \in \Sigma^\infty/_\equiv$ *we have:*

$$\Pi_D(\alpha\beta) = \Pi_D(\alpha) \cdot_{\mathrm{tr}} \Pi_D(\beta).$$

7   Conclusions

The morphisms to some special free monoids completely describes the traces. We used that fact to give transparent and uniform definition of traces, which is intuitive and equivalent to previously given. The cartesian product of mentioned free monoids is used in this definition. Two characterisations of projection sets related to traces were presented. The first one is based on the reconstruction algorithm that turns an arbitrary projection set to a trace. The second one constructively initiates the internal conflict-free projection sets. Finally, two methods of using the componentwise catenation of the projection sets to define the catenation of the traces are described.

References

1. Cartier, P., Foata, D.: Problèmes combinatoires de commutation et réarrangements. Lecture Notes in Mathematics, vol. 85. Springer, Berlin (1969)
2. Diekert, V.: Combinatorics on Traces. Springer, Berlin (1990)
3. Diekert, V., Rozenberg, G. (eds.): The Book of Traces. World Scientific, Singapore (1995)
4. Gastin, P.: Infinite traces. In: Guessarian, I. (ed.) LITP 1990. LNCS, vol. 469, pp. 277–308. Springer, Heidelberg (1990)
5. Gastin, P., Petit, A.: Infinite traces. In: Diekert, V., Rozenberg, G. (eds.) The Book of Traces, ch. 11, pp. 393–486. World Scientific, Singapore (1995)
6. Kwiatkowska, M.Z.: Defining process fairness for non-interleaving concurrency. Foundations of Software Technology and Theoretical Computer Science 10, 286–300 (1990)
7. Mazurkiewicz, A.: Concurrent program schemes and their interpretations. Daimi report pb-78, Aarhus University (1977)
8. Mikulski, Ł.: Projection representation of Mazurkiewicz traces. Fundamenta Informaticae 85, 399–408 (2008)
9. Shields, M.W.: Concurrent machines. The Computer Journal 28(5), 449–465 (1985)

The Comparison of an Adapted Evolutionary Algorithm with the Invasive Weed Optimization Algorithm Based on the Problem of Predetermining the Progress of Distributed Data Merging Process

Daniel Kostrzewa and Henryk Josiński

Abstract. The goal of the project was to adapt the idea of the Invasive Weed Optimization (IWO) algorithm to the problem of predetermining the progress of distributed data merging process and to compare the results of the conducted experiments with analogical outcomes produced by the evolutionary algorithm. The main differences between both compared algorithms constituted by operators used for transformation of individuals and for creation of a new population were taken into consideration during the implementation of the IWO algorithm. The construction of an environment for experimental research made it possible to carry out a set of tests to explore the characteristics of the tested algorithms. The results of the conducted experiments formed the main topic of analysis.

Keywords: evolutionary algorithm, Invasive Weed Optimization algorithm, query optimization, distributed database, genetic operators.

1 Introduction

Because database users formulate queries in a non-procedural manner, a query must be transformed into a cost effective execution plan. The transformation of a query addressed to a distributed database requires the predetermination of the progress of distributed data merging process. The authors propose the Invasive Weed Optimization (IWO) algorithm as a method for performing that task and compare the results of the experiments with the outcomes of the research based on the evolutionary algorithm described in [2].

Daniel Kostrzewa · Henryk Josiński
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail:{daniel.kostrzewa,henryk.josinski}@polsl.pl

2 Problem Formulation

The distributed data merging process is initialized by a user query addressed to
a distributed database and produces a set of records called a *final set*. This is a
multistage process that consists of many *merging operations*. The goal of a sin-
gle operation is to merge two sets of records; each of them can be either get
from local databases forming a distributed database (*original set*) or built by one
of earlier realized merging operations (*intermediate set*). A merging operation
can be carried out by any workstation where a merging program has been in-
stalled. The most frequently performed merging operation is a *join* [8]. Hence,
the goal of the research was to construct an optimization algorithm that deter-
mines the order and the performers of joins in such a way that the final set ar-
rives at the *target node* (a workstation where a user is waiting for an answer to
his distributed database query) as fast as possible. A search space for the consid-
ered issue consists of a set of *query execution plans* where a query is represented
by a *join graph* – a set of vertices connected by undirected edges. Vertices are
equivalents of original sets. Each edge denotes a fact of existence of a *connect-
ing expression* between two original sets that constitutes a criterion for matching
records from these sets to each other. The distinctive shapes of a join graph are the
following:

- a *star* graph, in which one of the record sets appears in each connecting expres-
  sion (is a central vertex of the graph presented in Fig. 1a),
- a *chain* graph, where two sets of records (constituting extreme vertices of the
  graph presented in Fig. 1b) appear only once in connecting expressions, whereas
  the remaining sets – twice.

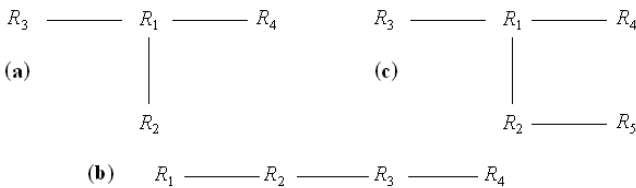The other shapes of query graphs are called *irregular* (Fig. 1c).

Fig. 1 Shape of a join graph: **(a)** star, **(b)** chain, **(c)** irregular

An exemplary join sequence for a centralized database and the join graphs pre-
sented in Figs. 1a, 1b consists of 3 operations producing successively the interme-
diate sets R_{21}, R_{321} and the final set R_{4321}: $R_{21} = join(R_1, R_2)$; $R_{321} = join(R_3, R_{21})$;
$R_{4321} = join(R_4, R_{321})$.

The optimization algorithm should find a solution for a considered problem for
both distributed databases and centralized ones. However, the aspect of data distri-
bution requires additionally the determination of workstations where the particular
joins will be performed.

The overview of bibliography describing the methods for solving the discussed problem would be unusually spacious. The studies related to the usage of evolutionary algorithm were mentioned in [2], whereas the IWO algorithm, according to the authors' knowledge, has never been used to determine the order of joins.

3 Computational Model for the Goal Function

A complete formulation of an optimization task requires a definition of a goal function, interpreted in the considered problem as an estimated cost of joins execution. The model implemented in the IWO algorithm consists of the formulas for estimation of parameters characterizing a set of records generated by a join – its cardinality and length of a single record – as well as of the formulas for cost estimation of a join and of a data transfer.

Length of record in a set generated by a join is computed according to the formula:

$$len(join(X,Y)) = \eta\,(len(X) + len(Y)) \tag{1}$$

The symbols $len(X)$, $len(Y)$ denote lengths of records of both joined sets, η is a *reduction factor*, $\eta \in (0,1]$. The maximum value of η will be used when the connecting expression for a join is not given and in that case a join must be executed as a cartesian product of sets.

Computation of cardinality of a set generated by a join requires consideration of the following cases:

- if the connecting expression is not given, the cardinality is equal to the product of cardinalities of both joined sets $card(X)$, $card(Y)$:

$$card(join(X,Y)) = card(X) \times card(Y), \tag{2}$$

- if a relationship between values of records attributes appearing in the connecting expression is symbolically described as '1:1', that is a single attribute value from a smaller set is related to maximum one attribute value from the other set, the estimated cardinality is equal to the cardinality of a smaller set:

$$card(join(X,Y)) = min(card(X), card(Y)), \tag{3}$$

- if a relationship between values of records attributes appearing in the connecting expression is symbolically described as '1:n', that is a single attribute value from a set X is related to many attribute values from the set Y, and simultaneously a single attribute value from a set Y is related to only one attribute value from the set X, the estimated cardinality is equal to the cardinality of a set Y,
- in all other cases the following formula is used:

$$card(join(X,Y)) = \frac{1}{sel} \times min(card(X), card(Y)). \tag{4}$$

The symbol sel represents a factor of uniqueness for values of attributes appearing in the connecting expression: $sel = uniq(X.A)/card(X)$. Function $uniq(X.A)$

computes the number of unique values appearing in the attribute A from the set X. For simplification, during the experiments the same value of *sel* was assigned to all sets.

The estimated cardinalities along with computational power v of a workstation performing a join constitute elements of the formula for estimation of the cost of a join execution C_J [1]:

$$C_J = \frac{card(X) + card(Y) + card(join(X,Y))}{v}. \tag{5}$$

The cost of transfer C_T of a set Z between workstations A and B is estimated using the number of transferred records $card(Z)$, their length $len(Z)$ and network throughput $v_{A \to B}$ according to the formula [1]:

$$C_T = \frac{card(Z) \times len(Z)}{v_{A \to B}}. \tag{6}$$

The costs values are expressed in units of time. The value of the goal function is interpreted as a length of time interval starting along with the first operation and ending at the moment of arrival of the final set at the target node. A value of the fitness function is calculated as reciprocal of the goal function.

4 Methods for Solving the Problem

The IWO algorithm as a relatively new optimization method will be described in detail. Presentation of the evolutionary algorithm will be limited to its distinctive features.

4.1 Idea of the IWO Algorithm

The IWO algorithm is an optimization method, in which the penetration strategy of the search space (similarly to the evolutionary algorithm) is based on the transformation of a complete solution into another one [3]. Its idea was inspired by observation of dynamic spreading of weeds and their quick adaptation to environmental conditions [4, 5, 7].

There are three main stages of the IWO algorithm:

1. Sowing of seeds of the first weed population.
2. Reproduction – growth of plants and sowing of seeds.
3. Selection of the plants for the next population based on their fitness function.

During the first stage of the algorithm the seeds are randomly sowed in the search space, what corresponds with the random generation of a finite number of solutions of the considered problem. Cardinality of the initial weed population belongs to the algorithm parameters.

The number of seeds sowed by a single plant during the second stage depends on a degree of its adaptation to environmental conditions expressed in form of a fitness function – the greater the value of the fitness function for a given weed, the greater the number of its seeds. This relationship has a linear character. The distance between the place where a seed falls on the ground and its parent plant is described by a normal distribution with a mean equal to 0 and a standard deviation truncated to nonnegative values and decreasing along with each algorithm iteration according to the following formula [4, 5, 7]:

$$\sigma_{iter} = \left(\frac{iter_{\max} - iter}{iter_{\max}} \right)^m (\sigma_{\text{init}} - \sigma_{\text{fin}}) + \sigma_{\text{fin}}. \tag{7}$$

The symbol σ_{iter} denotes a standard deviation computed for the iteration $iter$, $iter \in [1, iter_{\max}]$. As the next weed generation is processed in each iteration, the total number of iterations ($iter_{\max}$) is equivalent to the total number of weed populations. The symbols σ_{init}, σ_{fin} represent, respectively, initial and final values of the standard deviation, whereas m is a nonlinear modulation factor.

The distance between the parent plant and a new weed expresses the degree of difference between both plants. Its value is also a number of transformations of the parent plant creating a descendant.

In the second stage of the algorithm weeds can sow a considerable number of seeds. Consequently, during the third stage the weeds characterized by the low values of the fitness function are eliminated. In such a way the cardinality of the subsequent populations remains constant.

The second stage and the third one are repeated many times. The number of iterations and the other symbols taken from (7) form the parameters of the algorithm.

4.2 Representation of the Solution

The important feature of both algorithms is a common representation of a problem solution – a determined order of joins along with performers assigned to each operation – in a form of an individual (e.g., a weed from the IWO algorithm) illustrated in Fig. 2.

S_1	S_2	...	...	S_i	S_j				
W_1		...		W_k					

Fig. 2 Representation of a solution in both algorithms

The symbols S_i, S_j identify sets of records being joined by the workstation W_k. A triple (S_i, S_j, W_k) is called a *gene*. Number of genes in an individual is equal to $n - 1$ where n denotes the number of original sets mentioned in a query (original sets receive numbers from 1 to n). Configuration of genes accords with the order of joins execution. A result of the join described by the first gene becomes a set identified by the number $n + 1$ whereas the subsequent intermediate sets get subsequent

natural numbers. The intermediate sets will be processed in future joins, so they will appear in the next genes under restriction that a gene that includes an identifier of an intermediate set can be inserted in an individual not before a gene representing a join producing that intermediate set.

4.3 Modifications of an Individual

A transformation of a parent plant producing a new weed is performed on a single gene as a random change of a join performer or as an exchange of two randomly chosen sets participating in different joins. The probability values for both types of transformation belong to the parameters of conducted research. The mechanism applied in the IWO algorithm corresponds with the mutation operator used in the evolutionary algorithm.

A technique of choice of individuals modified by genetic operators to form a new population differs from the simple elimination of the weeds characterized by the low values of the fitness function. The method implemented in the evolutionary algorithm is based on the *roulette wheel selection* [6].

A recombination operator is a variety of *order crossover* [6]. A detailed analysis of particular phases of recombination realized by the evolutionary algorithm solving the considered problem was presented in [2].

5 Experimental Research

The experiments were carried out for both a distributed database and a centralized one. A user query was represented by a join graph in the shape of star, chain or in irregular form. The number of original sets forming the join graph in the subsequent series of tests was taken from the following group of values: $\{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$. The features of the particular original sets (length of records and cardinality), workstations (computational power) and network segments (throughput) have been differentiated. A series of 100 tests was carried out for each database type and for each shape of the join graph formed by the given number of the original sets. A stop criterion for a single test was defined as a number of processed populations. The workstation used for experiments is described by the following parameters: Intel Core2 Quad Q6600 2.4 GHz processor, RAM 2 GB 800 MHz. The following values were assigned to the parameters related to the computational model for the goal function:

- a factor of record length reduction $\eta = 0.9$,
- a factor of uniqueness for values of attributes appearing in the connecting expression $sel = 0.8$,
- probabilities of gene transformations: change of a join performer and exchange of sets – respectively: $1/3$ and $2/3$.

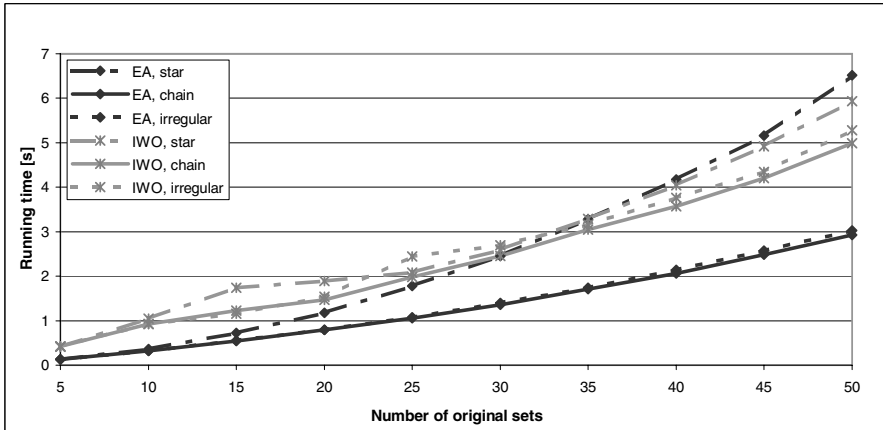The parameters of the IWO algorithm received the following values:

Fig. 3 The average running time of optimization algorithms for centralized data

- population cardinality – a constant value 60 in all populations,
- number of populations $iter_{max} = 60$,
- minimum and maximum number of seeds sowed by a weed – respectively: 1 and 10,
- initial (σ_{init}) and final (σ_{fin}) values of the standard deviation – respectively: 5 and 1,
- nonlinear modulation factor $m = 3$.

The values of the evolutionary algorithm parameters were the following:

- population cardinality – 100,
- number of populations – 110,
- probabilities of mutation and crossover – respectively: 0.1 and 0.9.

The relationship between running time of both algorithms and number of original sets used in a query for centralized and distributed data is illustrated in Figs. 3, 4, respectively ('EA' stands for 'evolutionary algorithm').

Analysis of Figs. 3, 4 shows that in the majority of tests the evolutionary algorithm solved the considered problem for the given number of original sets faster than the IWO algorithm. This fact can be explained by the usage of crossover operator, importance of which in the evolutionary algorithm is evaluated as fundamental, whereas the mutation is considered as a operator of secondary meaning. The idea of the IWO algorithm excludes the application of crossover, because this operator requires two parent individuals, while the transformation – only one.

The remarkable similarities between the data points marked as 'EA, chain' and 'EA, irregular' are caused by the fact that the tested irregular join graphs include many chain-shaped parts.

Another criterion used for comparison of both algorithms is based on the quality of solutions produced by them. As the goal function can be interpreted as an estimated user's waiting time for a final set, the lower the value of the goal function, the better the quality. The average values of the goal function computed for the star
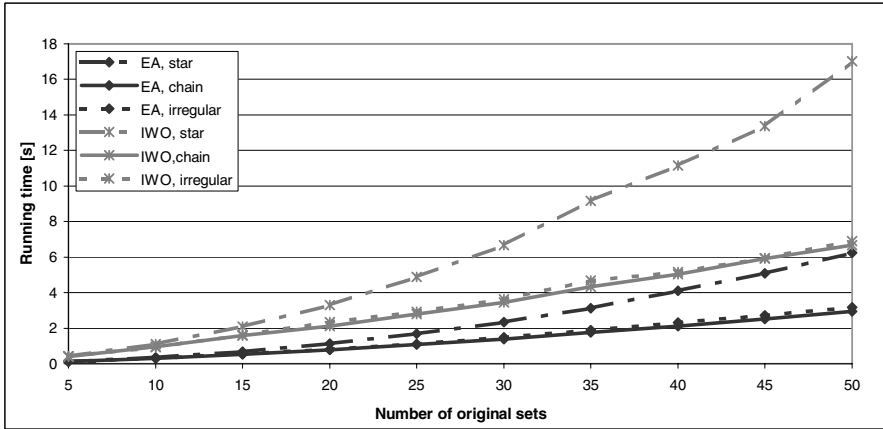
Fig. 4 The average running time of optimization algorithms for distributed data
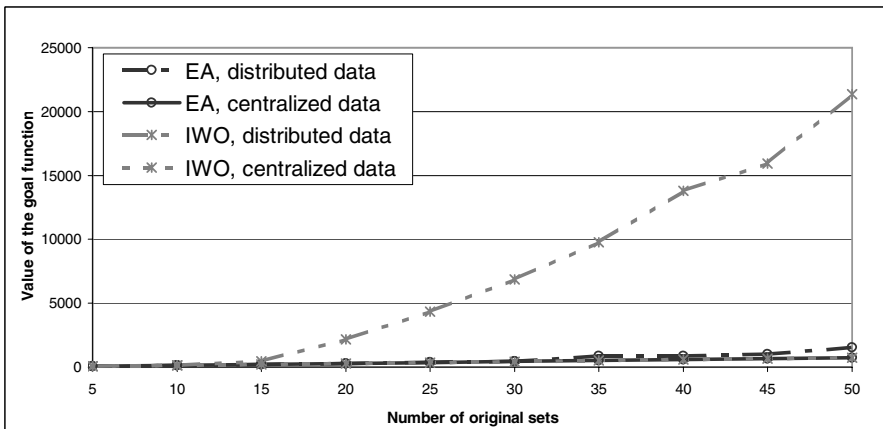

Fig. 5 The average values of the goal function for star-shaped join graph

shape of the join graph are shown in Fig. 5 and for chain and irregular shapes of the join graph – in Fig. 6.

The majority of tests shows that the solutions produced by the IWO algorithm are of superior quality, first of all for distributed data and in case of chain-shaped or irregular join graph (Fig. 6). However, solutions given by the IWO algorithm for star-shaped join graphs differ from this trend, especially for distributed data. Explanation for the advantage of the evolutionary algorithm in this case is based on the technique of crossover, which finds an appropriate join performer much faster than the drawing method implemented in the IWO algorithm. Noticeable difference between values of the goal function computed for star-shaped join graph and values related to the graphs of the other shapes is a consequence of the input data (first of all cardinalities of original sets).
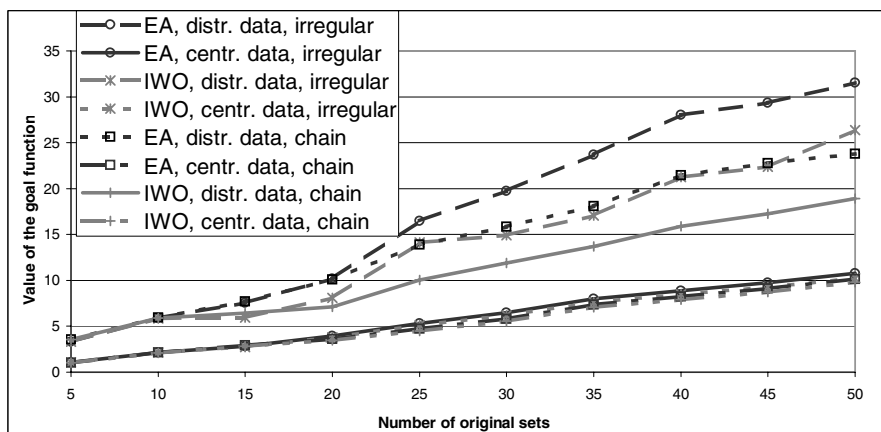
Fig. 6 The average values of the goal function for chain-shaped and irregular join graphs

6 Conclusions

The analysis of the research results leads to the following conclusions:

- the longer running time of the IWO algorithm is balanced by the higher quality of solutions generated by itself, particularly for queries represented by chain-shaped or irregular join graphs,
- the advantage of the evolutionary algorithm is revealed in case of queries operating on centralized data, because the running time of this method is shorter and the quality of solutions generated by both algorithms seems to be almost identical, however, in most cases the quality is slightly higher for the IWO algorithm.

The next phase of the research will comprise experiments related to the change of weeds spreading method and the construction of a hybrid version of the optimization algorithm.

References

1. Chen, Y.: A systematic method for query evaluation in distributed heterogeneous databases. Journal of Information Science and Engineering 16(4) (2000)
2. Kostrzewa, D., Josiński, H.: Planning of the process of distributed data merging by means of evolutionary algorithm. In: Kozielski, S., Małysiak, B., Kasprowski, P., Mrozek, D. (eds.) Architecture, Formal Methods and Advanced Data Analysis. Wydawnictwa Komunikacji i Łączności, Gliwice, Poland (2008) (in Polish)
3. Lanzelotte, R.S.G., Valduriez, P., Zaït, M.: On the effectiveness of optimization search strategies for parallel execution spaces. In: Proceedings of the 19th Conference on Very Large Data Bases, Dublin, Ireland (1993)
4. Mallahzadeh, A.R., Oraizi, H., Davoodi-Rad, Z.: Application of the invasive weed optimization technique for antenna configurations. In: Progress in Electromagnetics Research (2008)

5. Mehrabian, R., Lucas, C.: A novel numerical optimization algorithm inspired from weed colonization. Ecological Informatics 1(4) (2006)
6. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Wydawnictwa Naukowo-Techniczne, Warsaw (1999) (in Polish)
7. Sepehri Rad, H., Lucas, C.: A recommender system based on invasive weed optimization algorithm. In: Proceedings of the IEEE Congress on Evolutionary Computation, Singapore (2007)
8. Ullman, J.D., Widom, J.: A First Course in Database Systems. Wydawnictwa Naukowo-Techniczne, Warsaw (1999) (in Polish)

Cumulation of Pheromone Values
in Web Searching Algorithm

Urszula Boryczka and Iwona Polak

Abstract. In this paper, we propose a new ant-based searching algorithm called Seangàn. We describe a process of stigmergy and accumulation of pheromone values, leading to a degree of self-organization brought about through the independent actions and iterations of its individual agents. We use it in the construction in our continually evolving system, Seangàn. We discuss some of the issues raised and attempt to explain some of its success as well as account for its failings. We analyze the main characteristics of the algorithm and try to explain the influence of parameters value on the behavior of this system.

Keywords: ant colony optimization, stigmergy, pheromone cumulation, web searching.

1 Introduction

This paper introduces pheromone-based system, a novel approach to the adaptive searching in the Internet. Seangàn is a distributed, mobile agents system that was inspired by recent work on the ant colony metaphor. This paper is organized as follows: Section 2 gives a detailed description of swarm intelligence, selforganization and its role in ACO. Section 3 presents the Seangàn algorithm. In Sect. 4, we present some of the results that have been obtained during the searching process. The last section concludes and discusses future evolutions of our approach.

Urszula Boryczka · Iwona Polak
Institute of Computer Science, University of Silesia,
Będzińska 39, 41-200 Sosnowiec, Poland
e-mail: `urszula.boryczka@us.edu.pl`

2   Swarm Intelligence and Ant Colony Optimization

Swarm Intelligence (SI) is an intriguing distributed intelligent paradigm for solving optimization problems that originally task its inspiration from the biological examples by swarming, self-organizing foraging phenomena in social insects. There are many examples of swarm optimization techniques, such as: Particle Swarm Optimization, Artificial Bee Colony Optimization and Ant Colony Optimization (ACO). This last approach deals with artificial systems, inspired by the natural behaviors of real ants, especially foraging behaviors based on the pheromone substances laid on the grounds.

The fundamental concept underlying the behavior of social insects is self-organization. SI systems are complex systems – collections of simple agents that operate in parallel and interact locally with each other and their environment to produce emergent behavior.

The basic characteristic of metaheuristics from nature could be summarized as follows [3]:

- they model a phenomenon in nature,
- they are stochastic,
- in case of multiple agents, they often have parallel structure,
- they use feedback information for modifying their own parameters – they are adaptive.

The foraging behavior of ants [4, 5, 8] has inspired a novel approach to distributed optimization, Ant Colony Optimization (ACO). Most of current ACO [2, 7] applications are either in combinatorial optimization or in communications networks routing. Ant System (AS), the first ACO algorithm, was applied to the traveling salesman problem (TSP). Although it had limited success from a performance standpoint it opened up the road to improvements and new applications. Ant colony System (ACS), a modified version of AS that extends AS and includes local search routines, exhibits top performance on asymmetric TSPs. A whole family of ACO algorithms is now applied to many different combinatorial optimization problems ranging from the quadratic assignment problem and the sequential ordering problems to vehicle routing and graph coloring problems. ACO algorithms for routing in communication networks, also called ant routing algorithms, have been particularly successful. This success is probably due to the dynamic nature of the routing problem which is well matched to the distributed and adaptive characteristics of ACO algorithms.

3   Seangàn – Pheromone Representation of the Relevance

Seangàn takes inspiration from previous work on artificial ant colonies techniques to solve combinatorial optimization problems [6]. The core ideas of these techniques are:

- the use of repeated and concurrent simulations carried out by a population of artificial agents called 'ants' to generate new solutions to the problem,

- the use by the agents of stochastic local search to build the solution in an incremented way,
- the use of information collected during past simulations to direct future search for better solutions [3].

In this paper we present an improved version of the ACO approach based on pheromone cumulative values in different sites for web searching. The most interesting feature of this algorithm is the formation of different regions based on corresponding user queries and the relevant subset of objects which can be represented by the pheromone values at each site (web page).

We propose to introduce the individually and group formation of the pheromone representation the quality of the relevance. The pheromone accumulation utilizes a common ratio, which is determined for the user query individually. Depending on this type of the query, a number of regions (clusters) represented by subtrees are analyzed through the data sets with the appropriate coefficients assigned to them. The query represented as a set of keywords, each of which can be identified uniquely from our data sets and compare with various web pages. The evaluation function for document d (node) in our network is determine by the correlation between this document d_j and the query q constructed by the set of terms k. The pheromone value depends on the frequency of the specific term in analyzed web pages. This frequency measure may be presented as follows:

$$tf_{d_jk} = tf_{d_jk}^{\text{BODY}} + \alpha \times tf_{d_jk}^{\text{TITLE}} + \beta \times tf_{d_jk}^{\text{DESCR.}} + \gamma \times tf_{d_jk}^{\text{KEYW.}} + \delta \times tf_{d_jk}^{\text{URL}}, \quad (1)$$

where:

- $tf_{d_jk}^{\text{BODY}}$ is a frequency of the term k in body of a web page,
- $tf_{d_jk}^{\text{TITLE}}$ is a frequency of the term k in the title,
- $tf_{d_jk}^{\text{DESCRIPTION}}$ is a frequency of the term k in a description of that document,
- $tf_{d_jk}^{\text{URL}}$ is a frequency of the term k in URL address.

The values of parameters: α, β, γ, and δ are established individually during the search process. Using this formula, each of the specific sites is assigned a probability measure, called as a random proportional rule in ACO. Then data sites are ordered according the descending values of the pheromone, occurring in the probability measures – transition rule. Each data site is partitioned into numerous regions (clusters) representing different subjects of the searching thematic range of study.

Agents are distributed across different regions (clusters) and then they analyze these pages accordingly to the accumulation of pheromone gathered during the searching process. In this situation we observe a subset of agents which formally identified with the group of pages. Agents try to find the relevant pages according the pheromone values cumulated in the node. This method assumes that all web pages in the analyzed subtree are similar and represent a class of documents which are interesting for the query. The sequence of visiting nodes are reinforce by the calculated pheromone values. The hierarchy of the analyzed web pages is represented in Fig. 3.

Overall, the effect of ants on the network is such that nodes (sites) which are visited frequently by predecessors will be favored nodes when building paths to route new links. This has the effect links close to the ant source node get higher reinforcement that are far away. There is a strong and complex interplay among routing nodes, routing of ants, updating the pheromone table.

The value of the pheromone τ'_{d_j} for each web page d_j is computed as follows:

$$\tau'_{d_j} = \zeta \times \frac{tf_{d_jk}}{\sum\limits_{j=1}^{K} tf_{d_jk}} \times \frac{1}{h+1}, \tag{2}$$

where:

- ζ is a cumulation coefficient equal to 1 in our experiments,
- tf_{d_jk} is a frequency of a term on the web page d_j (early described in 1),
- K is a set of all documents attached to the term k,
- h is a height of tree constructed for the analyzed terms.

The value of the pheromone τ''_{k_l} for the node (cluster) k in Seangàn system is determined as follows:

$$\tau''_{k_l} = \begin{cases} \zeta \times \dfrac{\sum\limits_{j=1}^{K} \tau'_{d_j}}{|K|}, & \text{if term } k \text{ is a leaf of the tree structure,} \\[2em] \zeta \times \dfrac{\sum\limits_{l=1}^{N} \tau''_{k_l}}{|N|}, & \text{otherwise,} \end{cases} \tag{3}$$

where:

- $|K|$ is a cardinality of a set K,
- N is a set of children of analyzed node,
- $|N|$ is a cardinality of N.

The idea presented in this article is similar to the Bilchev and Parmee approach [1]. Authors suggest considering a finite set of regions (initial links) at each iteration of the algorithm: agents are send to these regions, from which they explore random–proportionally selected directions within a coefficient of exploration and range of ant. Agents reinforce their paths according to their performance and coefficient of similarity/correlation. Trails evaporate and create a new region for exploration. This algorithm operates of three different levels:

- individual search strategy of individual agents,
- the cooperation between ants using pheromone trails to focus on searching serendipitous and receptive neighborhood,
- the exchange of information between different regions performed by some kind of 'diffusion' similar to a crossover in a genetic algorithm. We define the ant range dedicated to the special part of the network (nodes). When some nodes

will not be accepted as a 'tastiest morsels' by the ant algorithm, so that the path leading to it will soon decay.

Search and optimal foraging decision-making of ants can be compare with bacterial swarm foraging behavior [9], where we observe the chemical attractant evaporation concerning the environment and another communication medium – a slime trail. We may observe the similar process of diffusion after chemotactic steps of presented simulations. ACO owns a number of features that are important to computational problem solving:

- it is relatively simple and easy to understand and then to implement,
- it offers emergent complexity to deal with other optimization techniques,
- it is compatible with the current trend towards greater decentralization in computing,
- it is adaptive and robust and it is enable to cope with noisy data.

In order to design a more global approach and to avoid inefficient searching we introduce the following parameters:

- the initialization of the pheromone trail use the value depending on the parameter τ_0,
- the parameter m establishes the number of ants. Ants use the classical strategy of searching depending on the parameter q_0 (exploration/exploitation),
- agents use local pheromone updating rule exploiting the idea of evaporation $-\rho$,
- the evaluation function for document d (node) $- tf_{d_jk}$ in our network is determine by the correlation between this document d_j and the query q.

Seangàn combines the quality, effectiveness of searching process as well as the synergistic effect obtains by pheromone changes. So the mechanism of diffusion the cumulative values of pheromone use the standard evaporation process:

1. when an agent visits a node, algorithm cumulates the value according to the validation of this document,
2. this cumulative value will be diminished according to the evaporation factor (ρ),
3. the process of diffusion depends on two parameters: the set of children N depending on the particular node k, which has a possibility of cumulation the pheromone values,
4. in case of relevant documents we have an opportunity to increase weights of the appropriate nodes (via the cumulation parameter $\zeta \in \langle 1, 10 \rangle$).

The local updating rule is performed as follows:

$$\tau_d(t+1) = (1-\rho)\tau_d(t) + \tau'_{d_j}.$$ (4)

The global pheromone updating rule is executed as follows:

$$\sum_{i=0}^{\text{card}(N)} \Delta \tau_{d_j} = (1-\rho) \times \tau_{d_j} + \tau''_{k_l}.$$ (5)

Table 1 Results

No of exp.	Term	Rel. nodes	% of all nodes	Complet.	Accur.	Time [s]
1	merlin	26	2.12%	0.7826	1	1.25
2	nessie	23	1.87%	0.8571	1	1.32
3	haggis	9	0.73%	1	1	0.85
4	triskel	14	1.14%	0.5833	1	1.16
5	scones	7	0.57%	1	1	0.94
6	ciasteczka	16	1.30%	0.5	1	0.89
7	leprikon	12	0.98%	0.5	1	0.91
8	polka	12	0.98%	0.6	1	0.86
8	johnnie walker	24	1.96%	0.55	1	1.59
9	arthur griffith	39	3.18%	0.5926	0.8421	1.88
10	penguin books	25	2.04%	0.5	1	1.69
11	lorica hamata	7	0.57%	1	1	2.04
12	merlin artur graal	73	5.95%	0.6136	1	2.64
			average	0.7279	1	1.02

We consider a population of m ants. Ants are located in nodes of the highest rank values and they try to find the areas of their interest. The regions (clusters) created during the search process correspond to the nodes (sites) which have the greater values of pheromone. The number of visitors and the correlation process corresponds to these values.

4 Experiments

To evaluate the performance of Seangàn, we have run experiments with information about culture, history and other curiosities from the web pages of Ireland. There are 1227 web pages in our repository, including 179 nodes in specific term tree and 141 leafs. It is an example of a data set with a high thematic cohesion, so the results obtained during the search process should be optimistic.

We have tested these networks with different sets of parameters. The final parameter settings have been determined by first optimizing the parameters on each of these examples of data networks separately and then averaging the parameters over the eight different queries.

In Seangàn there are a number of parameters that need to be set. We set the following values: System configuration:

- number of ants $m = 75$,
- number of iterations $i = 25$,
- local initial pheromone $\tau_0 = 0.001$,
- evaporation ratio $\rho = \alpha = 0,05$,
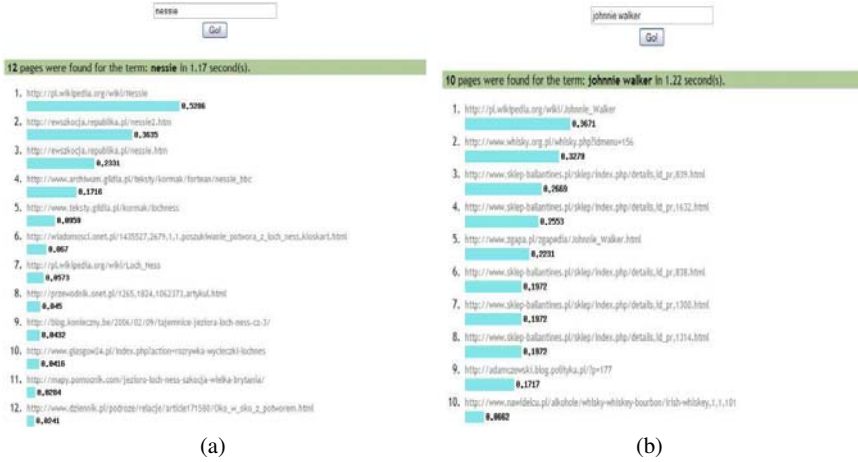- cumulation parameter $\zeta \in \langle 1, 10 \rangle$.

Fig. 1 Representation of pheromone values for the: (a) 2nd query, (b) 8th query

Figure 1a presents an example of a search process for a query: 'nessie'. The number of a relevant nodes is equal to 12 with the pheromone values range from 0.5286 to 0.0241. We analyze the whole data set. As it shown in Fig. 1b, the performance of Seangàn agents could find relevant documents in the longer question: 'Johnie Walker', we obtain the accuracy of searching process equal to 100% with the cost of the exploration this network – 1.96%. This good results obtained in all cases we can explain by the fact, that this information is located in small regions and queries are formulated with a satisfying precision.

The conclusions below can be drawn from the tests we have performed, but the reader should keep in mind that these conclusions are limited to the tested networks (Table 1). The number of ants in Seangàn does not seem to be a critical parameter in the case of a sequential implementation. The distribution of ants depends on the network size. Two properties of this approach seem to be crucial: the value of parameters: ρ and ζ are really important in the performing of the cumulation
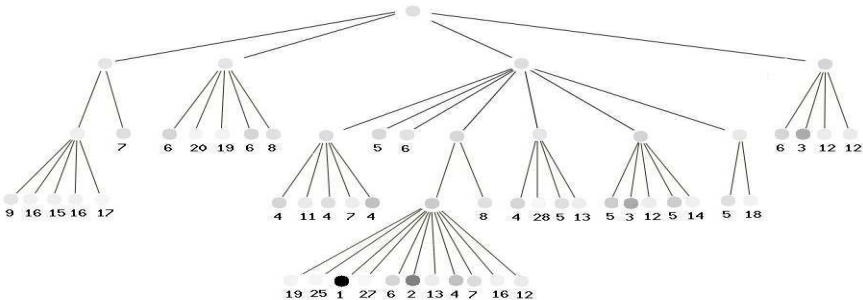

Fig. 2 An example of the tree search

process. In addition, the initial value of pheromone and updating the pheromone values' procedures guarantee fruitful communication between agents and they are quite important for the location of profitable regions. The lack of comparison in this phase of experimental study can be explained by the fact that we analyze a specific polish characteristics concerning search process. All in all, the experiments performed when testing the web sites suggest that the effectiveness (about 70%) is satisfying (Table 1). Preliminary tests of Seangàn on different queries seem to indicate that this algorithm can achieve robust performance for all the tested networks.

5   Conclusions

In this paper, we have proposed a new search algorithm inspired by the behavior of real ants. This algorithm has many features such as multiple local search processes concentrated around multiple regions, a strategy of pheromone cumulation sensitive to the success of evaluated nodes and the number of ants. Now we are using techniques of evolutionary biology to design information systems. These systems are designed to cooperate and interact with other memberships of the multi-agent system. Seangàn mimic natural systems use of stigmergy. We obtain the effectiveness of the searching process by the traveling through stigmergic medium-pheromone. Experimental studies suggest that Seangàn may achieve interesting results. We plan in the future to evaluate Seangàn with larger data sets and with several web sites to validate our approach.

References

1. Bilchev, G., Parmee, I.C.: The ant colony metaphor for searching continuous design spaces. In: Fogarty, T.C. (ed.) AISB-WS 1995. LNCS, vol. 993. Springer, Heidelberg (1995)
2. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence. From Natural to Artificial Systems. Oxford University Press, Oxford (1999)
3. Corne, D., Dorigo, M., Glover, F.: New Ideas in Optimization. McGraw-Hill, New York (1999)
4. Deneubourg, J.L.: Personal communication. Université Libre de Bruxelles, Brussels, Belgium (2002)
5. Deneubourg, J.L., Aron, S., Goss, S., Pasteels, J.M.: The self–organizing exploratory pattern of the Argentine ant. Journal of Insect Behavior 3, 159–168 (1990)
6. Dorigo, M., Gambardella, L.M.: Ant colonies for the Traveling Salesman Problem. Biosystems 43, 73–81 (1997)
7. Dorigo, M., Stützle, T.: Ant Colony Optimization. The Massetchussets Institute of Technology Press (2004)
8. Goss, S., Aron, S., Deneubourg, J.L., Pasteels, J.M.: Self–organized shortcuts in the Argentine ant. Naturwissenschaften 76, 579–581 (1989)
9. Passino, K.M.: Biomimicry for Optimization, Control, and Automation. Springer, London (2005)

Mining for Unconnected Frequent Graphs with Direct Subgraph Isomorphism Tests

Łukasz Skonieczny

Abstract. In the paper we propose the algorithm which discovers both connected and unconnected frequent graphs from the graphs set. Our approach is based on depth first search candidate generation and direct execution of subgraph isomorphism test over database. Several search space pruning techniques are also proposed. Due to lack of unconnected graph mining algorithms we compare our algorithm with two general techniques which make unconnected graph discovery possible by means of connected graph mining algorithms. We also perform undirected comparison of our algorithm with connected graph mining algorithms by comparing the number of discovered frequent subgraphs per second. Finally we derive a connected graph mining algorithm from our algorithm and show that it is competitive (though not winning) with popular connected graph mining algorithms.

Keywords: graph mining, unconnected frequent graphs.

1 Introduction

The problem of mining frequent graphs can be informally defined as follows: find all graphs which are subgraph isomorphic with large number of graphs in the given database. There are lots of classes of graphs which might be mined, e.g.: directed or undirected, labeled or unlabeled, simple or multi graphs, connected or disconnected [11]. In recent years, several subgraph mining algorithms were proposed including gSpan [12], MoFa [2], Gaston [10], FFSM [5], SPIN [6], FSG [9], AGM [7], AcGM [8]. Most of algorithms, however, discover only connected subgraphs. Among mentioned algorithms only AGM discovers unconnected graphs but is additionally limited to induced graphs only. We propose the algorithm UGM (Unconnected Graphs

Łukasz Skonieczny
Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: `L.Skonieczny@ii.pw.edu.pl`

Miner) to discover all frequent subgraphs (connected and unconnected), which contain at least one edge in graph components. In this paper we are considering labeled undirected graphs.

2 Basic Notions

We omit the known definitions of *graph*, *undirected labeled graph*, *subgraph*, *supergraph*, *graph and subgraph isomorphism*, *embedding*, *support* or *frequent graph* due to limited space. If necessary inspect, e.g, [11]. We introduce two new notions:

Definition 1 (edge descriptor). Edge descriptor *of the edge* $e = \{v_1, v_2\}$ *is a pair* $(\{lbl(v_1), lbl(v_2)\}, lbl(e))$, *where lbl(v) and lbl(e) are the label of vertice v and the label of edge e.*

Definition 2 (edge set). Edge set *of graph G, denoted ES(G) is a multiset of edge descriptors built from every edge of G. Two different graphs can have the same edge set. If G' is subgraph isomorphic to G then* $ES(G') \subseteq ES(G)$.

3 Mining for Frequent Subgraphs

Difficulties in efficient frequent graphs discovery come from the following facts:

1. Number of subgraphs of a graph is more than exponential with respect to the number of vertices of the graph. The mining algorithm should avoid generation of candidates which are not frequent.
2. Two different subgraphs of a graph might be isomorphic to each other and we are not interested in having both of them in a result as they represent the same structure. The mining algorithm should avoid generation of such duplicates.
3. Subgraph isomorphism problem is NP-complete. The mining algorithm should avoid direct subgraph isomorphism tests. Algorithm in Fig. 1 describes the simplest procedure for mining frequent graphs.

Such algorithm can be implemented in depth first search (dfs) or breadth first search (bfs) manner. Algorithm in Fig. 1 obviously suffers from all three mentioned reasons. To overcome duplicate candidates generation subgraph miners usually use some kind of a canonical form of a graph. The example of such a form is a DFSCode used in gSpan [12] and more examples are presented in [1]. Our algorithm does not use any kind of a canonical code. All generated graphs are stored in the set of isomorphic graphs which is navigated through edge sets. Experiments show that duplicate detection in such structure takes only a small percent of total execution time. The need for direct subgraph isomorphism tests is usually removed by keeping the embeddings list. Embedding is a mapping of subgraph vertices into supergraph vertices. There might be many embedding of one subgraph into a given graph. Embeddings can be easily updated during extension of the frequent graph. It is easy to tell if a candidate extended from a frequent graph is still supported by

---

Start with an empty graph and put it to list of candidates.

for all graphs in candidates list **do**

    Calculate support of a candidate by executing subgraph isomorphism test with every graph in D.

    if support of the candidate is greater than or equal to minSup **then**

        put it into result, generate all possible extensions of this candidate and put them to the candidates list, making sure they are not isomorphic to any other graphs in the candidate list or result list (or already processed candidates list).

    end if

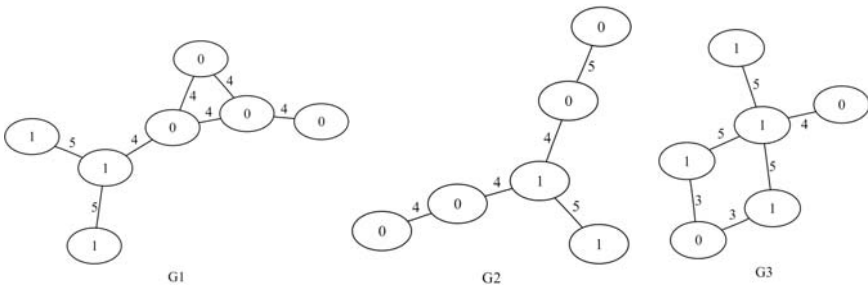**end for**

Fig. 1 Naive Graph Miner

Fig. 2 The example database of three graphs

graphs in the database. It just has to be checked if there are any embeddings which allow the extension. The UGM algorithm, to the contrary, performs direct subgraph isomorphism tests. Subgraph isomorphism testing (or subgraph matching) is actually finding the first embedding of the subgraph in supergraph or telling if there is no such embedding. We developed a new algorithm for subgraph matching which is based on solving Constraint Satisfaction Problem [4] and ideas similar to [3].

4 The UGM Algorithm

The UGM algorithm starts with removing all infrequent edges from the graphs in database. Then it generates *edge sets* (see Definition 2) for all graphs in the database. Next step is to mine all maximal frequent itemsets from edge sets. The result of this will be used later for pruning search tree, as graph is frequent only if its edge set is frequent which means it is a subset of some maximal frequent itemset. The edge sets of the graphs from Fig. 2 are presented below.

$ES(G_1) = \{(\{1,1\},5)(\{1,1\},5)(\{0,1\},4)(\{0,0\},4)(\{0,0\},4)(\{0,0\},4)(\{0,0\},4)\}$

$ES(G_2) = \{(\{0,0\},5)(\{1,1\},5)(\{0,1\},4)(\{0,1\},4)(\{0,0\},4)\}$

$ES(G_3) = \{(\{1,1\},5)(\{1,1\},5)(\{1,1\},5)(\{0,1\},4)(\{0,1\},3)(\{0,1\},3)\}$

Assuming minimal support value *minSup* = 2, the maximal frequent sets mined from these sets are:

$$ms_1 = \{((\{1,1\},5)(\{1,1\},5)(\{0,1\},4)\} \text{ and } ms_2 = \{((\{1,1\},5)(\{0,0\},4)(\{0,1\},4)\}$$

Having maximal frequent sets ot the edge sets the algorithm begins generating candidates starting with the empty graph in the depth first search manner. In the database from Fig. 2 there are three frequent edge descriptors: $a = (\{1,1\},5)$, $b = (\{0,1\},4)$, and $c = (\{0,0\},4)$. The maximal sets are then: $ms_1 = \{a,a,b\}$ and $ms_2 = \{a,b,c\}$. The algorithm will first try to extend empty graph with edge c as much as possible. It will stop after adding one c edge because $\{c,c\}$ is not frequent (because it is not subset of any set from maximal frequent sets). Then it will try to extend empty graph with b edge as much as possible, and then with c edge. The actual order is presented in Fig. 3. Notice, how maximal sets check reduces the search space. The path $\emptyset \rightarrow a \rightarrow b \rightarrow b \rightarrow \ldots$ will never be considered because $\{a,b,b\}$ is not frequent.The algorithm will add edges to the candidates the following way:

$$\emptyset \rightarrow c$$
$$\emptyset \rightarrow b \rightarrow c$$
$$\emptyset \rightarrow a \rightarrow c$$
$$\emptyset \rightarrow a \rightarrow b \rightarrow c$$
$$\emptyset \rightarrow a \rightarrow a \rightarrow b$$

This means that the empty graph will be extended in all possible ways (this means one way in this case) by edge c. The resulting graph will not be extended any more. Then the empty graph will be extended by edge b and the resulting graph will be extended in all possible ways (two in this case) by edge c. The resulting graphs will not be extended any more. The procedure for paths $\emptyset \rightarrow a \rightarrow c$, $\emptyset \rightarrow a \rightarrow b \rightarrow c$ and $\emptyset \rightarrow a \rightarrow a \rightarrow b$ is similar.
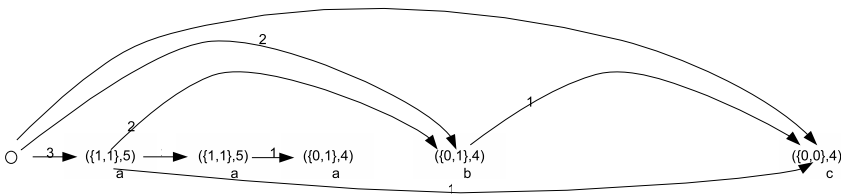

Fig. 3 Order in which edges are added to the candidate graphs

4.1 Graph Extension

The graph G is represented as a set of (CG,c) pairs where CG is a connected component[1] of graph G and c is the number of occurrences of CG in G. Graph $G = \{(CG_1,c_1),(CG_2,c_2),\ldots,(CG_n,c_n)\}$ can be extended by edge descriptor $ed = (\{lv_1,lv_2\},le)$ in three general ways:

---

[1] Connected component of a graph is a maximal connected subgraph of this graph.

1. Create new component. Precisely, add new separate edge and two new nodes. This may result in one new (CG, c) pair if the new one-edge component is not isomorphic to any existing components or in incrementing c in corresponding isomorphic component.
2. Modify existing component:

   - add new edge and one new node,
   - add new edge between existing nodes.

3. Add connections between components:

   - add connections between isomorphic components (that is within one (CG, c) pair) Possible only if $c \geq 2$,
   - add connections between non isomorphic components (that is between different (CG, c) pairs).

This procedure may result in duplicate candidates, so they must be eliminated by performing graph isomorphism tests.

4.2 Support Calculation

The support of each candidate is evaluated separately. A candidate is tested for being subgraph isomorphic with graphs from database which supported the parent of the candidate (lets call them D). If the candidate is subgraph isomorphic with the graph, then the graph supports candidate and candidates support is incremented. When the number of failed tests exceeds $|D| - minSup$ the loop is terminated as it is impossible that the candidate is frequent.

4.3 Candidates Pruning

There are cases when there is no need to calculate support of the candidate because we can before-hand tell that it is not frequent. If the candidate is a supergraph of the graph that was already found to be not frequent then it is also not frequent. That is why, the algorithm maintains so called *negative border* set, which is the structure that keeps all candidates that were found to be not frequent (except these found in this way). There is one more technique that prunes the search space. If all candidates generated from the given graph G by extension with edge descriptor ed are found to be infrequent, then the ed is put into set of *unsuccessful extensions* of graph G. Now, before extending some other graph G' with ed' the algorithm checks if G' is a supergraph of some graphs from *unsuccessful extensions* map, and if the corresponding set contains ed'. If it is found to be true, there is no need to generate candidates because they will all be not frequent. Please note, that *negative border* would also prune these candidates but this technique makes it earlier and faster.
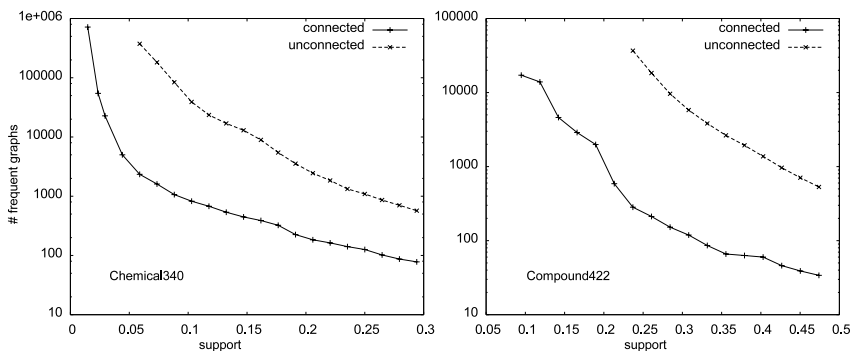
Fig. 4 Number of connected frequent graphs and all (connected and unconnected) frequent graphs for two datasets

5 Experiments

We implemented our algorithm as a part of ParMol[2] framework. Due to lack of unconnected graph miners we compare our algorithm with two general techniques which allow unconnected graph discovery by means of connected graph miners. First technique (called *UgmOnVirutalEdges*) is to add an artificial edges between every unconnected pairs of vertices in the input set of graphs, then perform typical frequent connected graphs discovery, and finally remove artificial edges from the resulting graphs. The other technique (*UgmOnConnected*) is to find all frequent connected graphs and join them pair wise creating unconnected candidates. Experiments show (Fig. 5) that both techniques are couple times of magnitude slower than *UGM* algorithm.
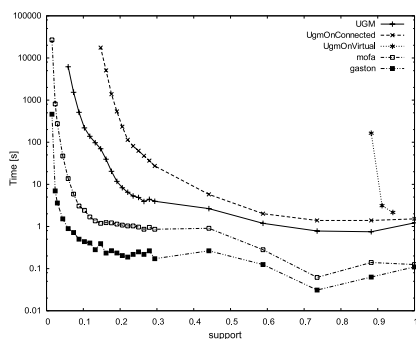

Fig. 5 Execution time of various graph miners for chemical340 graph set
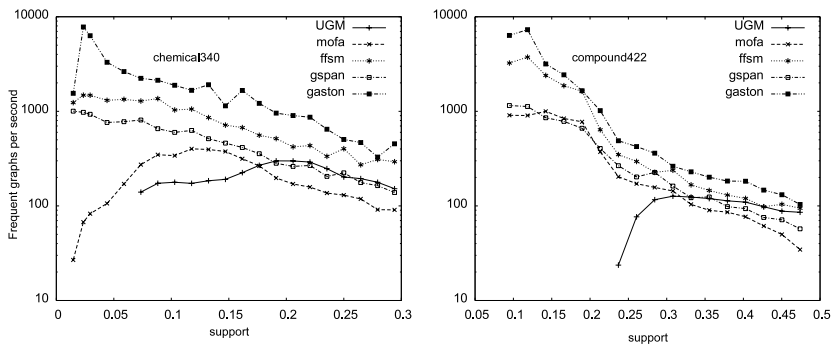
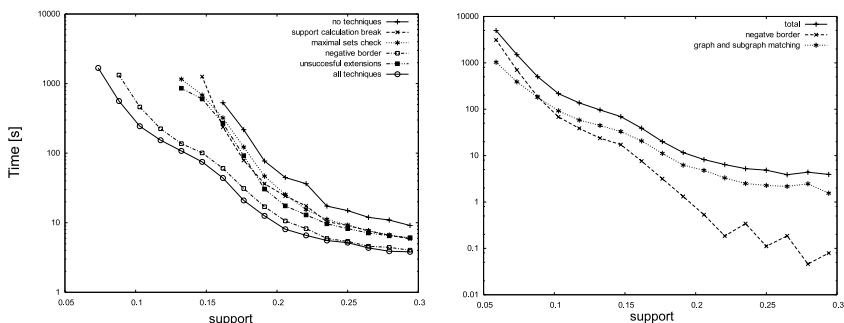**Fig. 6** Result generation speed. The number of frequent graphs found per second

Fig. 7 left Efficiency of optimizations. **right** Time of maintaining negative border, time of graph and subgraph isomorphism tests and total time
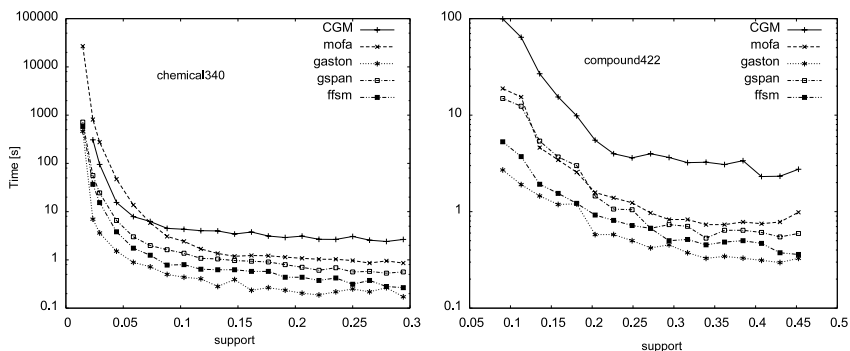

Fig. 8 Execution time of connected graph miners for two datasets

We also perform undirected comparison of our algorithm with connected graph miners by comparing number of discovered frequent subgraphs per second (Fig. 6). The *Gaston* algorithm is an unquestionable leader, though *UGM* is competitive with *mofa*.

Finally we derived a connected graph miner from our algorithm. This required simplifying graph extension procedure (omitting points 1 and 3) and abandoning proposed edge ordering. The *CGM* is competitive to other connected graph miners (especially *mofa*), though *Gaston* is significantly leading again (Fig. 8).

We tested our algorithms on three well known graph sets – Chemical340, Compound422, and Hiv-cacmci-99.

6 Conclusions

We have proposed a new algorithm called *UGM* for finding both connected and unconnected frequent graphs. Our approach is based on dfs candidate generation and direct execution of subgraph isomorphism test over database. We introduced several techniques that limit the search space of candidates. These are:

1. maximal edge sets check,
2. negative border,
3. unsuccessful extensions,
4. break in support calculation.

Figure 7 shows that all of them introduce great boost to the algorithm. Negative border looks the most promising but as one can see in Fig. 7 the cost of searching the negative border becomes the most time consuming part of the algorithm for low values of support (while normally subgraph matching is the most time consuming). It is therefore worth to make some research in effective storing of negative border.

References

1. Borgelt, C.: Canonical forms for frequent graph mining. In: Advances in Data Analysis, Studies in Classification, Data Analysis, and Knowledge Organization. Springer, Heidelberg (2007)
2. Borgelt, C., Berthold, M.R.: Mining molecular fragments: Finding relevant substructures of molecules. In: Proceedings of the IEEE International Conference on Data Mining, Washington, US (2002)
3. Dupont, P., Zampelli, S., Deville, Y.: Symmetry breaking in subgraph pattern matching. In: Proceedings of the 6th International Workshop on Symmetry in Constraint Satisfaction Problems (2006)
4. Frost, D.H.: Algorithms and heuristics for constraint satisfaction problems. Ph.D. thesis, University of California, Irvine, US (1997)
5. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraphs in the presence of isomorphism. In: Proceedings of the 3rd IEEE International Conference on Data Mining, pp. 549–552 (2003)
6. Huan, J., Wang, W., Prins, J., Yang, J.: Spin: mining maximal frequent subgraphs from graph databases. In: Proceedings of the 10th ACM International Conference on Knowledge Discovery and Data Mining, New York, US (2004)
7. Inokuchi, A., Washio, T., Motoda, H.: Complete mining of frequent patterns from graphs: Mining graph data. Machine Learning 50(3), 321–354 (2003)

8. Inokuchi, A., Washio, T., Nishimura, K., Motoda, H.: A fast algorithm for mining frequent connected subgraphs. Research Report RT0448 (2002)
9. Kuramochi, M., Karypis, G.: Frequent subgraph discovery. In: Proceedings of the IEEE International Conference on Data Mining, Washington, US (2001)
10. Nijssen, S., Kok, J.N.: A quickstart in frequent structure mining can make a difference. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, US, pp. 647–652 (2004)
11. Washio, T., Motoda, H.: State of the art of graph-based data mining. SIGKDD Explorations Newsletter 5(1), 59–68 (2003)
12. Yan, X., Han, J.: Gspan: graph-based substructure pattern mining. In: Proceedings of the IEEE International Conference on Data Mining, pp. 721–724 (2002)

Numerical Evaluation
of the Random Walk Search Algorithm

Arkadiusz Biernacki

Abstract. In the paper we propose numerical evaluation of Random Walk Search Algorithm (RWSA) which is one of the algorithms used for resource localization in peer-to-peer overlay networks. As a tool for the evaluation we chose Markov Chain. Each state of the chain represents a single random walker residing in particular network node. Using Markov Chain theory it is relatively simple to calculate basic statistical property of the RWSA for a single walker i.e. mean search time. In the analysis we included several scenarios using different network topology parameters and variable number of resource copies placed in network nodes. The calculated statistic may be useful in tuning the RWSA parameters to a given network topology.

Keywords: contend delivery networks, P2P networks, information retrieval.

1   Introduction

An overlay network is a layer of virtual network topology on top of the physical network (e.g., Internet), which directly interfaces to users. Overlay networks allow both networking developers and application users to easily design and implement their own communication environment and protocols on top of the Internet, such as data routing and file sharing management. Popular overlay networks include multicast overlays, peer-to-peer (P2P) overlays (e.g., Gnutella and Kazaa), parallel file downloading overlays (e.g., BitTorrent and eDonkey), routing overlays (e.g., Skype for VoIP). A P2P overlay network does not have the notion of clients or servers but only equal peer nodes that simultaneously function as both 'clients' and 'servers' to the other nodes on the network. Early P2P overlay networks systems, such as Napster, used a central server to store indices of resources stored in the overlay network.

Arkadiusz Biernacki
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
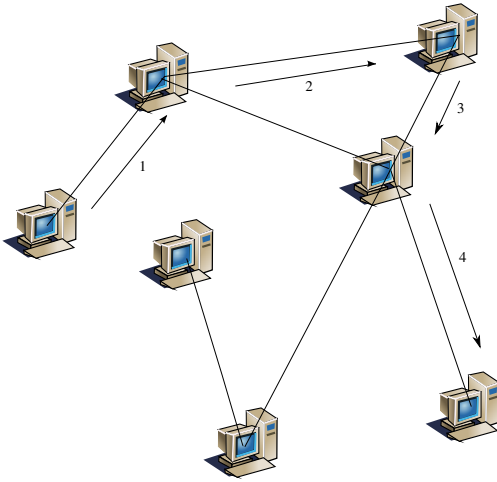e-mail: arkadiusz.biernacki@polsl.pl

Fig. 1 Random walk algorithm

This centralized design concerns of performance bottleneck and single point of failure. To avoid such possibility, instead of maintaining a huge index in a centralized system, a decentralized system distributes all searching and locating loads across all the participating network nodes. Though the decentralized approach concerns the overloading and reliability issues it is thought to build a more scalable P2P overlay system. A success of such decentralized system considerably depends on an efficient mechanism to broadcast queries (messages, packets) across a large population of network nodes. Because there are no coordinating nodes and a network topology is unknown addressing a node with a given attributes is a major concern in an unstructured network.

One of the algorithms eligible for the above task is a Random Walk Search Algorithm (RWSA). A node, called the querying node, that needs to locate a resource sends a search query (packet) to randomly selected neighbour. The search query is often referred as a random walker. Each random walker has a time to live (TTL) field that is initiated with some value $T > 0$ that limits the number of times the random walker is forwarded. When an intermediate node receives a random walker, it checks to see if it has the resource. If the intermediate node does not have the resource, it checks the TTL field, and if $T > 0$, it decrements T by 1 and forwards the query to a randomly chosen neighbour, else if $T = 0$ the query is not forwarded. The above algorithm is depicted in Fig. 1. RWSA is decentralized, and do not use any local information such as the identities and connectedness of their neighbours. It is used primarily in networks based on Gnutella protocol.

In this work we compute an expected number of steps (forwards) of a single random walker before it finds searched resource in overlay network. We call this value mean search time of RWSA and perform the analysis for several different parameters of network which topology is based on Waxman model [11]. For the

computation we use Markov Chain theory what is the primary contribution of our paper. We assume that TTL of a random walker is infinitive.

2 Previous Works

The reviews of search methods in unstructured networks are presented [6, 8, 10]. Generally search techniques in P2P overlay networks may be divided into three categories:

- Techniques based on Breadth-First-Search algorithm limited by query TTL parameter. The most popular in this category is flooding search algorithm. In the algorithm each node acts as both a transmitter and a receiver and each node tries to forward every message to every one of its neighbours except the source node. Hence the number of search queries grows exponentially with number of network nodes [7].
- Techniques based on Depth-First-Search algorithm without the returns [7]. The RWSA belongs to Depth-First-Search class.
- Hybrid techniques based on the two above mentioned techniques. The popular algorithms in this class amongst others have memory of previous searches and information about closest neighbours content [3, 9].

In depth analysis of RWSA is presented in [2]. For the algorithm analysis the authors used numerical analysis based on Coupon Collection, Chernoff Bounds and simulation. In [1] the performance analysis of RWSA is brought to solution of an equation with constraints. The constraints characterize the minimum acceptable performance of the search algorithm for a resource with a given popularity. In [12] authors evaluate the Random Pick Process and Coupon Collection model for RWSA. They state that these models are less accurate when the number of covered nodes becomes large and propose some refinements to the models.

3 Theoretical Background – Markov Chains

3.1 Definitions

Markov processes provide very flexible, powerful, and efficient means for the description and analysis of computer system properties and can be used to describe a system moving over time between different states [4]. A Markov Chain is a process that consists of a finite number of states and some known probabilities p_{ij}, where p_{ij} is the probability of moving from state j to state i. More formally:

Definition 1. *A Markov chain is a sequence of random variables $X_1, X_2, X_3, \ldots$ with the Markov property, namely that, given the present state, the future and past states are independent. Formally,*

$$\Pr(X_{n+1} = x | X_n = x_n, \ldots, X_1 = x_1) = \Pr(X_{n+1} = x | X_n = x_n).$$

The possible values of X_i form a countable set S called the state space of the chain.

If the state space S is finite Markov chains are may be described by a directed graph, where the edges are labeled by the probabilities of going from one state to the other states. The transition probability distribution can be represented by a matrix, called the transition matrix, with the (i, j)th element of P equal to

$$p_{ij} = \Pr(X_{n+1} = j | X_n = i).$$

A state s_i of a Markov chain is called absorbing if it is impossible to leave it (i.e., $p_{ii} = 1$). A Markov chain is absorbing if it has at least one absorbing state, and if from every state it is possible to go to an absorbing state (not necessarily in one step). In an absorbing Markov chain, a state which is not absorbing is called transient.

Theorem 1. *In an absorbing Markov chain, the probability that the process will be absorbed is 1 [4].*

3.2 Canonical Form of Markov Chain

The canonical form of transition matrix of an arbitrary absorbing Markov Chain with r absorbing states and t transient states is stated as

$$\begin{pmatrix} Q & R \\ 0 & I \end{pmatrix}.$$

Here I is an r-by-r indentity matrix, 0 is an r-by-t zero matrix, R is a nonzero t-by-r matrix, and Q is an t-by-t matrix. The first t states are transient and the last r states are absorbing.

Theorem 2. *Let t_i be the expected number of steps before the chain is absorbed, given that the chain starts in state s_i, and let t be the column vector whose ith entry is t_i. Then*

$$t = Nc, \tag{1}$$

where c is a column vector all of whose entries are 1 and $N = (I - Q)^{-1}$ [4].

3.3 Random Walk and Markov Chain

A random walk on a graph is a very special case of a Markov chain. Given an undirected, connected graph $G(V, E)$, where V are vertices and E are edges of the graph, a random 'step' in G is a move from some node u to a randomly selected neighbour v. A random walk is a sequence of these random steps starting from some initial node. The above process is discrete. We assume than our network topology graph is undirected, non-bipartite and not fully connected.

A RWSA on connected, undirected, non-bipartite graph G can be modeled as a Markov Chain, where the vertices V of the graph are represented by the states of the Markov Chain and the transition matrix is as follows:
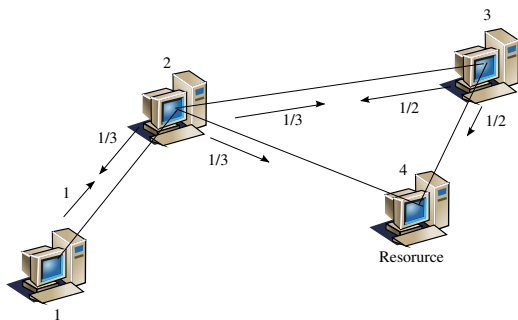
Fig. 2 Random walk graph

$$P(u,v) = \begin{cases} \frac{1}{d(u)} & \text{if } (u,v) \text{ vertices are connected,} \\ 0 & \text{if } (u,v) \text{ vertices are not connected,} \end{cases} \tag{2}$$

where $d(u)$ is a degree of vertex u. When a searched resource is placed in a given vertex we assume that random walker which encounters this vertex is absorbed. In Markov Chain theory any transition from the state representing that vertex to other states is impossible. Thus if we denote this state as a s_i then $p_{ii} = 1$.

The adjacency matrix for the network presented in Fig. 2 has form:

$$\begin{array}{c} (u,v) \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}. \tag{3}$$

Taking into account (2) a transition matrix for the Markov Chain based on the adjacency matrix from (3) has form:

$$\begin{array}{c} (u,v) \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 \\ 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}.$$

Note that states 1–3 are transient and state 4 is absorbing.

3.4 Network Topology

The random network topology generator introduced in [11] is a geographic model for the growth of a computer network. In this model the nodes of the network are uniformly distributed in a constrained plane and edges are added according to

probabilities that depend on the distances between the nodes. The probability to have an edge between nodes *u* and *v* is given by

$$P(u,v) = \alpha e^{-d/\beta L}, \tag{4}$$

where $\alpha > 0$, $\beta <= 1$, *d* is the distance from *u* to *v*, and *L* is the maximum distance between any two nodes. An increase in the parameter α increases the probability of edges between any nodes in the graph, while an increase in β leads to a larger ratio of long edges to short edges. An example network topology generated by the Waxman model is presented in Fig. 3.
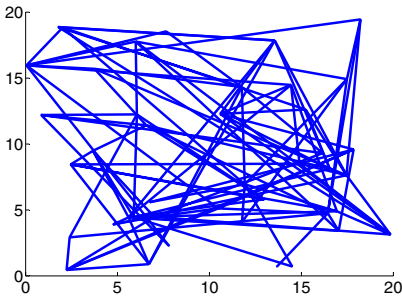

Fig. 3 Network topology based on Waxman model

4   Analysis and Its Results

For the topology generation we modified the Matlab code published on [5] which utilizes Waxman model (4). The number of nodes is Poisson distributed and proportional to the area of the domain. Hence, the nodes follow a Poisson process in the plane. We examined search time in the function of the following parameters:

- *replications* – number of copies of the resources which are to be searched. For our experiment we used a set consisting of the values: 1, 2, 4, 8, 16, and 32. When replication parameter is set to *n* it means there is a *n* instances of a resource in the network nodes,
- α – link probability between vertices of the graph, interpretation as in (4),
- β – parameter to control length of the edges, interpretation as in (4).
- λ – intensity of the Poisson process. Increased λ yields a higher number of generated vertices,
- *size* – size of the network limited by bounds for the square region in which nodes are placed. The higher the size the more vertices in the graph.

As a result of the code execution for the different sets of above parameters we obtained several graph adjacency matrices. These matrices were converted to Markov Chain transition matrices according to (2) and then the mean search time was calculated using (1).

In Fig. 4 we presented the influence of the *replications*, α, β, λ, and *size* parameters on search performance. What is obvious for all four analysis that with the increasing number of replication of searched resource mean search time drops (Fig. 4a). One may observe that with increase of link probability between vertices of the graph the search time gets a little higher. Probably due to more edges in the graph a random walker encounters more cycles but this impact is very subtle. The length of the edges of graph do not seem to have an much influence on search time (Fig. 4b). With the increase of networks vertices density and network size the search efficiency drops Fig. 4c and Fig. 4d.
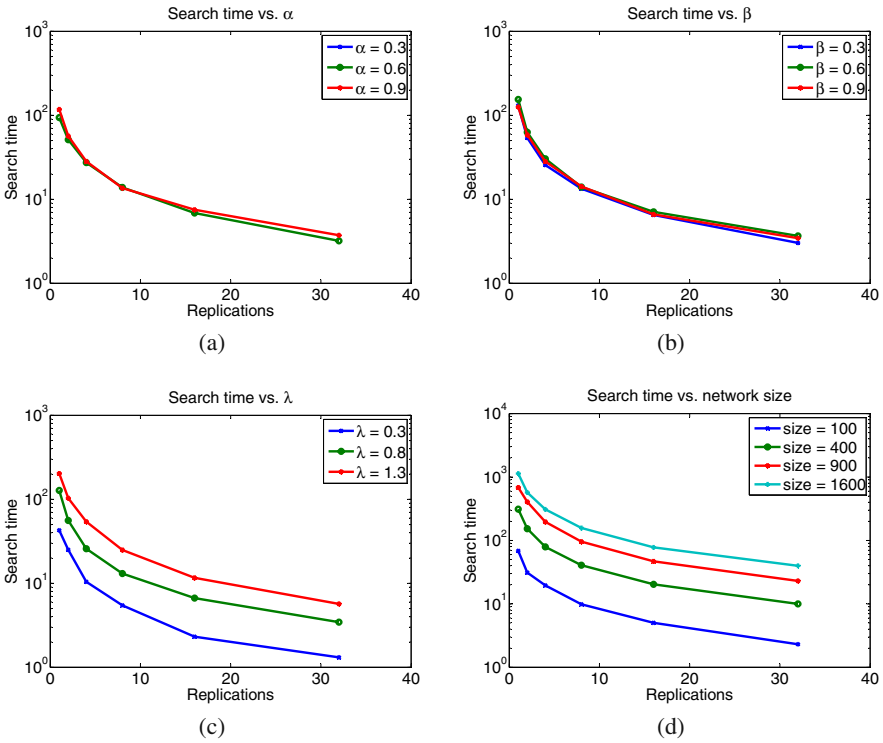
Fig. 4 **(a)** Search time vs. link probability between edges α. **(b)** Search time vs. length of the edges β. **(c)** Search time vs. edges density λ. **(d)** Search time vs. network area size

5 Conclusions

In the paper we proposed numerical evaluation of Random Walk Search Algorithm. As a tool for the evaluation we chose Markov Chain. We showed that using Markov chain theory it is relatively simple to calculate basic statistical property of the algorithm for a single walker, i.e., mean search time. In the analysis we included several scenarios using different network topology parameters and variable number

of copies of a resource placed in network nodes. The calculated statistic may be useful in RWSA parameters to a given network topology.

Further works may include analysis of RWSA with nonuniform forwarding probabilities (adaptive RWSA). Also it is possible to compare the performance of the algorithm when applied to different network topology models.

References

1. Bisnik, N., Abouzeid, A.: Modeling and analysis of random walk search algorithms in P2P networks. In: Proceedings of the 2nd International Workshop on Hot Topics in Peer-to-Peer Systems, pp. 95–103. IEEE Computer Society, Washington (2005)
2. Gkantsidis, C., Mihail, M., Saberi, A.: Random walks in peer-to-peer networks. In: Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 1 (2004)
3. Gkantsidis, C., Mihail, M., Saberi, A.: Hybrid search schemes for unstructured peer-to-peer networks. In: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3 (2005)
4. Grinstead, C.M., Snell, J.L.: Introduction to Probability. American Mathematical Society (1997)
5. Kaj, I., Gaigalas, R.: Waxman topology generator (2005), http://www.math.uu.se
6. Kalogeraki, V.: A local search mechanism for Peer-to-Peer networks. In: Proceedings of the 11th International Conference on Information and Knowledge Management, p. 300. Association for Computing Machinery (2002)
7. Knuth, D.E.: The Art Of Computer Programming, 3rd edn., vol. 1. Addison-Wesley, Boston (1997)
8. Risson, J., Moors, T.: Survey of research towards robust peer-to-peer networks: Search methods. Computer Networks 50(17), 3485–3521 (2006)
9. Tsoumakos, D., Roussopoulos, N.: Adaptive probabilistic search for peer-to-peer networks. In: Proceedings of the 3rd International Conference on Peer-to-Peer Computing, pp. 102–109 (2003)
10. Tsoumakos, D., Roussopoulos, N.: A comparison of Peer-to-Peer search methods. In: WebDB (2003)
11. Waxman, B.M.: Routing of multipoint connections. IEEE Journal on Selected Areas in Communications 6(9), 1617–1622 (1988)
12. Wu, B., Kshemkalyani, A.D.: Modeling message propagation in random graph networks. Computer Communications (2008)

On Two Variants
of the Longest Increasing Subsequence Problem

Sebastian Deorowicz and Szymon Grabowski

Abstract. Finding a longest increasing subsequence (LIS) of a given sequence is a classic problem in string matching, with applications mostly in computational biology. Recently, many variations of this problem have been introduced. We present new algorithms for two such problems: the longest increasing circular subsequence (LICS) and the slope-constrained longest increasing subsequence (SLIS). For LICS, our algorithm improves one of the most competitive techniques if the length of the output sequence is close to its expected value $2\sqrt{n} + o(\sqrt{n})$. In the algorithm for SLIS, we show how to gain from modern successor search data structures, which is not trivial for this task.

Keywords: longest increasing subsequence problem, sequence alignment.

1 Introduction

Finding a longest increasing subsequence (LIS) of a given sequence is a classic problem in string matching, with applications mostly in computational biology [9]. Given a sequence S of symbols over an integer alphabet (w.l.o.g. we can assume they are unique) we need to find a longest increasing subsequence (a subsequence is obtained by removing zero or more elements) of S. A simple dynamic programming algorithm with $O(n \log n)$ time complexity is optimal in the comparison model [8]. In the RAM model achieving $O(n \log \log n)$ worst-case time is possible [10]. In the

Sebastian Deorowicz
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: sebastian.deorowicz@polsl.pl

Szymon Grabowski
Technical University of Łódź,
Żeromskiego 116, 90-924 Łódź, Poland
e-mail: sgrabow@kis.p.lodz.pl

general case of arbitrary symbols, the sequence S must be first remapped onto the range $[1, n]$ in $O(n \log n)$ time.

In recent years several variants of this problem have been posed. In this paper we deal with two such variants: longest increasing circular subsequence (LICS) [1] and slope-constrained longest increasing subsequence (SLIS) [13]. In LICS the task is to find a longest subsequence of any *rotation* of a given sequence such that each integer of the subsequence is smaller than the integer that follows it. In the SLIS problem, the input data include also a non-negative slope boundary m, which is a real number. In [13] all that is assumed about the input sequence S is that its elements are comparable. The task is to find a maximum-length increasing subsequence of S, $s_{i_1} < s_{i_2} < \cdots < s_{i_k}$ for $i_1 < i_2 < \cdots < i_k$, such that the slope between two consecutive points is no less than the input ratio, i.e., $\frac{s_{i_{r+1}} - s_{i_r}}{i_{r+1} - i_r} \geq m, 1 \leq r < k$.

2 Slope-Constrained Longest Increasing Subsequence

2.1 The Yang and Chen Algorithm

The slope-constrained longest increasing subsequence problem was posed by Yang and Chen in [13], where an algorithm running in $O(n \log \ell)$ time and $O(n)$ space, where ℓ is the length of the obtained sequence, was also given. We are not aware of any other results for SLIS. In our algorithm we use the concept of so-called critical points introduced by Yang and Chen, hence we start with presenting the original technique (see [13, Fig. 2]) for pseudocode).

We follow their notation $s_i \prec_S s_j$, which denotes that $\frac{s_j - s_i}{j - i} \geq m$ for any $j > i$, and m is the given slope. We also say that s_i dominates s_j if $s_i \prec_S s_j$. It is easy to notice that the relation $\prec_S$ is transitive. The *rank* of element s_i is defined as the length of an SLIS (note the article 'n', instead of 'the', since the desired longest sequence may not be unique) ending with s_i. Each s_i has a positive rank, since an SLIS may start with an arbitrary element. The algorithm inspects the input sequence element by element, calculating their ranks. The whole sequence is partitioned into disjoint groups $R_1, R_2, \ldots, R_k$, according to their rank value. It is shown that in order to find out, for the currently processed element s_i, if it has any ancestor of rank k, it is enough to check only the last inserted (rightmost) elements of all ranks up to k, i.e., compare s_i against a subset of S of size k. The value k is obviously limited by the output size ℓ. The rightmost elements of rank k are called *critical points* (actually the cited authors use another definition of critical points and then prove the equivalence of both, but this discussion is irrelevant for our exposition). From the transitivity property, it implies that if any $s \in R_k$ at ith iteration dominates s_i, then all critical points in R_j, $j < k$, dominate s_i. The final conclusion is that binary search among the critical points can be performed at each iteration, to find the maximum rank r among those that dominate s_i, and set the rank of s_i to $r + 1$, or set it to 1 if s_i has no predecessor. The backtracking links are maintained, and the space for them and all

the critical points is $O(n)$ in the worst case. The time complexity of the algorithm, $O(n \log \ell)$, implies from the binary search applied to each of n input elements.

2.2   Our Algorithm

Assume that the universe for S is $[1, \ldots, n^c]$, where $c \geq 0$ is a constant and n^c is an integer, hence the elements of S lie on a grid. Assume also the word RAM model of computation, and the machine word has $w \geq c \log n$ bits (all logarithms in this paper are binary). Note that if the machine word is not large enough (i.e., if constant c is too large), we can simulate all single operations on machine words of size $c \log n$ with a constant number of operations on 'real' words, of size not less than $\log n$ bits, hence the complexities are not affected. We will replace binary search with a modern data structure for successor[1] search, working efficiently with integer keys if the universe is bounded. The following lemma contains an observation central to our algorithm.

Lemma 1. *For any algorithm iteration i, the ascendingly sorted projections of all critical points onto the vertical axis, where these projections use lines of slope m passing through the given points, correspond to the increasing ranks of the used critical points.*

Proof (by contradiction). Let the described projection of the critical point c_k of a rank k be p_k. Assume that the lemma does not hold. Therefore there exists a critical point c_{k-1} of rank $k-1$ whose projection p_{k-1} is greater than p_k. Note that $k > 1$. There are two cases: the index of c_{k-1} is greater than the index of c_k, or not. In the first case, the slope of the line segment connecting c_k and c_{k-1} is greater than m, hence c_{k-1} could be a successor of c_k and then its rank is greater than k, contradiction. In the second case, there cannot be a link from c_{k-1} to c_k, since the slope of the line segment connecting c_{k-1} and c_k is less than m. Consider now any slope-constrained increasing sequence of length k ending with c_k and its last link, from some s_i to c_k; from the previous sentence we know that s_i must be a different point than c_{k-1}. The rank of s_i is $k-1$, hence its index must be less than the index of c_{k-1}, by definition of a critical point. The point s_i lies either on the line of slope m passing through c_k, or below it. On the other hand, the point c_{k-1} lies above that line, from our assumption. This implies that s_i dominates c_{k-1} and hence must be of rank at most $k-2$. This contradiction ends the proof.   □

From the lemma above, it is clear that we can translate the original task for point s_i, i.e., finding the critical point with the maximum rank among those that dominate s_i, into finding the critical point with the largest projection onto the vertical axis which is not greater than the projection of s_i. Still, we do not need to record the rank of the key inserted into the structure, as it is enough to maintain a counter of stored elements. To update the structure, we must insert the key and remove its successor

---

[1] In literature, the term 'predecessor search' is more often used, but those are twin problems with no difference in used techniques and complexities.

(with strictly greater key value) if it exists. The problem now is how to represent the projections so as to enable efficient successor searches.

We assume that the minimum slope $m \geq 0$ is a number that we can operate with in constant time in a real computer, e.g., a rational number. However, we want to put more constraint onto the slope parameter we use. First, note that m is upper-bounded by $n^c - 1$, as any larger slope makes the problem trivial, with the output sequence of length 1. Note also it is legal to replace m with $m' \geq m$ as long as any point on the considered grid that lied on or above the line segment of slope m from the previous point is also 'accepted' using the slope m'. With those two basic observations in mind, there are only $O(n^{c+1})$ slopes to consider (in total, over all possible input parameters m), they are defined by the lines passing $(1, 1)$ and all other points on the grid, and each slope is a rational number, p/q, where $0 \leq p < n^c - 1$ and $0 < q < n$. Our goal is to minimize $p/q - m$, as long as $p/q \geq m$. Note that the expression $p/q - m$ is evaluated in constant time.

We will show how to find the best approximation of m in $O(n)$ time. The slopes to consider for a given m correspond to lines passing $(1, 1)$ and $(q + 1, p + 1) = (i, 1 + \lceil (i - 1)m \rceil)$ for all $i \in \{2, \ldots, n\}$. As there are $O(n)$ of them and evaluating $p/q - m$ takes constant time, we find the minimum (and hence m' used later) in promised time.

Now, the projection $P(i)$ of a point (i, s_i) onto the vertical axis is $s_i - ip/q$, where $p/q = m'$. The two extreme possible points are $(n, 1)$ and $(1, n^c)$, with projections $1 - np/q$ and $n^c - p/q$, respectively. We can rescale all projections multiplying them by q, to obtain the interval $[q - np, n^c q - p]$, and s_i mapped into $s_i q - ip$. Note that all integers in the obtained interval are within a range of $O(n^{c+1})$ size.

We have obtained an integer domain of $n^{O(1)}$ size, with at most ℓ values in it at any time (those corresponding to critical points), and the task is to perform n successor searches in it. We need an efficient dynamic structure for successor searches, with worst-case time guarantees for the operations *succ*, *insert* and *delete*, working in $O(n)$ space.

Note that several critical points may have equal projections. We require distinct keys in the structure, therefore our keys are of the form $P'(i) = P(i) << \log n + i$, where $<<$ denotes bitwise shift-left operation (for notation clarity, we assume that $\log n$ is an integer). The extra $\log n$ bits in a key do not affect the complexities. For the inserted key k, its successor is found and deleted. If k has no successor, the maximum rank counter is incremented by 1. After processing all n input points this counter stores the SLIS length. Backtracking links, to restore the sequence, are maintained just like in [13], adding only $O(\ell)$ time.

What are competitive dynamic data structures to handle successor queries in our scenario? We can use exponential search trees by Andersson and Thorup [2], which allow for $O(\sqrt{\log \ell / \log \log \ell})$ worst-case time successor queries, inserts and deletes to a dynamic set of ℓ integer keys in linear space. Another option is to use the work of Beame and Fich [3], to obtain $O(\log \log \ell \times \log \log U / \log \log \log U)$ worst-case time operations, where U is the universe size, which translates in our case to $O(\log \log \ell \times \log \log n / \log \log \log n)$ time complexities. Finally, there is a way to use the classic van Emde Boas (vEB) tree [7], with $O(\log \log U)$ worst-case time operations. To

this end, we cannot use the vEB directly, since either its space utilization would be $O(U)$, i.e., prohibitive in our case, or using standard ideas to reduce its space to linear in the number of elements would lead to randomization and we want to keep worst-case time bounds. Since we are not interested in actual values of our n keys, only in their relative order, we can first sort them and then remap trivially to the range $[1, n]$. Because $U = n^{c+1} = n^{O(1)}$, the textbook radix sort is enough to order the keys in $O(nc)$ time. For a large c, a better choice may be the algorithm from [11] with $O(n \log c)$ worst-case time.

It is easy to notice that using the technique by Beame and Fich can never lead to lower complexity than both other options. Still, we must have a criterion to select the better of the two possible techniques, which depends on the ratio between n and ℓ, and the problem is that we cannot estimate ℓ quickly. The solution is fortunately simple: we start the algorithm using the Andersson and Thorup structure, and run it as long as the length of the resulting sequence is small enough (i.e., $\sqrt{\log \ell / \log \log \ell}$ is not greater than $\log \log n$); once it gets too large (it may or may not happen for a given problem instance), we terminate the procedure and start again from scratch, this time using the vEB structure. Overall, we obtain an $O(n)$ space algorithm for calculating an SLIS of $O(n \min(\sqrt{\log \ell / \log \log \ell}, \log \log n))$ worst-case time complexity.

3 Longest Increasing Circular Subsequence

3.1 Background

The problem of finding a longest increasing circuital subsequence (LICS) [1] is to find a longest increasing subsequence among all rotations of a given sequence. There is no single algorithm outperforming the other, since which algorithm is the fastest in theory depends on the data. Chen et al. [5] presented a technique, which applied for LICS gives an algorithm working in $O(n\ell)$ time. Tiskin [12] gave a $O(n^{3/2})$-time solution. Recently, Deorowicz [6] proposed a hybrid algorithm based on the idea of cover representation of sequence and effective cover merging techniques of complexity $O(\min(n\ell, n \log n + \ell^3 \log n))$.

In this paper, we focus our attention at the algorithm from [6], so we present some of its details. In general, an LICS is a longest LIS among all n rotations of sequence S, but in [6] it is shown that it suffice to restrict to only some rotations ending at so-called *stop points*. Since the number of stop points is $O(\ell)$, the task becomes much easier. Sequence S is represented in the algorithm as a *greedy cover*, which contains all the symbols from S grouped in a sequence of decreasing lists. The properties of the greedy cover are:

- the position of the list which each symbol belongs to is the length of an LIS ending at this symbol,
- it is unique for a given input sequence,
- its size is the length of an LIS of S.

CoverMerging(C, C')

01	**for** $i \leftarrow 1$ **to** $|C|$ **do**
02	**if** $C[i]$ head is larger than $C'[|C'|]$ tail **then**
03	Add to C' an empty list
04	$j \leftarrow |C'|$
05	**while** $C[i]$ is not empty **and** $j > 1$ **do**
06	Find the largest symbol p of $C[i]$ smaller than $C'[j-1]$ tail
07	Move the symbols larger than p from $C''[i]$ to $C'[j]$
08	$j \leftarrow j-1$
09	Append the rest (if any) of $C[i]$ to $C'[1]$

Fig. 1 A general scheme of the cover merging procedure. ($C[i]$ means ith list of cover C. If there is no such a symbol p in line 06, no symbols are moved in line 07)

The algorithm starts from the cover C of S and virtually rotates S by one symbol at a time. A rotated symbol is removed form the actual cover, so C contains only the symbols not rotated. Each time the symbol to rotate is a stop point, the algorithm stops, and say the actual sequence is $s_{k+1} \ldots s_n s_1 \ldots s_k$. At this moment, C stores the cover of only $s_1 \ldots s_k$. Now a cover C' is computed from the rotated symbols $s_{k+1} \ldots s_n$. Then, a *cover merging* is performed to compute the cover for the rotated sequence by merging C' and C (Fig. 1). We repeat this process until we meet all the stop points. Finally, when the sizes of covers for all rotations ending at stop points are known, the maximum of them is the LICS length.

A central part of the algorithm is the way of fast merging two covers. In [6] two approaches are presented. Both detach the first list of C and split it into $O(\min(\ell, m_i))$ parts, where m_i is the number of symbols rotated since ith stop point ($\sum_{i=1}^{O(\ell)} m_i = n$). The split points are the symbols ending the lists of C'. The parts are now attached to the ends of lists of C' (after this C' size may increase by 1). This procedure is repeated as long as C is not empty. Then, C' stores the cover of the rotated sequence.

The time complexity of a single, ith, cover merging is

$$T_i = O\left(\sum_{j=1}^{O(\ell)} t'_j + \min(\ell, m_i)\ell(t'' + t''')\right),$$

where:

- t'_j is time to find the split points of the jth list of C containing c_j elements,
- t'' is time to split a list into two parts,
- t''' is time to merge two lists.

Cover C stores less than n elements, so $\sum_{j=1}^{O(\ell)} c'_j < n$. We maintain the cover as an ordered collection of sorted lists. In the first implementation, those are linked lists and then

$$t'_j = O(\min(\ell, m_i) + c_j), \qquad t'' = O(1), \qquad t''' = O(1),$$

so $T_i = O(n + \ell \min(\ell, m_i))$. Summing it over all ($O(\ell)$) cover merges (for all stop points) we obtain $T = \sum_{i=1}^{O(\ell)} T_i = O(n\ell)$.

In the second implementation, the sorted lists composing the cover are actually red-black trees. In this case:

$$t'_j = O(\min(\ell, m_i) \log n), \qquad t'' = O(\log n), \qquad t''' = O(\log n).$$

Therefore, $T_i = O(\min(\ell, m_i)\ell \log n)$. Summing over all stop points we obtain $T = O(\min(\ell^2, n)\ell \log n) = O(\min(\ell^3, n\ell) \log n)$.

3.2 Our Algorithm

The hybrid algorithm in [6] achieves $O(\min(n\ell, n \log n + \ell^3 \log n))$ complexity, by choosing the way of representing the covers after computing the initial cover, when ℓ can be bound. In this section, we show a data structure that can be used to obtain a non-hybrid algorithm of slightly better complexity.

We postulate to represent each ordered list of a cover as an ordered list of red-black (RB) trees of size $\Theta(p)$ (to be determined later) and to store for each RB-tree its minimum (to be accessible in $O(1)$ time). In this case:

$$t'_j = \min(\ell, m_i) \log p + \Theta(c_j/p), \qquad t'' = O(\log p), \qquad t''' = O(\log p).$$

This leads to:

$$T_i = O(\ell \min(\ell, m_i) \log p) + O(n/p) + \min(\ell, m_i)\ell\, O(\log p), \text{ so}$$
$$T = O(\ell \min(\ell^2, n) \log p) + O(n\ell/p) + \min(\ell^2, n)\ell\, O(\log p) =$$
$$= O(\ell \min(\ell^2, n) \log p) + O(n\ell/p).$$

It can be easily found that T is minimized for $p = \Theta(\lceil n/\ell^2 \rceil)$, and then

$$T = O(\ell \min(\ell^2, n) \log\lceil n/\ell^2 \rceil + n\ell/\lceil n/\ell^2 \rceil) = O(\min(\ell^3, n\ell) \log\lceil n/\ell^2 \rceil).$$

The last thing to consider is the question if the red-black trees can be effectively maintained, i.e., if it is possible to guarantee using few fixup operations that at any time each such tree is of size $\Theta(p)$ even if the trees are split or joined. We require that the size of each RB-tree (except for the only one in a list) is in range $[p, 2p]$. Each time an RB-tree is split (in time $O(\log p)$), the sizes of both derivatives are determined to fit in the range and if necessary, they are *normalized*. If some (possibly both) has size less than p, it is merged with its neighbor (in time $O(\log p)$ [4, Sect. 3.11]). The merged tree is of size in range $[p, 3p]$. If it is larger than $2p$, we split it to two trees of size in range $[p, 1.5p]$. The trees are similarly normalized after joining lists.

For completeness, we should also remember the cost of an initial cover build, which is $O(n \log n)$ in the RB-tree based method. This can be reduced as follows.

Firstly, we always build cover consisting of real lists (not trees) Then, the cost of building RB-trees from sorted lists is linear.

The time complexity of creating each ith cover C' is $O(m_i \log \ell)$, which sums over all stop points to $O(n \log \ell)$. The last thing influencing the complexity is removing symbols from cover C during rotations. Each symbol is removed only once, so the total cost for all of them is $O(n \log \lceil n/\ell^2 \rceil)$. The cost of possible normalizations of trees is the same.

Therefore, the total complexity of our algorithm with lists of RB-trees is:

$$O(\min(\ell^3, n\ell) \log \lceil n/\ell^2 \rceil + n \log \ell).$$

4 Conclusions

We have considered two problems involving finding a longest increasing subsequence in the given input. In the LICS problem, any rotation of the input is allowed, while in the SLIS problem each pair of successive points in the solution must form a steep enough line segment. For both problems we assumed the RAM model of computation. For LICS, we improved the algorithm from [6] to achieve $O(\min(\ell^3, n\ell) \log \lceil n/\ell^2 \rceil + n \log \ell)$ time, while for SLIS our solution yields $O(n \min(\sqrt{\log \ell / \log \log \ell}, \log \log n))$ time, both complexities are for the worst case.

References

1. Albert, M.H., Atkinson, M.D., Nussbaum, D., Sack, J.R., Santoro, N.: On the longest increasing subsequence of a circular list. Information Processing Letters 101, 55–59 (2007)
2. Andersson, A., Thorup, M.: Dynamic ordered sets with exponential search trees. Journal of the ACM 54(3), Article No. 13 (2007)
3. Beame, P., Fich, F.E.: Optimal bounds for the predecessor problem and related problems. Journal of Computer and System Sciences 65(1), 38–72 (2002)
4. Brass, P.: Advanced Data Structures. Cambridge University Press, Cambridge (2008)
5. Chen, E., Yang, L., Yuan, H.: Longest increasing subsequences in windows based on canonical antichain partition. Theory of Computer Science 378(3), 223–236 (2007)
6. Deorowicz, S.: An algorithm for solving the longest increasing circular subsequence problem. Information Processing Letters 109(12), 630–634 (2009)
7. van Emde Boas, P., Kaas, R., Zijlstra, E.: Preserving order in a forest in less than logarithmic time and linear space. Information Processing Letters 6(3), 80–82 (1977)
8. Fredman, M.L.: On computing the length of longest increasing subsequences. Discrete Mathematics 11, 29–35 (1975)
9. Gusfield, D.: Algorithms on strings, trees, and sequences. Cambridge University Press, Cambridge (1999)

10. Hunt, J.W., Szymanski, T.G.: A fast algorithm for computing longest common subsequences. Communications of the ACM 20(5), 350–353 (1977)
11. Kirkpatrick, D.G., Reisch, G.: Upper bounds for sorting integers on random access machines. Theoretical Computer Science 28, 263–276 (1984)
12. Tiskin, A.: Semi-local string comparison: Algorithmic techniques and applications. Mathematics in Computer Science 1(4), 571–603 (2008)
13. Yang, I.H., Chen, Y.C.: Fast algorithms for the constrained longest increasing subsequence problems. In: Proceedings of the 25th Workshop on Combinatorial Mathematics and Computing Theory, pp. 226–231. Hsinchu Hsien, Taiwan (2008)

Computing the Longest Common Transposition-Invariant Subsequence with GPU

Sebastian Deorowicz

Abstract. Finding a longest common transposition-invariant subsequence (LCTS) of two given integer sequences $A = a_1 a_2 \ldots a_m$ and $B = b_1 b_2 \ldots b_n$ (a generalization of the well-known longest common subsequence problem (LCS)) has arisen in the field of music information retrieval. In the LCTS problem, we look for an LCS for the sequences $A + t = (a_1 + t)(a_2 + t) \ldots (a_m + t)$ and B where t is any integer. Performance of the top graphical processing units (GPUs) outgrew the performance of the top CPUs a few years ago and there is a surge of interest in recent years in using GPUs for general processing. We propose and evaluate a bit-parallel algorithm solving the LCTS problem on a GPU.

Keywords: general processing on GPU, longest common subsequence problem, sequence alignment.

1 Introduction

A longest common subsequence (LCS) problem for two sequences A and B is to find a longest subsequence (a subsequence is obtained from a sequence by removing zero or more symbols) of both sequences. There are numerous algorithms solving it [1]. A transposition-invariant LCS (LCTS) is its generalization, which appeared recently in the music information retrieval field [2, 8, 9]. In the LCTS problem all the values in one of the integer sequences are allowed to be shifted by any amount.

One of the fastest algorithms for an LCTS computing is a bit-parallel method, which make use of the fact that neighbor cells in a dynamic programming matrix defined by a recurrence of the classical solution differ by 0 or 1. Such values occupy

Sebastian Deorowicz

Institute of Informatics, Silesian University of Technology,

Akademicka 16, 44-100 Gliwice, Poland

e-mail: `sebastian.deorowicz@polsl.pl`

one bit and it is possible to process information of many neighbor cells at a time using all 32 or 64 computer word bits.

The performance of CPUs are steadily growing according to Moore's law from over 30 years. Nevertheless, the performance of graphical processor units (GPUs) is growing even faster and the top AMD and NVidia GPUs outperform the top CPUs by about an order of magnitude [12]. This is caused mainly by a highly parallel architecture of GPUs (contemporary GPUs consist of hundreds of cores). A few years ago using the GPU performance for general processing was problematic, since most of the GPU tools were designed for graphic processing. Nowadays the C-compliant CUDA library by NVidia [12] and OpenCL language by AMD [13] facilitate using the GPU computer power in C programming language.[1] A catalog of research and commercial applications of CUDA can be found in [11].

The paper is organized as follows. Section 2 defines the problem and gives some background. In Sect. 3, a bit-parallel algorithm for the LCTS problem (our a point of departure) is described. The GPU computing model (CUDA) is presented in Sect. 4. Then, our algorithm is introduced (Sect. 5) and examined in practice (Sect. 6). The last section concludes the paper.

2 Definitions and Background

The sequences $A = a_1 a_2 \ldots a_m$ and $B = b_1 b_2 \ldots b_n$ are over Σ, where $\Sigma \subset \mathbb{Z}$, called an *alphabet*, is a finite subset of integers. For simplicity it is assumed that $\Sigma = \{0, \ldots, \sigma - 1\}$. A *subsequence* is obtained from a sequence by deleting zero or more symbols. A sequence is a *longest common subsequence* (LCS) of A and B when it is a subsequence of both A and B of largest possible length. (There may be many LCSs since only LCS length is unique.) A *transposed copy* of sequence A, denoted as $A + t$, for some $t \in \mathbb{Z}$ is $(a_1 + t)(a_2 + t) \ldots (a_m + t)$. A *longest common transposition-invariant subsequence* (LCTS) of A and B is an LCS of B and $A + t$ for any $t \in (-\sigma, \sigma)$. We assume, w.l.o.g., that $m \leq n$.

A classical recurrence for an LCTS computation is (for all valid i, j, t):

$$M(i,j,t) = \begin{cases} M(i-1,j-1,t)+1 & \text{if } a_i+t = b_j \wedge i, j > 0, \\ \max(M(i-1,j,t), M(i,j-1,t)) & \text{if } a_i+t \neq b_j \wedge i, j > 0, \\ 0 & \text{if } i = 0 \vee j = 0. \end{cases} \quad (1)$$

These are $2\sigma - 1$ independent recurrences (for all possible $t \in (-\sigma, \sigma)$). An example of matrix M for $t = 0$ is given in the right part of Fig. 1.

A number of algorithms solving the LCTS problem were proposed (e.g., [3, 4, 7, 10]). The fastest, both in theory and practice, are those by Deorowicz [3] of time

---

[1] The architectures of NVidia and AMD GPUs are similar, so the discussion on general processing on the GPU is mainly true for both dominating vendors. For a sake of implementation we chose, however, NVidia's CUDA library and the technical details, e.g., number of cores in a multiprocessor are given for NVidia's products. For brevity we, however, write the GPU instead of NVidia GPU.
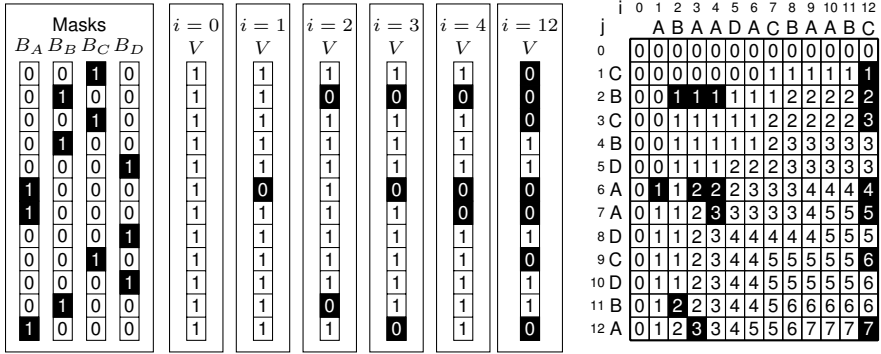
Masks

B_A	B_B	B_C	B_D
0	0	1	0
0	1	0	0
0	0	1	0
0	1	0	0
0	0	0	1
1	0	0	0
1	0	0	0
0	0	0	1
0	0	1	0
0	0	0	1
0	1	0	0
1	0	0	0

$i=0$ V	$i=1$ V	$i=2$ V	$i=3$ V	$i=4$ V	$i=12$ V
1	1	1	1	1	0
1	1	0	0	0	0
1	1	1	1	1	0
1	1	1	1	1	1
1	1	1	1	1	1
1	0	1	0	0	0
1	1	1	1	0	0
1	1	1	1	1	1
1	1	1	1	1	0
1	1	1	1	1	1
1	1	0	1	1	1
1	1	1	0	1	0

i	0	1	2	3	4	5	6	7	8	9	10	11	12
j	A	B	A	A	D	A	C	B	A	A	B	C
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1 C	0	0	0	0	0	0	1	1	1	1	1	1	1
2 B	0	0	1	1	1	1	1	1	2	2	2	2	2
3 C	0	0	1	1	1	1	1	1	2	2	2	2	3
4 B	0	0	1	1	1	1	1	1	2	3	3	3	3
5 D	0	0	1	1	1	2	2	2	3	3	3	3	3
6 A	0	1	1	2	2	2	3	3	3	4	4	4	4
7 A	0	1	1	2	3	3	3	3	3	4	5	5	5
8 D	0	1	1	2	3	4	4	4	4	4	5	5	5
9 C	0	1	1	2	3	4	4	5	5	5	5	5	6
10 D	0	1	1	2	3	4	4	5	5	5	5	5	6
11 B	0	1	2	2	3	4	4	5	6	6	6	6	6
12 A	0	1	2	3	3	4	5	5	6	7	7	7	7

Fig. 1 An example of the bit-parallel algorithm by Hyyrö for the LCS problem

complexity $O(mn\lceil \log \sigma / \log w \rceil + m\sigma)$, where w is the processor word size, the LCS computing algorithm by Hyyrö modified for the LCTS problem [5] of time complexity $O(n\sigma\lceil m/w \rceil)$, and a combination of the above by Grabowski and Deorowicz [4].

3 Bit-Parallel Algorithm for the LCTS Problem

The bit-parallel (BP) algorithm by Hyyrö [5] solving the LCTS problem is a direct application of the version for the LCS problem (Fig. 2) executed $2\sigma - 1$ times, i.e., for each possible transposition. The LCS BP algorithm maintains a bit vector V, an equivalent of a single column of matrix M, making use of the observation that neighbor cells differ by at most 1, which can be stored at 1 bit only. The positions of 0s in vector V reflect the cells of matrix M at which the value in an actually processed column increases (Fig. 1). The first stage of the algorithm (lines 01–02) is to compute the bit masks for all alphabet symbols. The mask vector B_c for any $c \in \Sigma$ contains bit 1 at position i iff $b_i = c$. In lines 03–06, the main processing is performed. Finally (lines 07–10), the number of 0s in V (the LCS length) is determined. The algorithm complexity for the LCTS problem is $O(m\lceil n/w \rceil \sigma)$.

Practical experiments show that the BP algorithm is faster than the classical DP computation about 25 ($w = 32$) or 50 ($w = 64$) times [3].

4 General Processing on the GPU

Roughly, CPUs are composed of a few (up to 4 nowadays) general processing units (cores), which operate on cache memory (of size of an order of 1 MB) and RAM (of size of an order of 1 GB). GPUs are designed in a different way (see Table 1 for comparison of the CPU and GPU programming models). There are tens of multiprocessors (30 in NVidia's GTX 285) and each of them contains a few cores (8 in NVidia's family) that gives hundreds of cores in total (240 in GTX 285). *Global* memory, shared by all multiprocessors (of size of an order of 1 GB), is not cached

---

BitParLCS()

{*Preprocessing*}
01   **for** $c \in \Sigma$ **do** $B_c \leftarrow 0^n$
02   **for** $i \leftarrow 1$ **to** n **do** $B_{b_i} \leftarrow B_{b_i} \mid 0^{n-i} 10^{i-1}$
{*Computing LCS*}
03   $V \leftarrow 1^n$
04   **for** $j \leftarrow 1$ **to** m **do**
05       $U \leftarrow V \;\&\; B_{a_j}$
06       $V \leftarrow (V + U) \mid (V - U)$
{*Calculating number of 0's in V*}
07   $r \leftarrow 0; \quad V \leftarrow \tilde{\;}V$
08   **while** $V \neq 0^n$ **do**
09       $r \leftarrow r + 1; \quad V \leftarrow V \;\&\; (V + 1)$
10   **return** r

Fig. 2 Bit-parallel algorithm computing the LCS length by Hyyrö [5]

Table 1 Advantages and disadvantages of CPUs and GPUs

	Advantages	Disadvantages
CPU	• easy communication of threads • large multilevel cache • independent cores (MIMD architecture) • many GP languages • a lot of experience on CPU programming	• only a few cores (up to 4 nowadays) • moderate memory bandwidth • small number of GP registers
GPU	• large number of cores • huge memory bandwidth • large number of fast registers • highly parallel architecture	• dependent cores (MIMT architecture) • global memory not cached • small shared memory • limited communication between threads • much care of effective memory access • specialized programming languages • little experience on GPGPU

and much care of their effective use is needed. The other types of memory (small, about tens of KBs per multiprocessor) are:

- *shared* – shared by all threads working on a single multiprocessor (cached),
- *constant* – similarly as above, but read-only and of different access (cached),
- *local* – similar as *shared* (not cached).

Each multiprocessor contains also 8 192 or 16 384 registers. Finally, *global* memory can be accessed by texture fetch (cached, optimized for 2D access).

In general, a proper choice of accessing memory is a crucial part of efficient programming for the GPU. To achieve maximum bandwidth and reduce the number of serializations it is required to properly align the memory requests – roughly speaking, successive threads should access successive words of *global* memory to make the requests coalesced. Accesses to *shared* memory should be made in a way avoiding bank conflicts [12].

There are hundreds of cores in the GPU and the number of threads should be one or two orders of magnitude larger, to gain maximum from the architecture, e.g., to 'hide' the delays of accessing *global* memory (about 500 cycles per access). The threads are grouped in *warps* (32 threads per warp) and warps are grouped in blocks (1 to 16 warps per block). The threads within a block are executed at the same multiprocessor and may synchronize efficiently. Several blocks can be also executed at the same multiprocessor (if there are sufficient free registers, *shared* and *local* memory), but in general threads of different blocks cannot synchronize and they may communicate only by *global* memory.

The GPU architecture is called by NVidia *Single Instruction Multiple Thread* (SIMT). It is similar to *Single Instruction Multiple Data* (SIMD) in this way that a single instruction is executed at many cores in parallel. A key difference is that in SIMT the threads can diverge, i.e., even every single thread can execute separate code. This means serialization, but sometimes only a small part of the code is different, and its serialization is acceptable.

5   Proposed Bit-Parallel Algorithm for GPU

Hyyrö's algorithm, in the variant presented in Fig. 2, assumes that vector V (as well other vectors) fits a single computer word. This is rarely true, so the bit vectors are emulated by arrays of unsigned integers. In this case, the arithmetic and bitwise operators must be implemented carefully. In the main loop (lines 05–06), bitwise operators, & (and), | (or), can run locally (on separate integers). Also, the subtraction is a local operation, since the set bits in U are a subset of set bits in V, so no carry is generated. Unfortunately, the addition may produce a carry, which need to be handled. In a serial algorithm (adding the arrays integer by integer) this is simple, but not in a parallel version.

We examine three ways of parallelizing the BP algorithm. The simplest one is to assume that each of $2\sigma - 1$ separate threads solves the LCS problem for a given t. This is a reasonable strategy for a CPU, where the number of cores is much smaller than the number of transpositions. A GPU architecture is, however, quite different and it is required that at least 32 threads are processed at one multiprocessor. Since top GPUs contain 30 (or even more) multiprocessors, a reasonable minimum is 960 threads (much more than typical 255 transpositions). Moreover, to hide large delays in accessing *global* memory, it is recommended that the number of threads per single multiprocessor is about 128–256. This means that such a simple parallel algorithm will make use of only a small part of a GPU and there is no perspectives to scale it up in the future, so we resigned from this design pattern for the GPU.

The other two strategies make a block of threads computing an LCS for a single transposition. Vector V is split into array of small bit-vectors (cells), i.e., $V_j(i)$ denotes the contents of the jth vector V cell (processed by the jth thread), after computing the ith column. To handle carries, the threads must be synchronized in some way. In the first approach, in the ith step all the threads compute the ith column of V. The propagation of carry between threads can be slow, so a fast addition

techniques of large integers is necessary. Two main categories of such adders are *carry-look-ahead adder* and *conditional sum adder* [6]. We selected the latter, since it is better suited for software implementation. The propagation of the carries for a single column is made in $O(\log_2\lceil n/w\rceil)$ cycles.

Finally, the cells of V can be computed in an anti-diagonal manner, i.e., in kth parallel step all $V_j(i)$ satisfying $i + j = k$ are computed. This guarantees that when $V_j(i)$ is to be computed, for some i, j, the carry from $V_{j-1}(i)$ is already known. For p threads per transposition, there are $m + p - 1$ parallel steps. An important drawback of this method (in the GPU model) is that each thread processes a different column, so the bit masks vectors are accessed randomly.

6 Technical Details and Experimental Results

For comparison, the classical BP algorithm (CPU version) was parallelized in the simplest way: one thread–one transposition. In fact, in the preliminary experiments, we found that grouping transpositions into threads gives a little speedup on the CPU, so in the experiments there are as many threads as number of cores in the CPU and each computes several transpositions.

In the fast-adders version, the read of mask bit vector is coalesced (a large part of memory is red at one cycle (cf. [12, Sect. 5.1.2]) to improve memory bandwidth. Also, the symbols describing the consecutive columns are red to *shared* memory in a coalesced way.

The memory access pattern in the anti-diagonal method is not well suited for the GPU and leads to large delays. The size of the mask vectors is too large to be stored in *shared* or *constant* memory. Also the texture fetch was (in a preliminary experiment) unsuccessful, since the locality of accesses does not fit the texture cache assumptions. Therefore, we carefully designed our own 'cache' in *shared* memory. Each 4th thread reads 128-bit integer from the masks array (reading 128 bits and 32 bits costs the same) and use the first 32-bit word of it, while the other words are stored in *shared* memory for future use of other threads (preliminary experiments showed significant advantage of this access pattern comparing to access 32 bits from *global* memory by each thread). Alas, such memory access is not coalesced so some memory bandwidth is wasted. It is possible to make a coalesced read of masks, but it needs to store in *shared* memory masks for 16 consecutive columns and (to fit in small *shared* memory) this requires to process sequence B in small parts (preliminary experiments proved this strategy unsuccessful). The symbols describing columns are red similarly, like in the fast-adders method. We experimented also with various sizes of V handled by a single thread, but the fastest one (for $n \leq 8192$) appeared the simplest: one thread – one 32-bit word, so we use this strategy in the main experiments.

The speed of the following BP algorithms were evaluated:

- ci1 – CPU (Intel Q6600, 2.4 GHz), 1 thread, $w = 32$,
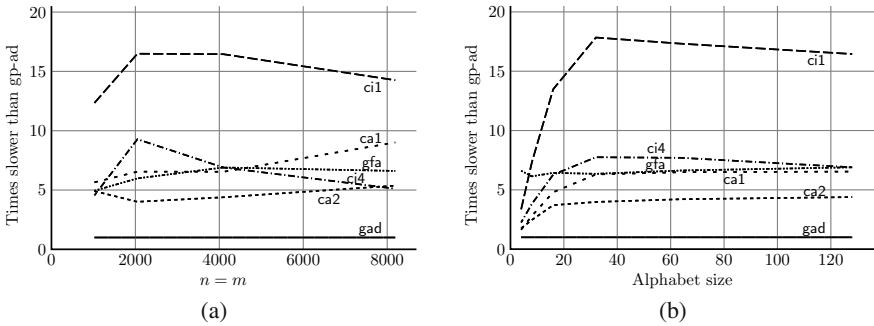- ca1 – CPU (AMD Athlon64X2 5000+, 2.6 GHz), 1 thread, $w = 64$,

Fig. 3 Comparison of the bit-parallel algorithms on the CPU and GPU: (**a**) $\sigma = 128$, variable sequence lengths, (**b**) $n = m = 4096$, variable alphabet size

- ci4 – CPU (Intel Q6600, 2.4 GHz), 4 cores, 4 threads, $w = 32$,
- ca2 – CPU (AMD Athlon64X2 5000+, 2.6 GHz), 2 cores, 2 threads, $w = 64$,
- gad – anti-diagonal on GPU (GF 8800Ultra, 1.5 GHz), 128 cores, $w = 32$,
- gfa – fast adders on GPU (GF 8800Ultra, 1.5 GHz), 128 cores, $w = 32$.

All the times were median values over 101 executions. For better presentation the times were normalized – the slow down according to gad algorithm is given.

In the first experiment (Fig. 3a), the alphabet size was $\sigma = 128$ (sequence generated randomly with uniform distribution), to emulate a typical situation in musical field (MIDI files). The equal-length sequences were of various length. The cis algorithm is about 15 times slower than gad. Due to the equal word size this result show best how many times the GPU version was sped up by parallelization according to the serial CPU algorithm. The ci4 algorithm (the fastest obtained at 4-core 32-bit CPU) is still 5–7 times slower than gad. The 2-core 64-bit version (ca2) is about 5 times slower than gad. The gfa algorithm was much slower than gad, which means that the additional cost of carry propagation was much larger than the cost of random accesses to the memory.

In the second experiment, we focused on the alphabet size. For $\sigma \leq 8$, the number of transpositions (blocks) is less than the number of the GPU cores and the speedup is small. For $\sigma = 16$ there is almost 2 times as many blocks as cores and the speedup looks good, but it increases even more for larger σ. This is due to the properties of the GPU, which 'hides' some delays in memory access if a few blocks can be assigned to a single multiprocessor.

7 Conclusions

In this paper, an approach of solving the LCTS problem on the GPU was presented. We chose one of the fastest classical algorithms computing an LCTS to see how it can be sped up on the GPU. The LCTS problem consists of many independent subproblems (LCS), so its parallelization perspectives looked very well. From the

opposite, the huge number of random memory accesses by the bit-parallel algorithm meant that slow *global* memory is in common use, which of course is not well suited to the GPU architecture.

The observed speedup between the mainstream (of similar market price) GPU and CPU (serial version) was about 15-fold. Moreover, even the parallelized version for CPU was several times slower. As it can be seen in the experimental results, our algorithm scales well in the measured range, and we can expect it will scale even more for large enough alphabets, probably up to the number of cores equal the alphabet size.

A general processing on the GPU is a fresh alternative to the CPU computing. The computing model is different from that known from the CPU world. Therefore some experience is necessary to gain from the assets of the GPU architecture. Nevertheless, thanks to the huge number of cores in GPUs and the relative easiness to use tools, like the CUDA library, the GPGPU is an attractive way of speeding up the algorithms by parallelization.

References

1. Bergroth, L., Hakonen, H., Raita, T.: A survey of longest common subsequence algorithms. In: Proceedings of String Processing and Information Retrieval Symposium, pp. 39–48 (2000)
2. Crawford, T., Iliopoulos, C., Raman, R.: String matching techniques for musical similarity and melodic recognition. Computing in Musicology 11, 71–100 (1998)
3. Deorowicz, S.: Speeding up transposition-invariant string matching. Information Processing Letters 100(1), 14–20 (2006)
4. Grabowski, S., Deorowicz, S.: Nice to be a Chimera: A hybrid algorithm for the longest common transposition-invariant subsequence problem. In: Proceedings of the 9th International Conference on Modern Problems of Radio Engineering, Telecommmunication and Computer Science, Lviv-Slavsko, Ukraine, pp. 50–54 (2008)
5. Hyyrö, H.: Bit-parallel LCS-length computation revisited. In: Proceedings of the 15th Australasian Workshop on Combinatorial Algorithms (2004)
6. Koren, I.: Computer Arithmetic Algorithms. A.K. Peters, Limited (2002)
7. Lemström, K., Navarro, G., Pinzon, Y.: Practical algorithms for transposition-invariant string-matching. Journal of Discrete Algorithms 3(2-4), 267–292 (2005)
8. Lemström, K., Tarhio, J.: Searching monophonic patterns within polyphonic sources. In: Proceedings of Recherche d'Information Assistée par Ordinateur, pp. 1261–1279 (2000)
9. Lemström, K., Ukkonen, E.: Including interval encoding into edit distance based music comparison and retrieval. In: Proceedings of Symposium on Creative & Cultural Aspects and Applications of Artificial Intelligence & Cognitive Science, pp. 53–60 (2000)
10. Mäkinen, V., Navarro, G., Ukkonen, E.: Transposition invariant string matching. Journal of Algorithms 56(2), 124–153 (2005)
11. NVidia Corporation: CUDA Zone—The resource for CUDA developers, `http://www.nvidia.com/object/cuda/_home.html`

12. NVidia Corporation: NVidia CUDA^{TM}Programming Guide, version 2.1 (12/08/2008),
    http://www.nvidia.com/object/cuda_get.html
13. Trevett, N.: OpenCL. The open standard for heterogeneous parallel programming,
    http://www.khronos.org/developers/library/overview/opencl_
    overview.pdf

Part X
Databases and Data Warehousing

Usage of the Universal Object Model in Database Schemas Comparison and Integration

Marcin Budny and Katarzyna Harężlak

Abstract. The issue of database schemas comparison and integration has been discussed in this paper. The research was connected with problems of database schemas integration that are maintained by various database management systems. This operation is often realized in programmers' group work as well as in database administrators' duties. Automation of this process will allow the avoidance of many mistakes and will increase efficiency of work. Fulfillment of these requirements entails transformation of mechanisms supported by one database management system to mechanisms existing in the other. To make this transformation possible and easy the universal object model was proposed in the research. This model contains database schemas read from given database management systems. Thanks to this, information can be compared without taking its derivation into consideration.

Keywords: database schema, integration, comparison.

1 Introduction

In recent years, computer science has been developing rapidly. Computers have revolutionized the working methods of many people, as well as influenced their way of thinking. Nowadays, almost every activity is supported by software in order to achieve better results in a shorter amount of time. There are also tasks, which cannot be performed without computers. Development tools play a vital role in the process

Marcin Budny
Silesian University of Technology,
Gliwice, Poland
e-mail: marcin.budny@gmail.com

Katarzyna Harężlak
Institute of Informatics, Silesian University of Technology,
Akademicka 16, Gliwice, Poland
e-mail: katarzyna.harezlak@polsl.pl

of production and maintenance of software systems – including database applications. This paper covers the problem of automating the database schema comparison and integration for databases maintained by different Relational Database Management Systems (RDBMS). These activities are a part of the daily work of both developer and database administrator. Automation of this process will increase work efficiency and will eliminate the possibility of making a mistake. There are already on the market some tools supporting database schema comparison but they all have one disadvantage – they are dedicated to one database server. Within these tools we can enumerate:

- SQL Compare of the RedGate company [6], and Microsoft Visual Studio Team System for Database Professionals Package [5] designated for MS SQL Server database schemas comparison,
- DBCompare of the Automated Office Systems company [1], designed to operate on various database management systems, but with usage of ODBC libraries, what can lead to loss of specific solutions information.

In order to integrate schemas of databases maintained by two different RDBMS, mechanisms provided by one of them have to be translated to mechanisms of the second. In this paper, a universal object model which represents all database objects loaded from compared databases has been proposed. Employment of this model allows ignoring database object origin during the comparison.

2 Metadata Access Methods

One of the basic activities one has to perform during database schema comparison is the loading of information about database objects from server's resources. Some of those objects are described in a fully relational way and the others in form of textual DDL definition. An application can access this data by executing SELECT command of SQL language on system tables and views. Table 1 contains a listing of selected system views in SQL Server and Oracle systems [3, 4].

3 The Universal Object Model

In order to compare information extracted from the system views, a structure called universal object model has been proposed. All database objects, regardless of the original RDBMS, are transformed to universal model objects. The model is presented as UML diagram in the Fig. 1.

Classes of the model and their relations form a tree (except for foreign keys). The Database class, representing a database, is the root of this tree. It contains a collection of instances of the Schema class, which represents a database schema. This class in turn contains collections of instances of the Sequence and Table classes. Other database objects are represented in a similar way.

Table 1 Listing of selected system views with database object information

Object type	SQL Server 2005	Oracle 10.2g
Database schema	sys.schemas	ALL_USERS
Column	sys.columns	ALL_TAB_COLS
Datatype	sys.types	ALL_TYPES
Primary key	sys.key_constraints	ALL_CONSTRAINTS
Unique key	sys.key_constraints	ALL_CONSTRAINTS
Foreign key	sys.foreign_keys	ALL_CONSTRAINTS
	sys.foreign_key_columns	ALL_CONS_COLUMNS
CHECK constraint	sys.check_constraints	ALL_CONSTRAINTS
Index	sys.indexes	ALL_INDEXES
	sys.index_columns	ALL_IND_COLUMNS
None	(not applicable)	ALL_SEQUENCES
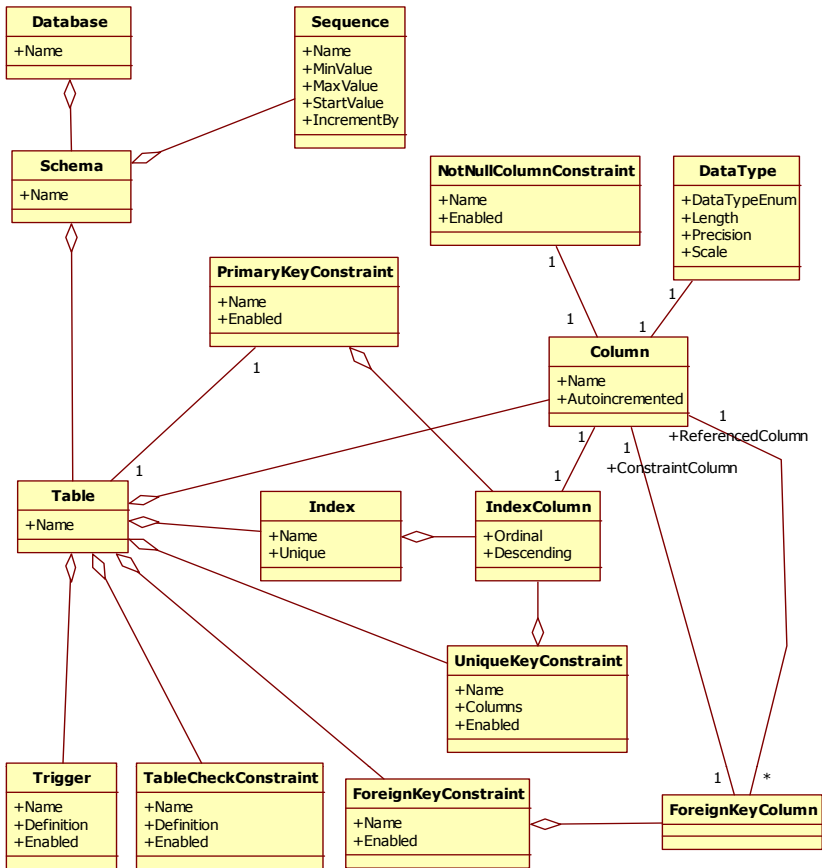
Fig. 1 The universal object model of database schema

Two of the classes require special consideration: *IndexColumn* and *ForeignKey-Column*. *IndexColumn* class represents index column and extends definition of a regular column (the Column class) with ordinal number and sorting order information. The ordinal number (Ordinal property) specifies a position of the column in index's definition. The *IndexColumn* class is also used in primary key's and unique key's definitions (*PrimaryKey* and *UniqueKey* classes respectively) because their columns are defined in the same way as index columns. The *ForeignKeyColumn* class represents a column of a foreign key and relates two columns existing in separate tables. The column in 'child' table is denoted *ConstraintColumn*, and the column in 'parent' table – *ReferencedColumn*. Definition of a foreign key can contain more than one column.

The object model is designed so that it is as universal as possible and able to store information originating from different RDBMS. For that reason it contains some object types which are not supported in all RDBMS. An example of such is a sequence, which is not supported by SQL Server system. Another example is an ability to define the sorting direction of a primary key. It is supported by SQL Server, but not by Oracle.

4   Universal Object Model Implementation

The universal object model has been implemented in the application for integrating database schemas maintained by different relational database management systems. Classes representing database objects have been defined in *ComDb.Core* package. A simplified UML diagram showing supporting classes for the object model is presented in the Fig. 2. All database objects have a name and some other common properties and methods, which have been extracted and allocated to the base and helper classes. The most basic element of this model is the *INamedObject* interface representing named database object. Additionally, this interface contains the *ComparisonInfo* property, which handles information about the result of the comparison of a given database object with another database object of the same type. This information consists of result of the comparison (*ComparisonResult* enumeration) and a flag to indicate whether or not the application user wishes to include this object into a differential DDL script generation process (user sets this flag while browsing database object tree in main application window). *ComparisonResult* enumeration type has following elements:

- *NoAction* – The object in the source database is the same as in the destination database, so there is no need to perform an action.
- *Create* – The object exists in the source database, but does not exist in the destination database, which means it needs to be created.
- *Drop* – The object does not exist in the source database, but does exist in the destination database, which means it needs to be dropped.
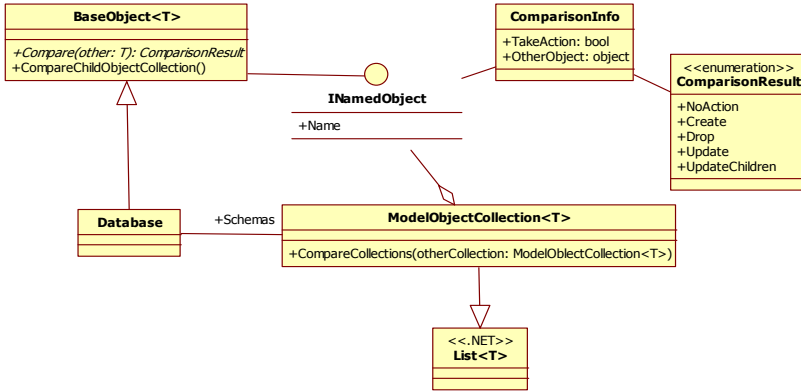
Fig. 2 Supporting classes of the universal object model

- *Update* – Definition of the object in the source database differs from the one in the destination database, which means the object needs to be altered, so that the definition in the destination database matches the one in the source database.
- *UpdateChildren* – The object is the same in the source and destination databases, but some of its child objects (down the tree hierarchy) require taking one of the following actions – create, drop or update.

The object from the source database is a basis, to which the object from the destination database is compared. A reference to the destination object is stored in the source object, in *ComparisonInfo* property.

The *BaseObject* class implements the *INamedObject* and defines methods and properties common to all objects of universal object model. The diagram in Fig. 2 shows a sample inheritance relation between *BaseObject* and *Database classes*. *BaseObject* class defines *Compare* abstract method, which is implemented by derived classes. This method compares two database objects of the same type and returns the result in the form of *ComparisonResult* enumeration type. If a given object has child objects (for example a database has a collection of schemas), *Compare* methods of these child objects are called recursively.

The *ModelObjectCollection* is also a significant element, which represents the collection of universal object model objects. It derives from the *List < T >* class, which is a part of .NET Framework platform [7]. It can contain only instances of classes implementing *INamedObject* interface. Apart from standard list functionality (adding, removing, searching objects), this class allows the search of objects by name with case insensitivity and the comparision of two collections. The result of the latter is information about which objects are in both collections and which are only in one of them. This functionality is used to determine which objects are to be created, dropped or updated.

The *ModelObjectCollection* class is used to store a list of child objects of a given database object. As an example, there is a relation marked on the diagram showing that the *Database* class has a collection of instances of *Schema* class.
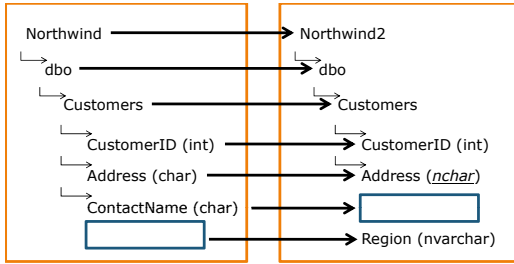
Fig. 3 The comparison of two database schemas

5   Comparison of Two Database Schemas

The comparison of two database schemas is performed recursively, starting from the root of the tree representing a database schema and moving downwards to the leaves. Since two databases are compared, there are two object trees in the memory, one of which represents the source database and the other – the destination database. The comparison algorithm processes these two trees simultaneously. Objects for comparison at a given tree level are chosen by name. An assumption is made for RDBMS, that in a given database schema scope, objects of a particular type have unique names. Figure 3 shows a sample diagram of comparison between two database schemas. It is a modified fragment of the sample Northwind database.

There are two databases compared in the example: Northwind and Northwind2. A diagram shows the comparison of the Customers tables in dbo schema. There are four possible cases shown:

- The CustomerID column is identical in both databases and does not require any action.
- The Address column in the source database has a different data type than in the destination database. It has to be therefore modified in the destination database, so that it is of CHAR data type instead of NCHAR.
- The ContactName column exists only in the source database. It has to be created in the destination database.
- The Region column exists only in the destination database. It has to be dropped in the destination database for the schemas to become identical.

Figure 4 shows a simplified UML sequence diagram [2] explaining operations performed during the comparison of the two databases. For clarity, calls on *ModelObjectCollection* class instances were omitted.

Each object of the universal object model implements the *Compare* method, which is defined in BaseObject base class. This method performs a comparison specific to a given object type. For example comparison of two tables includes comparison of: primary key, column list, unique key list,index list, trigger list, CHECK constraints. Comparison of the two table columns includes a comparison of: data type, autoincrementation property, NOT NULL constraint. The comparison results
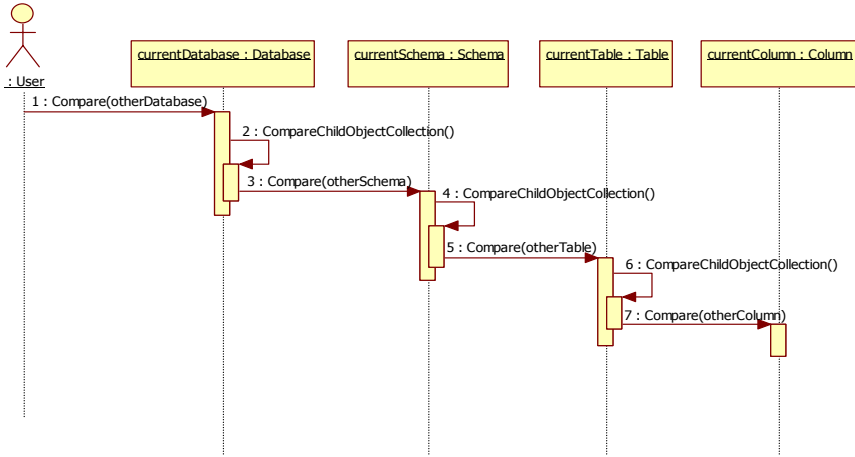
Fig. 4 UML sequence diagram for database schema comparison operation

in information regarding whether a source database object differs from a destination database object or if there is a difference in the lower level objects. This information is displayed to the user during the difference verification phase. It is also used during the generation of differential scripts.

It is also possible to compare objects without the schema name being taken into account. This can be performed under the condition that only one database schema is loaded both from the source and the destination database.

6   The Application Using the Universal Object Model

In order to start working with the application, the user has to set the connection parameters for both source and destination database. Figure 5 shows connection parameter windows for SQL Server and Oracle systems. In both cases user has to provide parameters specific to given RDBMS – for example server name, username and password, database name.

In the next step user runs the database comparison process by pressing the *Compare* button in the application's main window. The application's window appearance after this operation is presented in Fig. 6. Area (1) contains a tree of database objects. It consists of merged information about both databases: all objects from the source database and objects, which exist only in the destination database. Apart from the column containing the object tree, area (1) contains also columns with information about the source object (2), the destination object (4) and the action to be performed (3). These actions are also symbolized by colors of the table's rows. User decides which actions are to be included in the differential DDL script by selecting checkboxes in the object tree. Selecting a node at certain level effects in all child nodes being selected. The same happens when a node is deselected. Text fields (5) and (6)
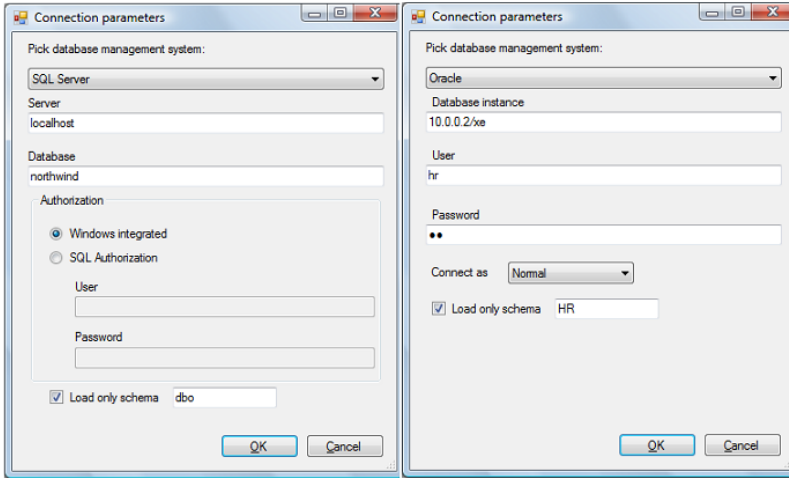
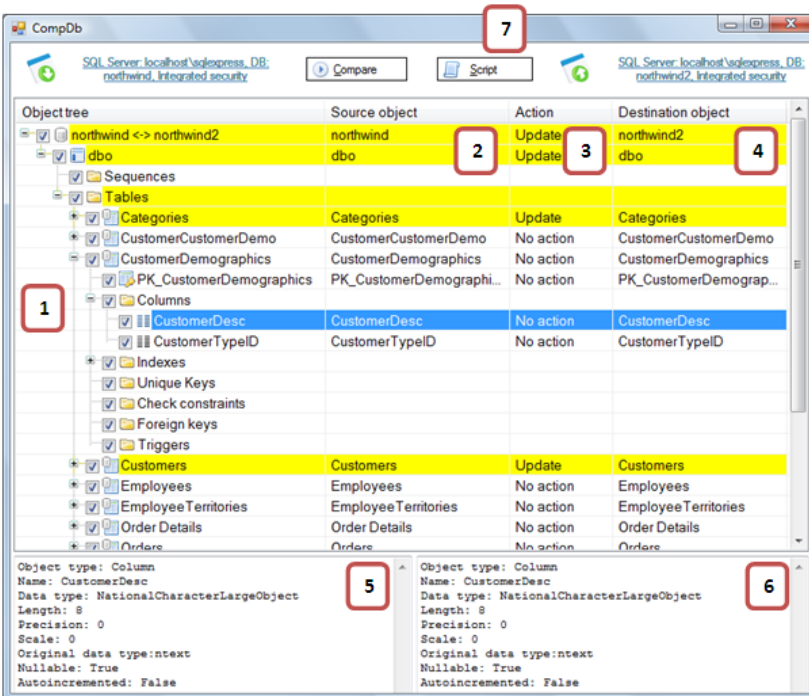**Fig. 5** Connection parameter windows for SQL Server and Oracle RDBMS

Fig. 6 The application's window after comparison between database schemas

display textual description of objects' definition for the source and the destination databases respectively.
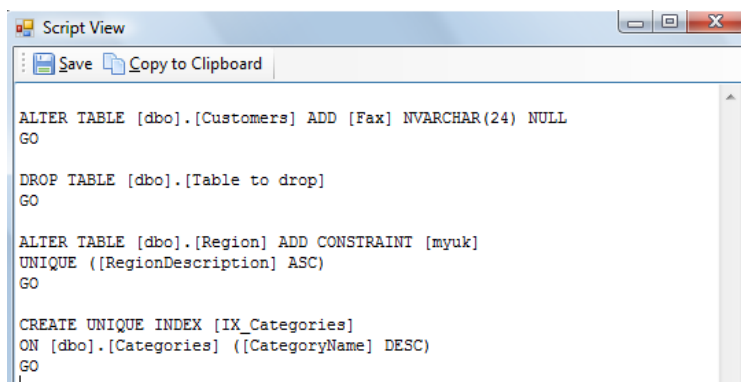
Fig. 7 The script view window

After verification of the differences between databases and selection of actions which are to be performed, users start a generation of scripts by clicking button (7). The script view window presented in Fig. 7 is displayed when generation is finished.

7 Conclusion

In this paper the problem of database schema comparison has been analyzed. This problem is often encountered by database application programmers and database administrators. Because of the lack of suitable tools, the process of database schema synchronization has to be performed manually, especially when databases are managed by different database management systems.

The solution to this problem is the usage of the universal object model, presented in this paper. Its hierarchical structure enables the gathering of information of database schema regardless of the servers ownery of these databases. An important aspect of this model is the ability to maintain specific object features. The proposed model was implemented in the sample application. The results of its operation confirmed the correctness of the proposed solution, which creates new possibilities in the automation of database schema management tasks.

References

1. Automated Office Systems, Inc.: Automated office systems. dbcompare (2007),
   http://www.automatedofficesystems.com/products/dbcompare/
2. Fowler, M.: UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley, Reading (2003)
3. Microsoft Corporation: Microsoft SQL Server Books Online (2008),
   http://msdn.microsoft.com/en-us/library/ms130214.aspx

4. Oracle: Oracle Database online documentation 10g Release 2,
   `http://www.oracle.com/pls/db102/homepage`
5. Randell, B.A.: Introducing Visual Studio 2005 Team Edition for database professionals (2007),
   `http://msdn.microsoft.com/en-us/magazine/cc163472.aspx`
6. Red Gate Software Ltd.: SQLCompare (2007),
   `http://www.red-gate.com/products/SQL_Compare/index.htm`
7. Troelsen, A.: Pro C# 2008 and the .NET 3.5 Platform. Apress (2007)

Computational Model for Efficient Processing of Geofield Queries

Piotr Bajerski and Stanisław Kozielski

Abstract. Many spatial data analyses demand processing geographic phenomena varying continuously over space, e.g., air pollution. In the paper such phenomena are called geofields and queries concerning them – geofield queries. When geofields are represented in a database as point measurements, than answering geofield queries requires computationally intensive interpolation. Writing a spatial analysis as one declarative geofield query makes it possible to optimize its execution. Such an optimization was the goal of the presented computational model consisting of: (a) discrete computational space, (b) relational representation of geofields based on Peano N space-filling curve and RVA, and (c) a set of operations extending relational algebra. The presented computational model made it possible to evaluate geofield queries by two orders of magnitude faster than in classical raster representation.

Keywords: spatial databases, GIS, query optimization.

1 Introduction

Many spatial data analyses demand processing geographic phenomena varying over space, e.g., air pollution, temperature or rainfall, as well as discrete spatial entities, e.g., districts, roads or bus stops [4, 6]. In this paper phenomena varying continuously over space are called *geofields* and discrete spatial entities are called *geoobjects*. Geofields are divided into quantitative geofields and qualitative geofields. A *quantitative geofield* is a two-dimensional function assigning elements of its domain values measured on interval or ratio scale. A *qualitative geofield* is a function assigning elements of its domain values measured on nominal or ordinal scale, e.g., numbers of intervals of a geofield values.

Piotr Bajerski · Stanisław Kozielski
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-101 Gliwice, Poland
e-mail:{piotr.bajerski,stanislaw.kozielski}@polsl.pl

The term *geofield query* denotes a query containing references to qualitative ge-ofields, e.g., *Retrieve names of counties over which whole area in 2008 average-yearly distribution of sulfur dioxide (SO₂) exceeded* $64\,\mu g/m^3$ *and distribution of airborne suspended matter fell between* $165\,\mu g/m^3$ *and* $220\,\mu g/m^3$. In the research, it was assumed that quantitative geofields are represented in a database as collec-tions of values measured in irregular measurement networks. Therefore, to answer geofield queries we need a procedure to estimate geofields values in points in which they were not measured. Such a procedure, based on interpolation, is called *geofield (mathematical) model*.

In [1] there was presented an extension of SQL, called GeoField Query Lan-guage (GFQL), facilitating writing geofield queries. Assuming that models of the geofields from the example query are stored in metadata under the names `SO2_08` and `SM_08` respectively, the query may be written in GFQL as:

```
SELECT c.Name
FROM Counties c, SO2_08 s Intervals (64),
    SM_08 m Intervals (165, 220)
WHERE s.Interval = 1 And m.Interval = 1 And
    Resolution = 0.05km And
    (c.Shape.Within(s.Shape.Intersection(m.Shape)) = 1 Or
     c.Shape.Equals(s.Shape.Intersection(m.Shape)) = 1);
```

The phrase `Intervals` allows for defining intervals of geofields values, which can be referenced in the query by the predefined attribute `Interval`. The `Inter-section` operation computes intersection of the given geometries while predi-cates `Within` and `Equals` check topological relations. Computations are to be carried out in the resolution determined by the condition on the pseudo-attribute `Resolution`.

In GIS, geofields are usually processed in a raster representation and capabili-ties to reference geofields in queries are very limited, so they are processed by a sequence of independent transformations. Writing a spatial analysis as one declara-tive geofield query makes it possible to optimize its execution. Such an optimization was the goal of the presented computational model. During its construction, it was assumed that users formulating queries are interested in some intervals of geofields values and the answers are to be presented in predetermined resolutions. The pre-sented *computational model for geofield processing* may be thought of as an abstract machine for processing geofields queries. Its main parts are constituted by: (a) dis-crete representation of the geographic space – Sect. 2, (b) discrete representation of geofields and geoobjects (called PNR representation) and integrity constraints – Sect. 3, and (c) set of operators for processing geofields and geoobjects – Sect. 4.

2 Computational Space

The area in which computations must be carried out to answer a geofield query is called *query context region*. In GFQL, a query context region may be ex-plicitly determined in the `Where` clause by a condition on the pseudo-attribute

`ContextRegion` as: a rectangle in geographic coordinates, a set of polygons returned by a subquery or an intersection of a rectangle and a set of polygons. A query context region may be also given implicitly by a collection of geoobjects referred to in the query [1]. The part of the query context region in which values of the given geofield must be computed to answer the query is called *geofield context region*. The context region of a given geofield is always determined in the context of the given query so it is a subset of the intersection of the geofield domain and the query context region.

In GFQL queries usually geographical coordinates are used. To ease the computations there is introduced a local coordinate system in which the curvature of the Earth may be neglected and distances are measured in meters relatively to lower left vertex of the minimal bounding rectangle (MBR) of the query context region. MBR of the query context region and the resolution given in a query determine the size of the smallest chunks of space that are not divided during computations. These chunks are called *elementary quadrants*. Next, the local coordinate system is transformed into *computational coordinate system*, in which the side length of the elementary quadrants is used as the unit. The area with the computational coordinate system and the Euclidean metric, within which the error introduced by the curvature of the Earth may be ignored, is called *continuous computational space*. Finally, the MBR of the query context region is tiled by elementary quadrants that are ordered by the Peano N space-filling curve. Coordinates of the elementary quadrant are called *discrete coordinates*. For example, discrete computational space shown in Fig. 1 has discrete coordinates ranging from 0 to 3. It is worth mentioning that localization of the nodes of measurement networks does not influence the location of discrete computational space. Its location depends only on geoobjects determining the query context region.

3 Peano Relation and PNR Representation

In [5] there was presented an idea of approximation of spatial location of geoobjects by quadtrees, storing these quadtrees ordered by Peano N space-filling curve in relations and processing them by extended relational algebra, called Peano-tuple algebra. Relations storing key ranges of Peano N space-filling curve were called Peano relations. In [5] a schema based on minimal Peano key value for a quadrant and quadrant side length was preferred. From now on, Peano relations with such kind of schema will be called *original Peano relations*.

For example, let us assume that we want to store information about counties identified by `Code` and described by `Name` and `Shape` (location in space). Using an original Peano relation we receive a schema `CountiesOld(`Code`, Name,` `KPmin, SideLength)`, which must be denormalized into the schemas: `Coun-` `tiesOldA(`Code`, Name)` and `CountiesOldB(`Code`, `KPmin`, Side-` `Length)` because of the partial key dependency: *Code → Name*. Attributes `KPmin` and `SideLength` may be replaced by attributes `KPmin` and `KPmax` storing Peano key range for a quadrant or continuous Peano key range for many quadrants. In

such case, the second relation would have schema: `CountiesOldB2(Code,` `KPmin, KPmax)`.

In the presented computational model Peano key ranges are stored in Relation Valued Attributes (RVA) [3]. Therefore, for the example presented we would have just relation with schema `Counties(Code, Name, Shape)`, where the attribute `Shape` is a RVA with schema `(KPmin, KPmax)`. To simplify algebraic representation of geofield queries and ease their optimization, RVA with schema (KPmin, KPmax) was joined with a set of operations on it and was called *RegionPNR type*. These operations are presented in Sect. 4. From now on, Peano relations storing Peano key ranges only in attributes of the RegionPNR type will be called *Peano relations*. Storage of approximation of geometries of geoobjects and qualitative geofields in Peano relations will be called *PNR representation* of geoobject and geofields respectively.

For qualitative geofields Peano relations schemas have predefined attributes: `Interval`, `LBound`, `UBound`, and `Shape`. The attribute `Interval` stores interval number (counted from 0) described by the given tuple, attributes `LBound` and `UBound` store lower and upper bound of the interval respectively and the attribute `Shape` stores Peano keys of all elementary quadrants approximating fragments of the computational space in which values of the geofield model fall into this interval. An example of geoobject and geofield stored in Peano relations is given in Fig. 1.

In [5] three conformance levels for original Peano relations were defined: (1) well-positioned object, (2) removal of overlaps and (3) compact object. These conformance levels must be respected by the instances of the RegionPNR type. In the computational model, it is assumed that in Peano relations tuples cannot differ only on values of attributes of the RegionPNR type. This assumption solves the problem of the original Peano algebra consisting in the fact that one Peano tuple could be equivalent to many other Peano tuples as long as they describe the same set of Peano keys and fulfill the conformance levels. Introducing the RegionPNR type also facilitates natural representation and processing of empty geometries (very often coming out as a result of an intersection), which was complicated in the original Peano algebra [1].

4 Peano Algebra

In [5] a concept of Peano-tuple algebra denoting a set of operations on original Peano relations was presented. From now on, this algebra will be called *original Peano algebra*. Introducing the RegionPNR type and focusing on processing geofields led to splitting operators from the original algebra into two groups: (1) containing operators taking whole Peano relations as arguments and treating attributes of the RegionPNR type as atomic values, and (2) containing operators taking values of the RegionPNR type as arguments. Both groups were modified to facilitate geofields processing. The main idea of the original Peano algebra was to enable processing of spatial data without employing computational geometry. The presented version of the algebra extends this idea to processing geofields by providing a
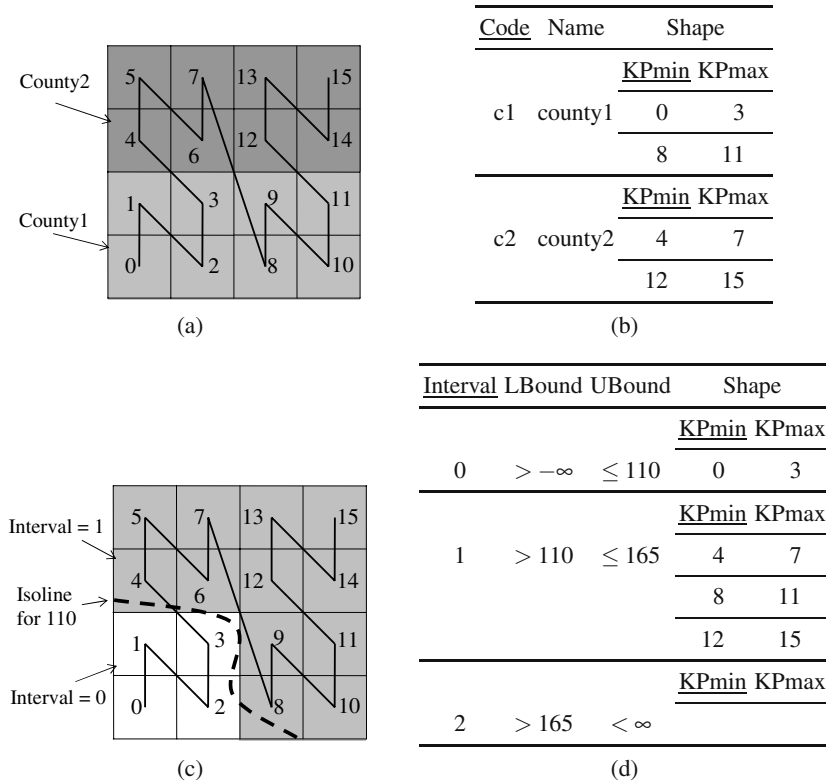
Code	Name	Shape	
		KPmin	KPmax
c1	county1	0	3
		8	11
		KPmin	KPmax
c2	county2	4	7
		12	15

(a)   (b)

Interval	LBound	UBound	Shape	
			KPmin	KPmax
0	> −∞	≤ 110	0	3
			KPmin	KPmax
1	> 110	≤ 165	4	7
			8	11
			12	15
			KPmin	KPmax
2	> 165	< ∞		

(c)   (d)

Fig. 1 PNR representation of exemplary geoobjects and a geofield. (a) Discrete representation of two counties. (b) Relation Counties stroning PNR representation of the geoobjects from a. (c) Discrete representation of a geofield. (d) Relation GF storing PNR representation of the geofield from c

formal notation that makes it possible to write execution plans of geofield queries in algebraic form and to formulate algebraic optimization transformation rules.

4.1 Operations of RegionPNR Type

Operations of the RegionPNR type process geometries of geoobjects in separation of other geoobjects geometries. They are mostly derived from operations defined by OGC for Simple Feature [7]:

- analytic – intersection, union, difference, symmetric difference, area, distance, buffer,
- topological relations – disjoint, overlaps, equals, contains, within, intersects,
- representation conversion, for which the most important is the conversion between vector and PNR representations.

As for topological relations, in the PNR representation interior and boundary elementary quadrants are undistinguishable so the following pairs of relations are undistinguishable: disjoint and meet, contains and covers, and within and coveredBy. For them the first name in the pair is used.

For some spatial operations care must be taken because they have to take into account whole groups of geoobject, in which case an operator of the Peano algebra must be defined instead of an operation of the RegionPNR type. This problem is shortly discussed in description of operator ψ in the next subsection.

4.2   Peano Algebra Operators

Section 4.1 presented operations processing individual geometries in the PNR representation stored in attributes of the RegionPNR type in tuples of a Peano relation. This subsection covers operators processing whole Peano relations delegating some processing of geometries to the operations of the RegionPNR type. Semantics of the relational algebra operators is based on [3].

Set Operations

Classical relational algebra contains the following set operations: intersection (denoted by $\cap$), union (denoted by $\cup$), difference (denoted by $\setminus$) and product (denoted by $\times$). In the presented Peano Algebra semantics of intersection, union and difference were extended to adjust them to processing Peano relations. In case of intersection of two Peano relations, the extension consists in computing an intersection of all corresponding attributes of the RegionPNR type for all tuples that meet the intersection requirement on the other attributes. The semantics of union and difference operations are extended analogically.

Project

Projection, denoted by π, produces a relation containing only the chosen attributes. As π may remove all candidate keys and by assumption Peano tuples cannot differ only on attributes of RegionPNR type, all tuples differing after projection only on attributes of RegionPNR type are replaced by one tuple with values in corresponding attributes of RegionPNR type merged.

Select

Selection, denoted by σ, generally works as in relational algebra. Attributes of the RegionPNR type are by definition atomic from the perspective of σ. Therefore, there is not way to directly reference Peano key ranges in a selection predicate. However, the predicate may use functions of the RegionPNR type (e.g., condition on area size).

Extend, Summarize, and Rename

Extension, denoted by ε, adds to the algebra computational abilities allowing to extend a relation with attributes storing the result of a given expression. Summarize, denoted by γ, allows to use aggregation functions. When attribute of the RegionPNR type is used in summarize it is treated as in projection. Rename, denoted by ρ, allows to rename a relation and its attributes.

Peano Join

Peano Join, denoted by $\bowtie_P$, is a kind of a join in which two relations are joined on attributes of the RegionPNR type. Peano join of two relations R and S on attributes A and B of the RegionPNR type may be formally defined by:

$$R \underset{A \cap B \to E}{\bowtie_P} S \equiv \pi_L(\sigma_{E.IsEmpty()=0}(\varepsilon_{A.Intersection(B) \to E}(R \times S))),$$
$$L = Attr(R) \setminus \{A\} \cup Attr(S) \setminus \{B\} \cup \{E\}. \tag{1}$$

The schema of the result relation is created as a union of the schemas of the joined relations without attributes used for the join (attributes A and B in (1)) extended with an attribute to hold intersections of the geometries (attribute E in (1)). Tuples of the result relation are created as combination of tuples from R and S for which intersection of geometries stored in A and B is not empty. The intersection of geometries is stored as a value of the attribute E. Table 1 shows an example of a Peano join. The presented version of Peano join is an extension of the Peano join from the original Peano algebra.

Table 1 *Counties $\bowtie_P$ GF* – Peano join of the relations `Counties` and `GF` from Fig. 1
                    Shape

CCode	CName	Interval	LBound	UBound	Shape	
					KPmin	KPmax
c1	county1	0	$> -\infty$	≤ 110	0	3
					KPmin	KPmax
c1	county1	1	> 110	≤ 165	8	11
					KPmin	KPmax
c2	county2	1	> 110	≤ 165	4	7
					12	15

Peano Natural Join

Peano natural join, denoted by $\bowtie$, is a combination of natural join and Peano join – it combines tuples of the joined relations with equal values of the common attributes except of attributes of the RegionPNR type, for which intersection of

geometries is computed and stored in the result as in the Peano join. However, tuples with empty intersections are not eliminated from the result relation as in case of the Peano join.

Peano Θ-join

Spatial joins are the joins in which the Θ-condition of the selection on product of the joined relations contains spatial predicates. Peano Θ-join, denoted by $\underset{C}{\bowtie}$, is a spatial join in which the Θ-condition references attributes of the RegionPNR type. An example of a Peano Θ-join is shown in Table 2.

Table 2 *Counties* $\underset{Counties.Shape.Within(GF.Shape)}{\bowtie}$ *GF* – Peano Θ-join of the relations Counties and GF from Fig. 1

CCode	CName	CShape		Interval	LBound	UBound	GFShape	
		KPmin	KPmax				KPmin	KPmax
c2	county2	4	7	1	> 110	≤ 165	4	7
		12	15				8	11
							12	15

Generation of PNR Representation of a Qualitative Geofield

Generation of a PNR representation of a qualitative geofield, denoted by η, results in a PNR representation of zones in which values of the geofield fall into the specified intervals. Two relations, $\eta_{\text{gf\_gen\_params,gf\_context\_area}}(R, M)$, are the arguments of the operator η. First of them, denoted by R, defines the geofield context region stored in its attribute specified by the parameter gf_context_area. The second relation, denoted by M, stores point measurements of the geofield. The parameter gf_gen_params describes: the geofield mathematical model, the quadrant classifier with its parameters, and the geofield value interval bounds.

The η operator produces a relation, which schema is an extension of the relation R schema with attributes Interval, LBound, UBound and Shape. These attributes store a PNR representation of the generated qualitative geofield. The body of the relation consists of replicated tuples from relation R extended with the PNR representation of the geofield. An example of the usage of the operator η is shown in Table 3. Implementation of the η operator is presented in [2] and covered in full detail in [1].

Generation of PNR Representation of a Layer

The RegionPNR type offers conversion between vector and PNR representations. In many queries however, we deal with a set of mutually disjoint geoobjects, which cover a given region (e.g., counties). Such a collection of geoobjects is often called

Table 3 $\eta_{gf\_gen\_params,Shape}(Counties, GF\_Measures)$ – generation of the PNR representation of the geofield GF in the area determined by the attribute `Shape` from the relation `Counties` shown in Fig. 1

CCode	CName	CShape		Interval	LBound	UBound	GFShape	
		KPmin	KPmax				KPmin	KPmax
C1	County1	0	3	0	$> -\infty$	≤ 110	0	3
		8	11					
		KPmin	KPmax				KPmin	KPmax
C1	County1	0	3	1	> 110	≤ 165	8	11
		8	11					
		KPmin	KPmax				KPmin	KPmax
C1	County1	0	3	2	> 165	$< \infty$		
		8	11					
		KPmin	KPmax				KPmin	KPmax
C2	County2	4	7	0	$> -\infty$	≤ 110		
		12	15					
		KPmin	KPmax				KPmin	KPmax
C2	County2	4	7	1	> 110	≤ 165	4	7
		12	15				12	15
		KPmin	KPmax				KPmin	KPmax
C2	County2	4	7	2	> 165	$< \infty$		
		12	15					

a layer. PNR representation of geoobjects setting up a layer must hold the requirement that every point of the area covered by the geoobjects belongs to exactly one of them. The problem is caused by the fact that the patch covered by an elementary quadrant may belong to more than one geoobject in the continuous computational space but in the discrete computational space it must be assigned to at most one PNR representation of the geoobjects. As creation of PNR representation of a collection of geoobjects setting up a layer must take into account their spatial relations a special operator, denoted by ψ, was introduced. The ψ operator takes as an argument a relation storing geoobjects from the layer and produces a relation which is an extension of the source relation with an attribute storing the create PNR representation. It is characteristic to geofield queries that a change in query context region or a change in geoobjects setting up layer (e.g., caused by pushing down selection during query optimization) may change PNR representation of the geoobjects, which in turn may influence the query answer. However, the influence decreases as the resolution used during query evaluation increases.

Other Operators

Operators for grouping and ungrouping (nesting and unnesting) relations with RVA were not included into the presented version of Peano algebra because there was no need for transformations between the presented Peano relations and the old Peano relations. Operators: translation, rotation, scaling, symmetry, simplifying and geometric projection from the original Peano algebra were left by because all PNR representations are created dynamically during query execution. If there would be needed for materialization of PNR representations, than these operators should take into account the whole layers of geoobjects and geofields as in case of the ψ operator. Also similar effects would appear as in case of the ψ operator. The extraction operator from the original Peano algebra was replaced by the more general intersection operation of the RegionPNR type. The divide operator was not included as it is rarely used and may be expressed by other operators.

5 Applying Peano Algebra to Query Optimization

This section illustrates usage of Peano algebra for algebraic representation of geofield queries and gives some references to their optimization. Let us assume that for the running example counties boundaries are stored in vector representation in column `Boundary`, point measurements for geofields of sulfur dioxide and airborne suspended matter are stored in relations `M_SO2_08` and `M_SM_08` respectively and their mathematical models read from metadata are available by parameters `SO2_gen_param` and `SM_gen_param` respectively. Than the query may be transformed into Peano algebra expression (2), which is the starting point for its optimization.

The most important optimization heuristic is to reduce geofields context regions as creating geofields PNR representation is usually the most time consuming operation. This may be achieved by using results of selection on some geofields as context region for other geofields and using results of non-spatial subqueries to restrict geofields context regions [1]:

$$
\begin{aligned}
Answer &:= \pi_{Name}(CtxRg \underset{CRg.Within(Shape)=1\,\text{Or}\,CRg.Equals(Shape)=1}{\bowtie} GF)\,, \\
GF &:= \pi_{Shape}(SO2 \underset{Shape}{\bowtie_P} SM)\,, \\
SO2 &:= \sigma_{Interval=1}(\eta_{SO2\_gen\_param,CRg}(CtxRg,M\_SO2\_08))\,, \\
SM &:= \sigma_{Interval=1}(\eta_{SM\_gen\_param,CRg}(CtxRg,M\_SM\_08))\,, \\
CtxRg &:= \psi_{Boundary \to CRg}(Counties)\,.
\end{aligned}
$$

6 Conclusions

The presented computational model laid the foundation for optimization of geofield queries and their efficient evaluation in an experimental Geofield Server [1]. Peano

algebra acted as an intermediate language into which GFQL queries were translated. Such an algebraic representation eased query optimization as classical optimization rules may be used for classical relations and new rules for Peano relations. Discretization of the computational space and facilitating to address its elements of different size by Peano keys enabled efficient implementation of Peano algebra operators and RegionPNR type operations. The presented computational model made it possible to evaluate geofield queries by two orders of magnitude faster than in raster representation. Especially speedup of generation of PNR representation of a qualitative geofield proved to improve efficiency of geofield queries processing [2].

Characteristic to geofield query optimization and processing is the influence of computational space, optimization rules and Peano algebra operators implementation on query execution time and distortion of the answer. It is difficult to optimize both of the criteria as generally shortening query evaluation time increases distortion so the optimizer must look for a compromise, which in the experimental Geofield Server was based on a hint given by a user [1].

References

1. Bajerski, P.: Using Peano algebra for optimization of domain data sets access during analysis of features distribution in space. Ph.D. thesis, Silesian University of Technology, Gliwice, Poland (2006) (in Polish)
2. Bajerski, P.: How to efficiently generate PNR representation of a qualitative geofield. In: Proceedings of the International Conference on Man-Machine Interactions, Kocierz, Poland (2009)
3. Date, C.J., Darwen, H.: Databases, Types and the Relational Model. The Third Manifesto. Addison-Wesley, Pearson Education, Incorporated (2007)
4. Laurini, R., Paolino, L., Sebillo, M., Tortora, G., Vitiello, G.: Dealing with geographic continues fields - the way to a visual GIS environment. In: Proceedings of AVI, Gallipoli, Italy (2004)
5. Laurini, R., Thompson, D.: Understanding GIS. Academic Press Limited, London (1994)
6. Miller, H., Wentz, E.: Representation and spatial analysis in geographic information systems. Annals of the Association of American Geographers 93(3) (2003)
7. Open Geospatial Consortium, Inc.: OpenGIS implementation specification for geographic information – simple feature access – part 1: Common architecture. Ver. 1.2.0. OGC Inc. (2006)

Applying Advanced Methods of Query Selectivity Estimation in Oracle DBMS

Dariusz R. Augustyn

Abstract. The paper shows the solution of the query selectivity estimation problem for certain types of database queries with a selection condition based on several table attributes. The selectivity parameter allows for estimating a size of data satisfying a query condition. An estimator of a multidimensional probability density function is required for an accurate selectivity calculation for conditions involving many attributes and correlated attribute values. Using multidimensional histogram as a nonparametric density function estimator is mostly too much storage-consuming. The implementation of the known unconventional storage-efficient approach based on Discrete Cosine Transform spectrum of a multidimensional histogram is presented. This solution extends functionality of the Oracle DBMS cost-based query optimizer. The method of experimental obtaining error-optimal parameters values of spectrum storage for typical attributes value distributions is considered.

Keywords: selectivity query estimation, multidimensional probability density function, Discrete Cosine Transform, database query optimizer extension, Oracle Data Catridge Interface Statistics.

1 Introduction

The paper presents the practical solution of a query selectivity estimation problem for range queries with a composed selection condition based on a few table attributes with a continuous domain. The selectivity factor is used by the database query optimizer for estimating the size of data satisfying query condition. Using a value of estimated selectivity the optimizer can choose the most efficient method of query executing (so-called the best query execution plan). For single-table queries the

Dariusz R. Augustyn
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: draugustyn@polsl.pl

selectivity value is a number of rows satisfying query condition divided by a number of all table rows.

Most of Database Management System (DBMS) query optimizers are based on an independence attribute value assumption (AVI [7]) that selectivity for a composite condition is a product of simple component condition selectivities. The AVI assumption is based on the probability multiplication rule for independent events. Mostly, the AVI rule using results in an inaccuracy of obtained values of a query selectivity estimator for correlated data. An estimator of a multidimensional probability density function (PDF) is required for accurate selectivity calculations for query conditions involving many attributes. For continuous attribute domains the selectivity value of a range query is a value of definite integral of multivariate PDF.

The direct use of a multidimensional histogram as a non-parametric estimator of PDF is very space consuming for high dimensions. A space-efficient method of representation of attribute value joint distribution is needed. There are many techniques of a multidimensional distribution representation used for selectivity estimation e.g. kernel estimator [8, 4], PHASED [7], MHIST [7], GENHIST [4], STHoles [1], Bayesian Network [3], Discrete Cosine Transform [5], cosine series (with a triangular sampling zone) [9], Discrete Wavelets Transform [2], and many others.

This paper concentrates on the application of the approach using Discrete Cosine Transform (DCT [5]). The presented software solution extends the functionality of the Oracle DBMS query optimizer by using the Oracle Data Cartridge Interface Statistics module [6]. It is the implementation of a spectrum-based selectivity estimation method.

The method of an experimental obtaining of selectivity estimation error-optimal parameters for multidimensional DCT spectrum storage is also presented.

2   Discrete Cosine Transform Spectrum Representation of Multidimensional Attribute Value Distribution

The approach to the selectivity estimation method using DCT was proposed in [5], where a space-efficient DCT-spectrum representation of a database table attribute values distribution was considered. The spectrum-based selectivity calculation method was proposed in [5] as well.

For a 2-dimensional case and given definitions:

- X, Y – attributes of relation R; both with continuous domain,
- $F - \{f(m,n) : m = 0, \ldots, M-1 \land n = 0, \ldots, N-1\}$, $M \times N$ matrix of frequencies, estimator of PDF for joint distribution of X and Y, values of a 2-dimensional equi-width histogram,
- $G - \{g(u,v) : u = 0, \ldots, M-1 \land v = 0, \ldots, N-1\}$, $M \times N$ matrix of DCT coefficients,

The 2-dimensional Discrete Cosine Transform (DCT spectrum) is defined as follows:
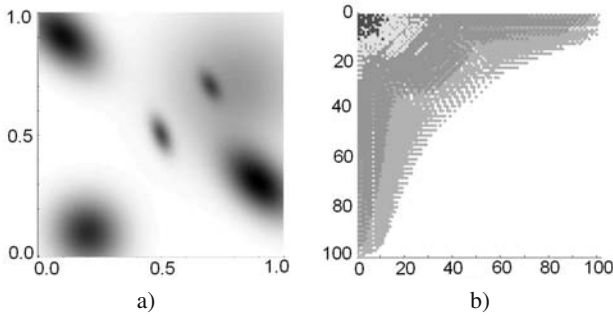
Fig. 1 (a) Sample bivariate PDF of 6 Gaussian clusters. **(b)** Corresponding DCT spectrum with regions of comparable coefficient absolute values

$$
\begin{aligned}
g(u,v) \quad &= \sqrt{\tfrac{2}{M}} k_u \sum_{m=0}^{M-1} \left\{ \sqrt{\tfrac{2}{N}} k_v \sum_{n=0}^{N-1} f(m,n) \cos\left(\tfrac{(2n+1)v\pi}{2N} \right) \right\} \times \\
&\quad \times \cos\left(\tfrac{(2m+1)u\pi}{2M} \right),
\end{aligned}
\tag{1}
$$

$$
\text{where } k_r = \begin{cases} 1/\sqrt{2} & \text{for } r = 0, \\ 1 & \text{for } r \neq 0, \end{cases}
$$

and the 2-dimensional Inverse Discrete Cosine Transform (IDCT) is defined below:

$$
\begin{aligned}
f(m,n) \quad &= \sqrt{\tfrac{2}{M}} \sum_{u=0}^{M-1} k_u \left\{ \sqrt{\tfrac{2}{N}} \sum_{v=0}^{N-1} k_v g(u,v) \cos\left(\tfrac{(2n+1)v\pi}{2N} \right) \right\} \times \\
&\quad \times \cos\left(\tfrac{(2m+1)u\pi}{2M} \right).
\end{aligned}
\tag{2}
$$

DCT and IDCT can be easily extended for transforming a k-dimensional F hyper-rectangle and G hyper-rectangle.

The energy compaction property of DCT enables a storage-efficient k-dimensional joint distribution representation. For correlated data most of significant spectrum coefficients $g(u_{1i}, \dots, u_{ki})$ are concentrated near the coordinate system origin of the $U_1 \times \cdots \times U_k$ space. This property is presented in Fig. 1 for the 2-dimensional case. A sample PDF of distribution with 6 Gaussian clusters (Fig. 1a) and corresponding DCT-spectrum (Fig. 1b) were shown. Different grayscales of regions in Fig. 1b show grouped spectrum coefficients that absolute values are within intervals: $(+\infty, 10], (10, 1], (1, 0.1], (0.1, 0.01], (0.01, 0]$.

For correlated data many small absolute value coefficients could be omitted (zeroed) without a significant loss of accuracy of the data reconstructed from such reduced spectrum. There are many methods of assigning a region in the $U_1 \times \cdots \times U_k$ space for zeroing coefficients values. These methods are known as zonal sampling techniques [5] (e.g., rectangular, spherical, triangular, reciprocal). The best accuracy experimental results are achieved by applying the reciprocal zonal sampling defined below (from [5]):

$$Z = \left\{ (u_1, \ldots, u_k) \colon \prod_{j=1}^{k} (u_j + 1) \le b \right\}. \tag{3}$$

Parameter b determines a depth of cutting the spectrum region. Such lossy compressed spectrum will be denoted as $G^\wedge$. Shapes of sample spectrum regions shown in Fig.1b are approximately compliant with reciprocal zones.

3 DCT-Spectrum-Based Selectivity Estimation

The most important advantage of the approach in [5] is the capability of selectivity calculation directly from spectrum G (without a reconstruction of the histogram F).

This technique will be presented for the 2-dimensional space. Using definitions:

- X, Y – relation attributes with domains normalized to $[0, 1]$,
- Q – range query with selection condition: $a < X < b \wedge c < Y < d$,
- $X \times Y - [0, 1]^2$ space divided into $M \times N$ partitions by set of pairs (x_i, y_j):
  $x_i = \frac{2i+1}{2M}$, $y_i = \frac{2j+1}{2N}$, $i = 0, \ldots, M-1$, $j = 0, \ldots, N-1$,

distribution can by expressed using x_m, y_n in (2) instead of m, n:

$$f_{xy}(x_m, y_n) = f(m, n) =$$
$$= \sqrt{\tfrac{2}{M}} \textstyle\sum_{u=0}^{M-1} k_u \left\{ \sqrt{\tfrac{2}{N}} \sum_{v=0}^{N-1} k_v g(u, v) \cos(y_n v \pi) \right\} \cos(x_m u \pi) \tag{4}$$

and selectivity of query Q can be obtained as follows (from [5]):

$$sel = \int_c^d \int_a^b f_{xy}(x, y) \, \mathrm{d}x \mathrm{d}y =$$
$$= \int_a^b \sqrt{\tfrac{2}{M}} \textstyle\sum_{u=0}^{M-1} k_u \left\{ \int_c^d \sqrt{\tfrac{2}{N}} \sum_{v=0}^{N-1} k_v g(u, v) \cos(y v \pi) \, \mathrm{d}y \right\} \cos(x u \pi) \, \mathrm{d}x. \tag{5}$$

Finally, (4) and (5) enable to obtain the estimator of selectivity (from [5]):

$$sel^\wedge = \sqrt{\tfrac{2}{M}} \sqrt{\tfrac{2}{N}} \textstyle\sum_{(u,v) \in Z} k_u k_v g(u, v) \int_c^d \cos(v \pi y) \, \mathrm{d}y \int_a^b \cos(u \pi x) \, \mathrm{d}x \tag{6}$$

by using not all spectrum coefficients $g(u, v)$, but only these ones where $(u, v) \in Z$, according to assumed reciprocal zonal sampling:

$$Z = \{ (u, v) \colon (u+1)(v+1) \le b \}.$$

For validation of estimation accuracy after zoning, the formula of the relative error was assumed as follows:

$$ERR = \frac{|sel - sel^\wedge|}{sel} \times 100\%. \tag{7}$$

4   Implementing Selectivity Estimation in Oracle DBMS

ODCIStats (**O**racle **D**ata **C**artridge **I**nterface **Stat**istics) is a mechanism for extending the functionality of standard Oracle DBMS statistics [6]. It supports creating domain-specific user-defined extensions for the query optimizer module. Those extensions can be easily maintained by administrators using standard Oracle commands (e.g., ANALYZE TABLE, EXEC DBMS_STATS.gather_table_stats).

The use of DCT-statistics implementation is presented for a simple domain of points in a 2-dimensional space $[0, 1]^2$. Relevant software elements for this sample solution are shown below. The main functionality of DCT-statistics is implemented in a Java package, which is registered in the database catalog.

CreateZonedSpectrum Java method creates a 2-dimensional DCT-spectrum for *schema_name.table_name.col_name* attribute for given b (depth of spectrum cutting) and N (size of the $N \times N$ histogram). The spectrum representation is created in the operating memory and then persisted in a newly inserted table row (in a BLOB-type column). The identifier of the spectrum representation is returned.

```
int CreateZonedSpectrum (String schema_name,
                         String table_name,
                         String col_name, int b, int N)
```

CountSelectivity Java method retrieves the DCT-spectrum from database into operating memory using *id_stat* identifier and then returns a calculated selectivity for the query $Q(a < X < b \wedge c < Y < d)$ according to (6).

```
double CountSelectivity (int id_stat, double a, double b,
                         double c, double d)
```

PointType defined below is a domain composed type.

```
CREATE TYPE PointType AS OBJECT (x NUMBER, y NUMBER)
```

SomeTable is a domain table with a standard-type attribute (*id*) and a user-type attribute (*atr*). Values of *atr.x* and *atr.y* are correlated.

```
CREATE TABLE SomeTab (id NUMBER, atr PointType)
```

IncludePointChckFnc is a domain PL/SQL function which acts on user-type objects. It returns the value 1 if a point (*arg*) is placed inside a rectangle defined by a top-left corner (*tl*) and a bottom-right one (*br*).

```
CREATE FUNCTION IncludePointChkFnc
   (arg PointType, tl PointType, br PointType) RETURN NUMBER ...
```

PointDCTStatsType is a type which implements the ODCIStats interface. It is responsible for plugging the user-defined statistics functionality into DBMS.

```
CREATE TYPE PointDCTStatsType AS OBJECT (
   STATIC FUNCTION ODCIStatsCollect
   (col sys.ODCIColInfo, ...) ...
```

```
STATIC FUNCTION ODCIStatsSelectivity(..., sel OUT NUMBER,
    ..., arg PointType, top_left PointType,
    bottom_right PointType, ...) ...)
```

ODCIStatsCollect function creates a statistics for values of *col* table column by invoking *CreateZonedSpectrum*. *ODCIStatsSelectivity* function calculates the selectivity by invoking *CountSelectivity* and returns the selectivity value in *sel*.

After an association created by the command listed below, *PointDCTStatsType.-ODCIStatsSelectivity* function will be invoked for a selectivity estimation for any query with a selection condition based on *IncludePointFnc*.

```
ASSOCIATE STATISTICS WITH FUNCTIONS IncludePointFnc
    USING PointDCTStatsType
```

The command listed below associates *atr* attribute of *SomeTab* table with *PointDCTStatsType*. When ANALYZE TABLE command is executed *PointDCTStatsType.ODCIStatsCollect* function will be invoked for *atr*.

```
ASSOCIATE STATISTICS WITH TYPES PointType USING
    PointDCTStatsType
```

The next command creates a so-called database statistics for all attributes of *SomeTab*.

```
ANALYZE TABLE SomeTab COMPUTE STATISTICS
```

The following are created: a standard 1-dimensional equi-depth histogram for *id* attribute and a non-standard 2-dimensional DCT-spectrum for *atr* attribute.

Finally, for a sample range query $Q(0 < atr.x < 5 \land 1 < atr.y < 4)$

```
SELECT  *  FROM  SomeTab  WHERE
    IncludePointCheckFnc(atr, PointType(0,1), PointType(5,4))=1
```

the user-defined DCT-spectrum-based selectivity estimation function is used in the process of obtaining query execution plan.

The DCT-spectrum-based method will be used by the optimizer for any query with a selection condition based on *IncludePointCheckFnc*.

5   Tuning of DCT-Statistics Storage Parameters

The previous section shows the use of the implemented DCT-statistics. However, there were no hints for the database administrator what values of b (depth of the spectrum reciprocal zone) and N (size of the spectrum hypercube) should be used.

C shall denote a given total number of coefficients which are planned to store in a database catalog. The value of C is the cardinality of set Z (3). C determinates the space size required for storing the DCT-spectrum-based statistics. This section explains a proposed method for obtaining values of parameters b and N for given C using the criterion of the least mean selectivity estimation error (based on (6)). The idea behind this simple method will be explained for the 2-dimensional case.
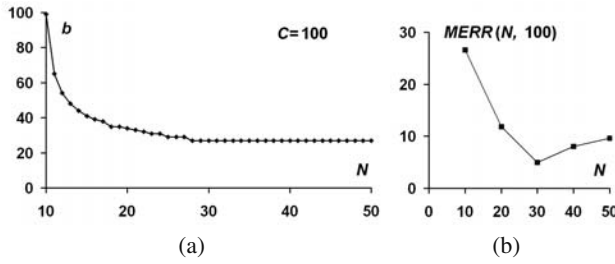
Fig. 2 **(a)** $b(N)$ – depth of spectrum cropping as a function of resolution for $C = 100$ – given the total number of coefficients. **(b)** Experimentally obtained $MERR(N, 100)$ – mean relative selectivity estimation error
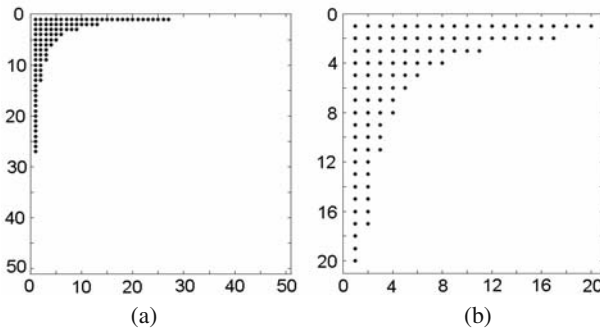
Fig. 3 Sampling zones for resolutions $N = 50$ **(a)** and $N = 20$ **(b)** where $C = 100$

Figure 2a shows $b(N)$ function for $C = 100$. This dependency between b and N is obtained from (3).

The distribution representation becomes more accurate when N (histogram resolution) becomes larger. Then histogram F better approximates PDF resulting in a better accuracy of the selectivity estimation based on corresponding full spectrum G. However, because of the low bandwidth filter property of DCT this rule may not be true for the cropped spectrum $G^{\wedge}$ (with the reciprocal zone and the predeterminated value of C). For example, which spectrum shown in Fig. 3 will give less selectivity estimation error? The problem can by generally formulated as higher histogram resolution versus stronger low bandwidth filter of zoning (the effect of concentration of the reciprocal zone near the spectrum-space origin).

The set of tests was implemented in Matlab for experimentally finding the best spectrum representation for the predefined storage size. Test returns optimal value of N for a given value of C and the least $MERR(N,C)$ – approximate mean selectivity estimation error.

The pseudocode of the algorithm for calculation $MERR(N,C)$ is presented in Fig. 4.

40 distributions with random numbers of different Gaussian cluster were used in line 03. 40 different query range pairs were used in line 06. The left range query

```
01 Set values for N and C
02 Obtain value of b for given N and C (see Fig. 2a for C=100)
03 For each distribution D from set of sample distributions:
04   Obtain frequencies matrix F for distribution D
05   Obtain compressed spectrum G^ for F
06   For randomly generated query range bounds (a, b, c, d):
07     Calculate estimated sel^ from G^
08     Calculate accurate sel directly from PDF of D
09     Obtain ERR value using sel and sel^ (see (7))
10 Calculate MERR by averaging all over ERR values
```

Fig. 4 Pseudocode of the algorithm for calculation $MERR(N,C)$

bound a was generated using uniform distribution $[0,1)$. The right query range bound b was generated from $(a,1]$ using a truncated exponent distribution to prefer smaller query ranges. The same method was used for c and d range bounds generation for the second dimension.

Figure 2b shows an experimental result – the numerically obtained $MERR(N,100)$ function. The optimal value of N for the least $MERR(N,100)$ is about 30.

The method of finding the error-optimal value of N for given C (based on the described algorithm) can be used for more than 2-dimensional case. Such obtained set of values (b and N for a predetermined C and number of dimensions) may be stored in the database catalog and used implicitly by the DBMS extension module (presented in the previous section) when the DCT-spectrum will be created.

The method is similar to the one presented in [5], but this method only concerns the reciprocal zone using.

6 Conclusions

The paper affects the problem of an accurate selectivity calculation for multi-attribute query selection conditions. For accuracy reason a representation of multidimensional PDF of attribute values is required.

The approach based on storage-efficient Discrete Cosine Transform representation [5] was implemented. The presented practical solution of selectivity estimation is fully integrated with Oracle DBMS. It is based on the Oracle ODCIStat interface for extending functionality of the cost-based query optimizer. Using user-defined DCT-based statistics extension is transparent for formulated queries. The paper shows the simplicity of using developed software elements.

The problem of obtaining an optimal resolution of the distribution representation for a predetermined number of DCT spectrum coefficients was discussed. The method of experimental finding error-optimal size of cropped spectrum was shown and applied.

Future work may concentrate on extending the method and implementation by distinguishing the approach for each distribution dimension (using frequency hyper-rectangle instead of hypercube). Each edge size of a multidimensional

histogram hyper-rectangle can be calculated separately using the criterion of the least AMISE (approximated mean integrated standard error [8]) for each equi-width 1-dimensional histogram of marginal distribution.

References

1. Bruno, N., Chaudhuri, S., Gravano, L.: STHoles: a multidimensional workload-aware histogram. In: Proceedings of ACM SIGMOD International Conference on Management of Data, New York, US, pp. 211–222 (2001)
2. Chakrabarti, K., Garofalakis, M., Rastogi, R., Shim, K.: Approximate query processing using wavelets. The Very Large DataBases Journal 10(2-3), 199–223 (2001)
3. Getoor, L., Taskar, B., Koller, D.: Selectivity estimation using probabilistic models. In: Proceedings of ACM SIGMOD International Conference on Management of Data, New York, US, pp. 461–472 (2001)
4. Gunopulos, D., Kollios, G., Tsortas, V.J., Domeniconi, C.: Selectivity estimator for multidimensional range queries over real attributes. The Very Large DataBases Journal 14(2), 137–154 (2005)
5. Lee, L., Deok-Hwan, K., Chin-Wan, C.: Multi-dimensional selectivity estimation using compressed histogram estimation information. In: Proceedings of ACM SIGMOD International Conference on Management of Data, Philadelphia, US, pp. 205–214 (1999)
6. Oracle: Oracle 10g. using extensible optimizer, http://download.oracle.com/docs/cd/B14117_01/appdev.101/b10800/dciextopt.htm
7. Possala, V., Ioannidis, Y.E.: Selectivity estimation without the attribute value independence assumption. In: Proceedings of the 23rd International Conference on Very Large Databases, Athens, Greece, pp. 486–495 (1997)
8. Scott, D.W., Sain, S.R.: Multi-dimensional Density Estimator. Handbook of Statistics, vol. 24. North-Holland Publishing Company, Amsterdam (2004)
9. Yan, F., Hou, W.C., Jiang, Z., Luo, C., Zhu, Q.: Selectivity estimation of range queries based on data density approximation via cosine series. Data & Knowledge Engineering 63(3), 855–878 (2007)

How to Efficiently Generate
PNR Representation of a Qualitative Geofield

Piotr Bajerski

Abstract. In the paper, geographic phenomena varying continuously over geographical space (e.g., air pollution) are called geofields and queries concerning them are called geofield queries. Theoretical and experimental research showed that usually the most time consuming operation in geofield queries processing is generating the PNR representation of qualitative geofields. As the PNR representation is based on ordering a quadtree by Peano N space-filling curve, a central role in an implementation of the operator is played by square classifiers making decisions whether to assign classified quadrants to one of the intervals given in the query or split them and classify their children. Presented classifiers make the decision computing geofields model values in predefined points or in points chosen using a geofield variability model. The variability is predicted using conditional quantile functions, modeling dependence of change of geofield value module between points on the distance between the points.

Keywords: spatial databases, GIS, query optimization.

1 Introduction

In the paper, geographic phenomena varying continuously over geographical space (e.g., air pollution, temperature or rainfall) are called geofields and are divided into quantitative geofields and qualitative geofields. A *quantitative geofield* is a two-dimensional function assigning elements of its domain values measured on interval or ratio scale (so spatial interpolation is possible). A *qualitative geofield* is a function assigning elements of its domain values measured on nominal or ordinal scale, e.g. numbers of intervals of a geofield values. Queries concerning geofields are called *geofield queries*. In the research, it was assumed that the user is interested in intervals of quantitative geofields values. Specifying these intervals in geofields queries converts quantitative geofields into quantitative geofields [8].

Piotr Bajerski
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-101 Gliwice, Poland
e-mail: `piotr.bajerski@polsl.pl`

Interval	LBound	UBound	Shape	
			KPmin	KPmax
0	$> -\infty$	≤ 110	0	3
			KPmin	KPmax
1	> 110	≤ 165	4	7
			8	11
			KPmin	KPmax
2	> 165	≤ 220	12	15
			KPmin	KPmax
3	> 220	$< \infty$		

(a) (b)
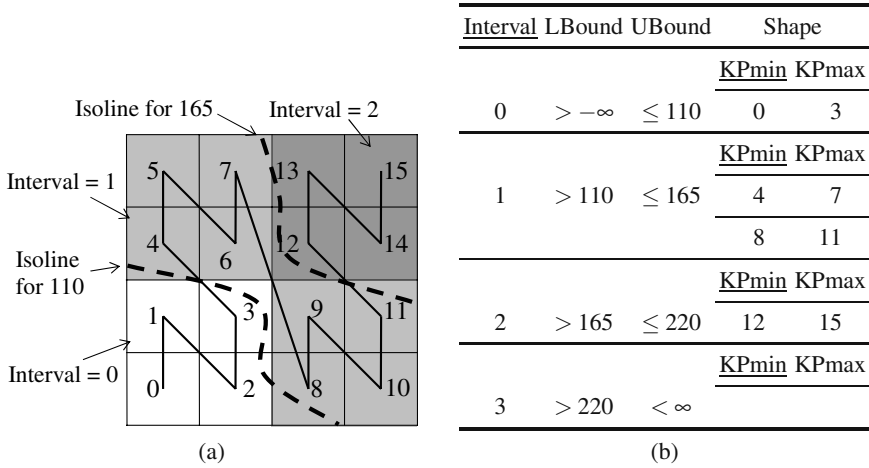
Fig. 1 Discrete representation of a hypothetical geofield used in examples and its PNR representation. (a) Discrete representation of a geofield. (b) Relation GF storing PNR representation of the geofield from a

In [3] a computational model for efficient processing of geofield queries is described. Its main parts are constituted by: (a) a discrete representation of the geographic space, which elements are ordered by N Peano space-filling curve, (b) discrete representation of geofields and geoobjects as sequences of N Peano keys stored in Peano relations (called PNR representation), and (c) set of operators for processing geofields and geoobjects in the PNR representation. In the research, it was assumed that quantitative geofields are represented in a database as collections of values measured in irregular measurement networks. Therefore, to answer geofield queries we need a procedure to estimate geofields values in points in which they were not measured [6]. Such a procedure, based on interpolation, is called *geofield (mathematical) model*. This paper accompanies [3] by describing implementation of the generation of the PNR representation of a qualitative geofield, denoted by η. In GFQL queries usually geographical coordinates are used. To ease the computations during processing queries geographical coordinates are mapped to *discrete coordinates* in which as units sides lengths of the elementary quadrants are used [3]. The area in which values of the given geofield must be computed to answer the query is called *geofield context region*.

Theoretical and experimental research has shown that usually the most time consuming operations in geofield queries processing is usage of the operator η [2, 1], which generates a PNR representation of a qualitative geofield using point measurements of the geofield, its mathematical model, and interval boundaries. The PNR representation of a geofield consists of the PNR representations of zones in which geofield model values belong to the same intervals (Fig. 1).

Let us consider the following example of a geofield query: *Generate PNR representation of average-yearly distribution of airborne suspended matter over all*

counties stored in the table `Counties` using intervals 110, 165, 220 $\mu g/m^3$. *The answer is to be shown as a raster with resolution 2000 pixels.* In [1] there was presented an extension of SQL, called GeoField Query Language (GFQL), facilitating writing geofield queries. Assuming that point measurements for the geofield of airborne suspended matter are stored in the relation `M_SM_08` and its mathematical model is given by Ordinary Kriging with semivariogram stored in metadata with ID = 71 and with choosing independently six nearest nodes in four sectors but not further than 11 km, the query may be written in GFQL as:

```
CREATE GEOFIELD FROM M_SM_08
WHERE SquareClassifier = 'BPO' And Beta = 0.8 And
      Interpolation.Method = 'OK' And
      Interpolation.Semivariogram.ID = 71 And
      NodeSearch.Type = 'Quads' And
      NodeSearch.N = 6 And NodeSearch.R = 11 And
      Intervals = (110, 165, 220) And
      ContextRegion = (SELECT Boundary FROM Counties) And
      Resolution = 2000;
```

The command `Create Geofield` may be used on its own or may be used in the Select command to create the PNR representation of a geofield, which is seen in the query as a view with predefined attributes. The phrase `Intervals` allows for defining intervals of geofields values. Computations are to be carried out in the resolution determined by the condition on the pseudo-attribute `Resolution`. Figure 1 shows PNR representation of a hypothetical geofield created by the example query.

2 Quadrant Classifier Concept

In the research, it was assumed that a user is expecting an answer as if the asked query was evaluated using a raster representation. Such an answer is called a *referential answer*. The difference between referential answer and the answer obtained using presented method is called *distortion*. It is assumed that in both cases the same geofield model is used. To settle a compromise between speedup of queries execution and distortion of their answers a *parameter β* was introduced.

There are two general approaches to implementation of the operator η: bottom-up and top-down. The first creates raster representation of the geofield, which in turn is converted to PNR representation. Implementation of this solution is simple but it can yield hardly any speedup to geofield query execution. In the top-down approach, a PNR representation of the context region is created and there is an attempt to classify as large as possible squares belonging to this representation. The main problem of this approach is to efficiently decide whether a given quadrant may be classified to one of the intervals of the geofield values given in the query or should be split and the classification should be applied to its children.

For qualitative geofields a special marker, called unknown geofield value (UGV), was introduced to indicate that it has been impossible to determine the geofield value in the given point. For example, there may be not enough measurement network

nodes (MNN) chosen in local interpolation or the point may lie outside the domain of the geofield. From now on $MNN(q)$ will denote measurement network nodes which lay in the quadrant q.

A quadrant classifier is a decision function which domain consists of quadrants that can belong to a PNR representation and codomain contains interval indexes and two special values: SPLIT and UGV. The set of values used by a classifier to make a decision how to classify a given quadrant q is called *classifier decision set* and denoted by $W(q)$. No other research on such classifiers has been found.

All classifiers first check values in $MNN(q)$. The decision SPLIT is taken when first difference in intervals is found or there is UGV marker. Elementary quadrants are always classified as only one value is computed for them. It is assumed that an elementary quadrant can contain at most one MNN, which restricts the minimal resolution that can be used in geofield queries.

The formula $\widehat{\mathscr{G}_I}(x,y)$ denotes value of the geofield model in the point (x,y) in the continuous computational space, while $\widehat{\mathscr{G}_I^{\mathbb{D}}}(x,y)$ denotes value of the geofield model in discrete computational space. If the elementary quadrant with coordinates (x,y) contains a measured value, the value is returned, otherwise it is assumed that $\widehat{\mathscr{G}_I^{\mathbb{D}}}(x,y) \equiv \widehat{\mathscr{G}_I}(\lfloor x \rfloor + 0.5, \lfloor y \rfloor + 0.5)$.

3   Geofield Value Change Prediction

Spatial interpolation is based on the assumption that there is a positive spatial dependence in the data [4, 5, 6]. It means that if we know value, v_0, in a given point, s_0, values in the neighbourhood of s_0 are similar to v_0. The nearer a given point, s_i, is to s_0, the more similar are the values in s_i and s_0. As the distance increases the similarity disappears and from some distance, h_g, there is no correlation between the values, which means that the spatial dependency no longer holds. Positive spatial dependency is fundamental to spatial interpolation so the main idea behind geofield value change prediction was to use it also to predict the distance on which geofield model values will not change a given interval.

To predict geofield value change there are needed two functions $\widehat{\mathscr{G}_I^-}(\cdot)$ and $\widehat{\mathscr{G}_I^+}(\cdot)$ with the following property:

$$\underset{s \in \mathscr{R}_{GC}}{\forall} \quad \underset{s_1 \in [s, s+h] \subset \mathscr{R}_{GC}}{\forall} \widehat{\mathscr{G}_I^-}\left(\widehat{\mathscr{G}_I}(s), h\right) \leq \widehat{\mathscr{G}_I}(s_1) \leq \widehat{\mathscr{G}_I^+}\left(\widehat{\mathscr{G}_I}(s), h\right), \quad 0 \leq h \leq h_g,$$

which tells that on the segment determined by points s and $s+h$ geofield model values belong to $\left[\widehat{\mathscr{G}_I^-}\left(\widehat{\mathscr{G}_I}(s), h\right), \widehat{\mathscr{G}_I^+}\left(\widehat{\mathscr{G}_I}(s), h\right)\right]$ and this does not depend on the location in the geofield context region $\mathscr{R}_{GC}$. To enable settling a compromise between speedup of queries execution and distortion of their answers the parameter β was introduced so functions $\widehat{\mathscr{G}_I^-}\left(\widehat{\mathscr{G}_I}(s), h, \beta\right)$ and $\widehat{\mathscr{G}_I^+}\left(\widehat{\mathscr{G}_I}(s), h, \beta\right)$ were looked for. For this

function the parameter β controls how often the predicted value does not belong to the interval $\left[\widehat{\mathscr{G}_I^-}\left(\widehat{\mathscr{G}_I}(\boldsymbol{s}),\boldsymbol{h},\beta\right),\widehat{\mathscr{G}_I^+}\left(\widehat{\mathscr{G}_I}(\boldsymbol{s}),\boldsymbol{h},\beta\right)\right]$. As β increases the frequency that $\widehat{\mathscr{G}_I}(\boldsymbol{s}+\boldsymbol{h})$ does not belong to the interval decreases.

In geostatistics a quantitative geofield is modelled as a realization of a random field $\boldsymbol{Z}=\left\{Z(\boldsymbol{s}):\boldsymbol{s}\in D\subset\mathbb{R}^2\right\}$ [4, 6]. Using this approach values are interpolated using a Kriging (in the experiments Ordinary Kriging was deployed) which needs a semivariogram defined as $\gamma(\mathbf{s}_1-\mathbf{s}_2)\equiv 0.5\mathrm{var}(Z(\mathbf{s}_1)-Z(\mathbf{s}_2))$. The semivariogram models the structure of the spatial dependency in the geofield. A classical estimator of a semivariogram is given by $\widehat{\gamma}(\boldsymbol{h})=\frac{1}{2|N(\boldsymbol{h})|}\sum_{N(\boldsymbol{h})}\left(Z(\boldsymbol{s}_i)-Z(\boldsymbol{s}_j)\right)^2$, where $\{Z(\boldsymbol{s}_i):i=1,\ldots,n\}$ is the set of the measured values of the geofield and $N(\boldsymbol{h})$ is the set of measurement nodes separated by $\boldsymbol{h}$ (with some tolerance in practise).

The basic idea used for geofield value change prediction (GVCP) was to exploit modules of differences between measured geofield values separated by distance $\boldsymbol{h}$. It was assumed that sets of such differences may be treated as realization of random variables, $U(\boldsymbol{h})\sim U(\boldsymbol{s}_1,\boldsymbol{s}_2)=|Z(\boldsymbol{s}_2)-Z(\boldsymbol{s}_1)|$, where $\boldsymbol{h}=\boldsymbol{s}_2-\boldsymbol{s}_1$. It was further assumed that these variables are stochastically ordered, so $\forall_{p\in(0,1)}0<\boldsymbol{h}_1<\boldsymbol{h}_2\leq\boldsymbol{h}_g\Rightarrow Q_U(p|\boldsymbol{h}_1)<Q_U(p|\boldsymbol{h}_2)$, where $Q_U(p|\boldsymbol{h})$ denotes a conditional quantile function or order p. Using these assumptions functions $\widehat{\mathscr{G}_I^-}\left(\widehat{\mathscr{G}_I}(\boldsymbol{s}),\boldsymbol{h},\beta\right)$ and $\widehat{\mathscr{G}_I^+}\left(\widehat{\mathscr{G}_I}(\boldsymbol{s}),\boldsymbol{h},\beta\right)$ were constructed as:

$$\widehat{\mathscr{G}_{\mathscr{I}}^-}\left(\widehat{\mathscr{G}_{\mathscr{I}}}(\boldsymbol{s}),\boldsymbol{h},\beta\right)=\widehat{\mathscr{G}_{\mathscr{I}}}(\boldsymbol{s})-\widehat{\Delta\mathscr{G}_{\mathscr{I}}}(\boldsymbol{h},\beta),$$
$$\widehat{\mathscr{G}_{\mathscr{I}}^+}\left(\widehat{\mathscr{G}_{\mathscr{I}}}(\boldsymbol{s}),\boldsymbol{h},\beta\right)=\widehat{\mathscr{G}_{\mathscr{I}}}(\boldsymbol{s})+\widehat{\Delta\mathscr{G}_{\mathscr{I}}}(\boldsymbol{h},\beta),$$

where as $\widehat{\Delta\mathscr{G}_{\mathscr{I}}}(\boldsymbol{h},\beta)$ a quantile regression curve of order β was used, which limited β values to the interval $(0,1)$. In the research, it was assumed that the same function family is used for quantile regression curves as for the semivariogram model used in Kriging. In the experiments, exponential functions were used. Quantile regression curves were fitted using quantile regression [7] in R CRANE with the parameter β used as the rank.

4 Quadrant Classifiers

This section shortly describes the most important classifiers worked out during research [1]. The classified quadrant will be denoted by q and its centre by $q.c$. The term *working interval* will denote an interval to which the first element of the decision set belongs.

Classifier $\mathscr{F}_{S4}$

Decision set of the classifier $\mathscr{F}_{S4}$ consists of $MNN(q)$, the value interpolated for the centre of q, and four values interpolated for the centres of the children of q.

Classifier $\mathscr{F}_{\mathrm{B}}$

During interpolation it is very rare that the interpolated values are outside the range of values in $MNN(q)$. On this observation the classifier $\mathscr{F}_{\mathrm{B}}$ is based, which decision set consists of $MNN(q)$ and all elementary boundary quadrants of q.

Classifier $\mathscr{F}_{\mathrm{BO}}$

Classification of the quadrants forming PNR representation of a geofield context region is ordered by N-Peano space filling curve. Therefore, if q has south, west and south-west neighbours, they have already been classified. This information may be used to skip interpolation for such boundary quadrants that have neighbours classified to working interval. The classifier using this optimization was named $\mathscr{F}_{\mathrm{BO}}$.

Classifier $\mathscr{F}_{\mathrm{SP1}}$

Decision set of the classifier $\mathscr{F}_{\mathrm{SP1}}$ consists of $MNN(q)$ and the lowest and the highest geofield values predicted for q. The prediction is based on the value interpolated for the centre of the q and the prediction of the maximal geofield model value change on the distance equal to the half of the diagonal, h_{d2}, of q using the GVCP model given in the query. The decision set for the classifier $\mathscr{F}_{\mathrm{SP1}}$ may be written as:

$$W_{\mathrm{SP1}}(q) = MNN(q) \cup \left\{ \widehat{\mathscr{G}_I^-}\left(\widehat{\mathscr{G}_I}(q.c), h_{d2}, \beta\right), \widehat{\mathscr{G}_I^+}\left(\widehat{\mathscr{G}_I}(q.c), h_{d2}, \beta\right) \right\}.$$

For example, let us assume that q has side length 8, intervals bounds are 110 and 165, and $\beta = 0.9$. (All geofield values are in $\mu g/m^3$.) Let us assume further that $MNN(q) = \{136\}$, the value interpolated in the centre of q equals 140 ($\widehat{\mathscr{G}_I}(q.c) = 140$), and the GVCP model is given by $\widehat{\Delta\mathscr{G}_I}(h, \beta)$, such that $\widehat{\Delta\mathscr{G}_I}(4\sqrt{2}, 0.9) = 15$. Than $W_{\mathrm{SP1}}(q) = \{136, 140 - 15, 140 + 15\}$. As all these values fall into the interval with index 1 with intervals $(110, 165]$, therefore q will be classified to interval with index 1.

Classifier $\mathscr{F}_{\mathrm{BPO}}$

The classifier $\mathscr{F}_{\mathrm{BPO}}$ an extension of the classifier $\mathscr{F}_{\mathrm{BO}}$ using GVCP for choosing elementary boundary quadrants for which geofield values will be interpolated. The idea of using GVCP for checking elementary quadrants forming north boundary of q with side length 8, located in left bottom corner of the context region ($KPmin = 0$) is shown in Fig. 2. The GVCP model given as before by $\widehat{\Delta\mathscr{G}_I}(h, \beta)$, $\beta = 0.9$, and limit of positive spatial dependency in the GVCP model, h_g is limited to 4 units. Let us assume that $MNN(q)$ is empty, so the classification starts from interpolating geofield model value for the q_e making the right upper of q with discrete computational coordinates $(7,7)$. This value $v_1 = \widehat{\mathscr{G}_I^{\mathbb{D}}}(7,7)$. Let us assume that $\mathscr{P}(v_1) = 1$, which
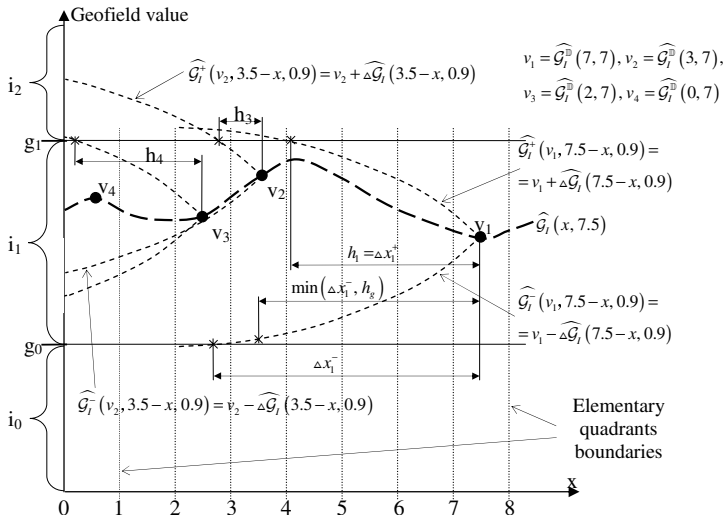
Fig. 2 Using geofield value change prediction in the classifier $\mathscr{F}_{\text{BPO}}$

means that it belongs to the interval with index 1. Next GVCP is used to find the maximal distance on which geofield model will not change interval. From the GVCP model yields the value 3.3, so the elementary quadrants $(6,7)$, $(5,7)$, and $(4,7)$ can be classified to interval 1 without further computations and the next interpolation will occur for the elementary quadrant with coordinates $(3,7)$. This will repeat until the end of the north boundary is reached.

5 Experimental Results

This section presents some results of the experimental evaluation of the presented quadrant classifiers. In the experiments, ambient air pollutant data from Upper Silesia gathered by District Sanitary-Epidemiological Station (SANEPID), Katowice, Poland was used. The experiments were run on PC with Pentium 4 2.8 GHz and 1 GB RAM.

Figure 3 presents speedup and distortion coefficient for the classifiers from Sect. 4. The distortion coefficient is the ratio between the number of elementary quadrants classified to different intervals in the answer computed using PNR representation than in the referential answer and the number of all classified elementary quadrants. For classifiers $\mathscr{F}_{\text{B}}$ and $\mathscr{F}_{\text{BO}}$ distortion coefficient is not shown in the chart as its value was below 10^{-6}. Speedup denotes how many times faster PNR representation of a geofield is created than its raster representation using the same geofield model and context region. Generation of a raster representation of a geofield in resolution 2000 took about 40 s.
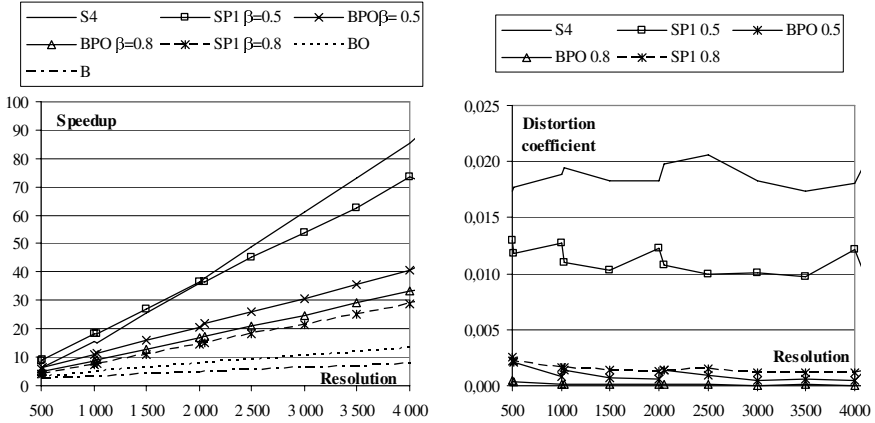
Fig. 3 Speedup and distortion coefficient for the presented classifiers

Figure 4 presents speedup and distortion coefficient for the classifier $\mathscr{F}_{\text{BPO}}$ for different values of the parameter β. Distortion coefficient for $\beta > 0.8$ not shown because the values were too small.

In the experiments, speedup depended on the number and location of the bounds of geofields values intervals. In the presented results four intervals deployed in practice by SANEPID were used. Generally, when more intervals is used the speedup drops and for less or wider intervals the speedup increases, especially when
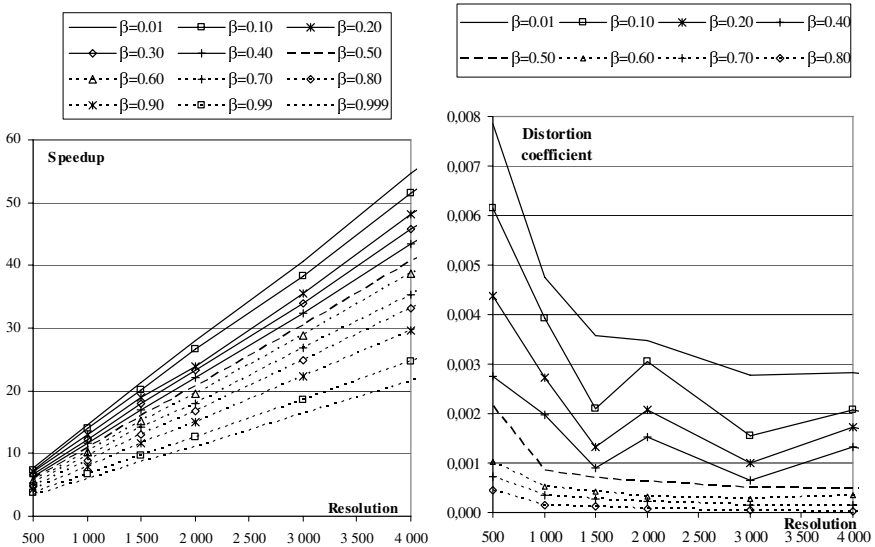
Fig. 4 Speedup and distortion coefficient for the classifier $\mathscr{F}_{\text{BPO}}$

classifier $\mathscr{F}_{BPO}$ is used. Speedup also depends on the geofield context area – is higher for simpler areas and lower for areas with complicated boundaries. The results shown are for context region determined by Upper Silesia, which has a rather complicated boundary. However, such context regions are to be expected in practice, especially when geofield queries optimization rules are deployed [2].

6   Conclusions

During research many classifiers were devised. They differed in the compromise between speedup and distortion. The best of the classifiers turn out to be the classifier $\mathscr{F}_{BPO}$ basing its decisions on boundary elementary quadrants chosen using prediction of geofield variability. Using the parameter β as the rank of quantile regression curve enables to efficiently predict geofield model changes. For the classifier $\mathscr{F}_{BPO}$ small changes of the parameter β gave small changes in speedup and distortion coefficient. Even for small values of β the PNR representation was satisfactory. When there is no geofield model variability and high responsiveness is needed, the classifier $\mathscr{F}_{S4}$ is a good choice. However its usage is connected with introducing noticeable distortion. When the answer must be very similar to raster representation the classifier $\mathscr{F}_{BO}$ is the best choice.

References

1. Bajerski, P.: Using Peano algebra for optimization of domain data sets access during analysis of features distribution in space. Ph.D. thesis, Silesian University of Technology, Gliwice, Poland (2006) (in Polish)
2. Bajerski, P.: Optimization of geofield queries. In: Proceedings of the 1st International Conference on Information Technology, Gdansk, Poland (2008)
3. Bajerski, P., Kozielski, S.: Computational model for efficient processing of geofield queries. In: Proceedings of the International Conference on Man-Machine Interactions, Kocierz, Poland (2009)
4. Cressie, N.: Statistics for Spatial Data. John Wiley & Sons, Chichester (1995)
5. Diggle, P., Ribeiro Jr., P.: Model-based Geostatistics. Springer Science+Business Media, LLC, Heidelberg (2007)
6. Haining, R.: Spatial Data Analysis: Theory and Practice. Cambridge University Press, Cambridge (2003)
7. Koenker, R.: Quantile Regression. Cambridge University Press, Cambridge (2005)
8. Shekhar, S., Chawla, S.: Spatial Databases: A Tour. Prentice Hall, Englewood Cliffs (2003)

RBTAT: Red-Black Table Aggregate Tree

Marcin Gorawski, Sławomir Bańkowski, and Michał Gorawski

Abstract. This paper presents a new spatio-temporal index – Red-Black Table Aggregate Tree (RBTAT). The RBTAT's algorithms were implemented in a Grid Data Warehouse system supported by Software Agents. The carried out experiments focused mainly on RBTAT structures loading time and responses to range queries.

Keywords: data warehouse, spatio-temporal index, query, red-black tree indexes.

1 Introduction

The main function of a data warehouse system is, apart from data storing, searching of adequate information realized by range queries. In the spatio-temporal data warehouse systems the user is interested not in single but rather in aggregated data. To optimize responses to the range spatio-temporal queries, various indexing techniques were implemented [9, 2]. The Spatial Telemetric Data Warehouse System [7] supported by Software Agents [6, 8, 3] process has multidimensional character thus need special multidimensional indexing structures [5]. The first index we tested in the MVB-tree index [1], which fully supported temporal queries, while spatial queries were processed in a database. To upgrade this structure to fully process multidimensional query, MVB-tree was merged with the R-tree which supports spatial query [5]. This new hybrid index that efficiently manages spatio-temporal queries was denoted as the R-MVB tree [4]. The R-MVB index greatly decreased query response time but certain drawbacks were also observed -significant usage of a HDD space and index building time. These problems mainly originated from characteristics of the MVB-tree, so a new approach was taken. As an effect new indices emerged – the RBTAT and the RBTAT-MT. These indices solve above mentioned

Marcin Gorawski · Sławomir Bańkowski · Michał Gorawski
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail:{Marcin.Gorawski,Slawomir.Bankowski,
Michal.Gorawski}@polsl.pl

problems and effectively support spatio-temporal queries. In the following part of the article the RBTAT index will be presented along with tests performed on this structure.

2 Red-Black Table Aggregate Tree (RBTAT)

The RBTAT structure (Red-Black Table Aggregate Tree) was created as an alternative to above mentioned MVB-tree, and to eliminate R-MVB structures drawback. The RBTAT index is a compilation of four structures – the red-black tree, singly linked list, set of tables and free access point. RBTAT characterizes with significant decrease of a building time and HDD space consumption in comparison with the R-MVB-tree. The main drawback is the 3–4 times increase of a query response time in comparison to the R-MVB-tree, however it's average response time is near 200–400 ms so for the RBTAT it will be around 600–1600 ms – this is still satisfactory value. Also the modifications of the RBTAT were implemented – The RBTAT-MEM and RBTAT-MT. The RBTAT-MEM upgrades structure buffering abilities and changes HDD writing method. The RBTAT-MT (RBTAT Multi Thread) enables effective support for the multi core processors and ensures full usage of such systems.

2.1 Building the RBTAT Structure

The RBTAT is implemented in the module architecture. Each module is a adequate structure that can be combined with other modules parts. The presented structure is database independent, all data is stored in a binary file on a HDD. The first module is a Red-Black tree (Fig. 1). In this balanced tree the sensor ID is set as a main key, and the object stored in a structure is a node represented by the RBtatRoot class object. The tree does not contain any data about sensor (counter) localization, as in R-MVB tree the R-tree manages spatial queries and is responsible for feeding the structure with data. After obtaining data from the R-tree (counter ID), finding the adequate node in a RBTAT tree is quite fast, because the structure is balanced. This means that there cannot be a situation when the tree height is significantly different for various leafs. The highest tree cannot be greater than a logarithm of a number of leafs on the lowest level increased by one to the base 2.
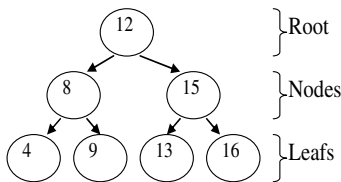
Fig. 1 Structure of RB tree module
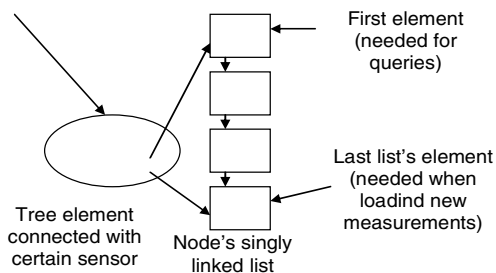
Fig. 2 Singly linked list

The second module is a singly linked list, different for each node (Fig. 2). The node has a pointer to the list's first element, and a list's element is represented with a RBtatList class. The first and the last element is needed to search the list and in the ETL process. Each list's element stores the table of pointers to the measurements table and the measurements aggregates far all tables and each table separately. The table size can be configured and depends of several factors (number of measurements in a node, size of the ETL buffers) – this parameter is set dynamically basing on experiments results. The node for a certain sensor has references to the first and the last element. The first element is needed when processing queries and the last element is needed during the ETL process when sensor's new measurements are added. Typically there are several singly linked list elements in each node. The value is basically 2–10. The last module consists of one dimensional table, each element has several tables. For each table we calculate measurements aggregates and store it in element's list, each table consists of measures from the same sensor. The number of elements of a single table can be configured, there is a possibility to have tables of different sizes for the same sensor.

All tables, except of the one for each last element are entirely filled. The information about all tables (Fig. 3) address, max number of elements, in brackets: min and max measurement, measurements sum is stored in every element. In Fig. 3 the first table has address 2380, it's size equals 4 elements, minimal measurement equals 2 and maximal measurement is 4, Measurements sum equals 12. There are also additional information in the element such as factor summing up all tables (the upper part of the element); minimal and maximal measurement, measurement's sum and quantity.

All tables are filled entirely (except the last table), so there is no need to store information about actual number of lists in tables for all elements. Information about the last table is stored in a tree element for certain sensor. Such approach enables us to save space in sensor list's elements, moreover it ensures fast access to the last table witch is simultaneously active while saving new elements. In a following example, on a Fig. 3a. in a sensor tree's element (ellipse) the table address is 2394, and a number of stored elements equals 2. The next element should be written on address $2394 + 2 + 1$. This is a virtual address, the physical address is calculated basing on a number of bytes needed to store one measurement. Example: is there
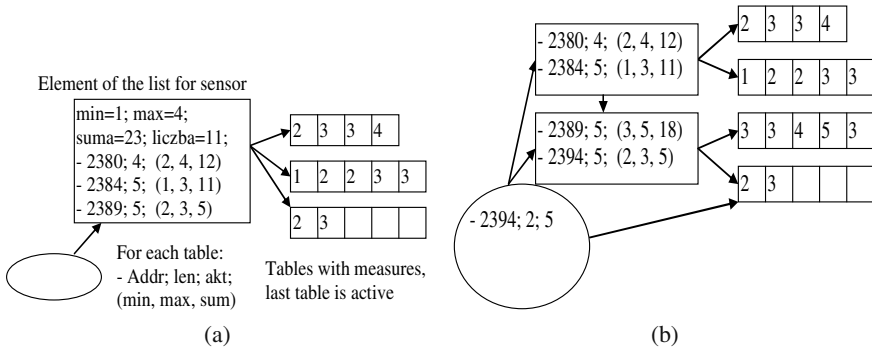
Fig. 3 Examples
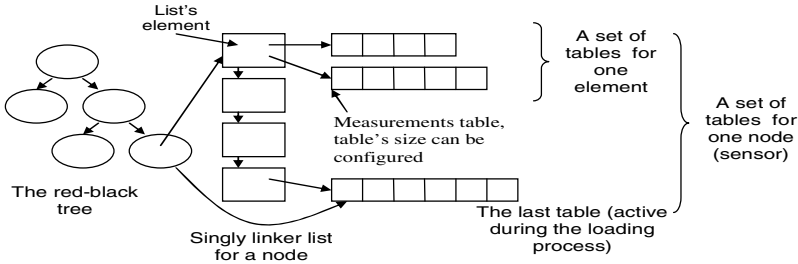
Fig. 4 Structure of RBTAT index

is 8 B needed for one measurement then a file address will be $(2394 + 2 + 1) \times 8$. So to obtain a table address and a table's index where a new measurement should be placed we only have to find an element in a RB-tree and perform some add and subtracts operations. The whole RBTAT structure is presented in Fig. 4.

2.2 Using a Multicore Architecture (Full Support for a Multicore Processors)

During RBTAT tests on a double core processor we observed only 50% CPU usage. The RBTAT used only one CPU's core, to increase the performance we propose a solution in which several separate applications simultaneously builds a common structure.

The dividing application assigns sensors to inserters; moreover a hashing table is created so finding adequate inserter for next measurements is quite fast (Fig. 5). In case of a multicore architecture there is a considerable speedup during the ETL process – each core can process separate procedures connected witch data loading. In case of queries the crucial part is a HDD access, calculations are not problematic,
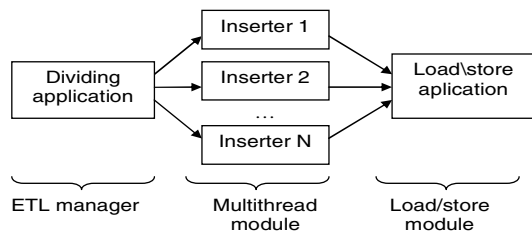
Fig. 5 Use of multi-core processor

data is already processed and only new data is added to stored values. Analogically like in a multicore architecture case we can divide the RBTAT between several computers. In case of such distribution, increased performance can be observed not only during the ETL process but also during query processing. The only boundary during the ETL process is a connection throughput that should be able to manage quick data loading to several structures simultaneously.

3 RBTAT Tests

The tests were performed using a Core2Duo E5500 PC with 2GB RAM. The telemetric data used three dimensions (X, Y, Z – a sensor position in space), measurement time (as a number of milliseconds since 1970), and a sensor type (gas counter, water counter, electricity counter). The query is presented as a data cube in a certain time boundaries. Experiments show a comparison of RBTAT's performance with a database and other multidimensional indices (e.g., the MVB-tree).

3.1 Structures Creation

Data is loaded from a test file to certain structures : DB – Oracle 9.2i Data Base, MVB, STCAT, MDPAS, RBTAT.

These tests (Figs. 6, 7) shows that a loading time and structures size does not differ from a data base results (Oracle 9.2i data base).

3.2 RBTAT Selectivity

The selectivity is calculated as a following factor: The Selectivity is a quotient of a average number of bytes acquired from a HDD during query, to a total number of bytes stored on a HDD. The selectivity around 0.1% means that for each 1 GB of a stored structure, 1 MB has to be acquired.
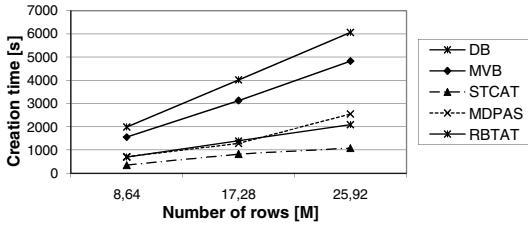
Fig. 6 Number of measures influence on time of creating different structures
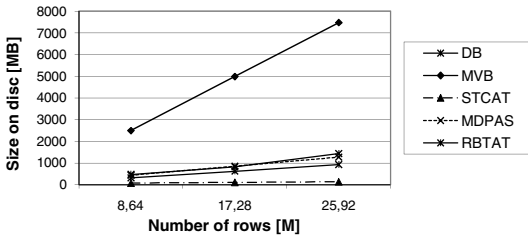
Fig. 7 Number of measures influence on size of structures
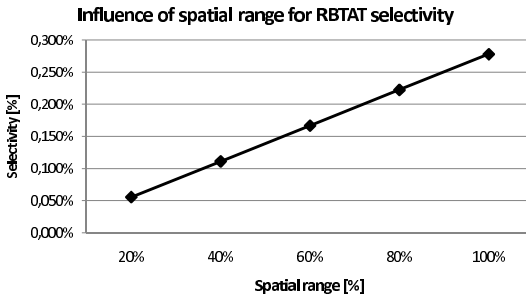
Fig. 8 Spatial range influence on selectivity of RBTAT

The RBTAT structure Does not possess any aggregation on a space level (aggregation of several sensors), so the selectivity increased directly proportional to a space range (sensors are regularly placed in a tested space) (Fig. 8). The next test (Fig. 9) considered queries in an environment with a variable size of a list's elements buffer. The test was performed for 10 000 sensors, each sensor generated around 53 000 measurements, what gave us total 525 000 000 measurements, and the structure used around 17 GB of a HDD space.

We can observe that along with a buffer size increase selectivity decreased proportionally. Above a value of 25.33 MB selectivity remain unchanged, which means that all elements were loaded into memory and further buffer size
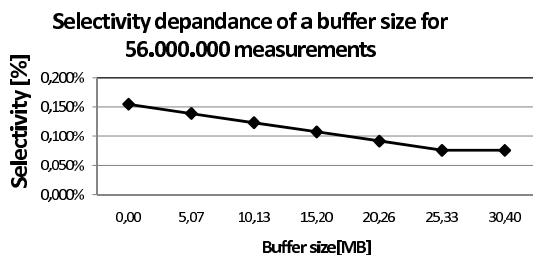
Fig. 9 Buffer size influence on selectivity for 56 millions measures
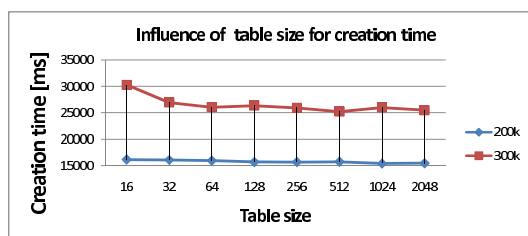
Fig. 10 Table size influence of loading time

increase is pointless. The appropriate use of a list buffer can significantly decrease selectivity.

3.3 *Loading Time in Dependance of a Measurement Tables Size*

The appropriate size of the measurement tables is crucial for a data loading performance.

As presented in Fig. 10, the more data we have, the less influence of a table size. The lading time changes only for values 16–512.

3.4 *RBTAT and a Multicore CPU*

The experiment used a double core processor (Fig. 11) and tested a data loading time.

We can observe that even using only one inserter the loading time is significantly decreased while using the RBTAT-MT modification. This is caused because on a server in one thread data is retrieved from a HDD and in a inserter application loaded to a structure. The average processor usage is around 100% (not 50% like in a standard RBTAT). The increase of a inserters quantity does not influence loading speed on a double core processor, however on a quad core processor speedup will be noticeable.
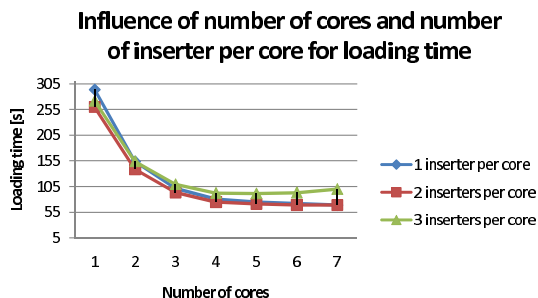
Fig. 11 Loading data using multi-core processor

4 Conclusions

All mentioned indices are implemented in Java (1.5) and database independent – all information is stored in a binary file and all additional database engines are expendable. The presented RBTAT index is designated mainly for the sensor data, in a case when there are not many sensors but many measurements from a single sensor. The RBTAT ensures fast responses to aggregational range queries and enables a change of focus on an ETL performance phase or a query processing phase, depending on an allocated memory. The support for multicore CPU's is a response to rapid evolution of a multicore PC's; moreover it can be easily used in a sensor system distribution between several PC's. The RBTAT performance in case of data loading is almost identical to databases performance, but greatly exceeds database systems in case of responses to range queries. Future works concerns further upgrades of presented indices, spatial performance measuring, and a balanced data division in indexing structures.

References

1. Becker, B., Gschwind, S., Ohler, T., Seeger, B., Widmayer, P.: An asymptotically optimal multiversion B-tree. The Very Large DataBases Journal 5(4), 264–275 (1996)
2. Beckerman, N., Kriegel, H.P., Schneider, R., Seeger, B.: The R*-tree: An efficient and robust access method for points and rectangles. In: Proceedings of SIGMOD International Conference on Management of Data, pp. 322–331 (1990)
3. Cao, J., Spooner, D.P., Jarvis, S.A., Saini, S., Nudd, G.R.: Agent-based grid load balancing using performance-driven task scheduling. In: Proceedings of the 17th International Symposium on Parallel and Distributed Processing, p. 49 (2003)
4. Gorawski, M., Gorawski, M.J.: Modified R-MVB tree and BTV algorithm used in a distributed spatio-temporal data warehouse. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) PPAM 2007. LNCS, vol. 4967, pp. 199–208. Springer, Heidelberg (2008)
5. Gorawski, M., Gorawski, M.J.: Balanced spatio-temporal data warehouse with R MVB, STCAT and BITMAP indexes. In: Proceedings of the 5th International Symposium on Parallel Computing in Electrical Engineering, pp. 43–48 (2006)

6. Gorawski, M., Gorawski, M.J., Bańkowski, S.: Selection of indexing structures in grid data warehouses with software agents. International Journal of Computer Science & Applications 4(1), 39–52 (2007)
7. Gorawski, M., Malczok, R.: On efficient storing and processing of long aggregate lists. In: Proceedings of the International Conference on Data Warehousing and Knowledge Discovery, Copenhagen, Denmark, pp. 190–199 (2005)
8. Keung, H.M.L.C., Cao, J., Spooner, D.P., Jarvis, S.A., Nudd, G.R.: Grid information services using software agents. In: Proceedings of the 18th Annual UK Performance Engineering Workshop, pp. 187–198 (2002)
9. Manolopoulos, Y., Nanopoulos, A., Papadopoulos, A.N., Theodoridis, Y.: R-Trees: Theory and Applications. Springer, Heidelberg (2006)

Performing Range Aggregate Queries in Stream Data Warehouse

Marcin Gorawski and Rafał Malczok

Abstract. The number of various areas of everyday life where data warehouse systems find their application grows rapidly. They are used not only for business purposes, as it used to be a few years ago, but also in many various domains where huge and rapidly changing data volumes must be stored and processed. More and more often we think about such data volumes as endless data streams which require continuous loading and processing. The concept of data streams results in emerging a new type of data warehouse – a steam data warehouse. Stream data warehouse, when compared to standard data warehouse, differs in many ways, the examples can be a continuous ETL process or data mining models which are always kept up-to-date. The process of building stream data warehouse poses many new challenges to algorithms and memory structures designers. The most important concern efficiency and memory complexity of the designed solutions. In this paper we present a stream data warehouse cooperating with a network of sensors monitoring utilities consumption. We focus on a problem of performing range aggregate queries over the sensors and processing data streams generated by the chosen objects. We present a solution which, basing on the results of our previous work, integrate a dedicated memory structure with a spatial aggregating index. The paper includes also practical tests results which show high efficiency and scalability of the proposed solution.

Keywords: data warehouse, data streams, range aggregate.

1 Introduction

Human activities produce daily large volumes of data which require storing and processing. Data warehouse systems offer the ability to perform those operations

Marcin Gorawski · Rafał Malczok
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail:{Marcin.Gorawski,Rafal.Malczok}@polsl.pl

efficiently, thus they become more and more popular. This trend is supported by products of large software companies who enrich they business offer with ready-to-use data warehouse solutions integrated with broadly known database systems. Systems built from a scratch and dedicated for a given company are no longer the only way to create a data warehouse.

The extension of the domain in which data warehouse systems are applied results in the need of supporting various types of data. Very interesting are aspects of adapting data warehouse to be used for processing stream data. There are projects [1, 6, 9] focused on designing systems which make possible to register and evaluate continuous queries [1]. Stream data warehouse systems pose many new challenges which do not occur in standard data warehouses. One of the most important is the problem concerning the data loading process (ETL process). Changing the nature of the ETL process from batch to continuous forces a designer of a stream data warehouse to provide efficient mechanisms for processing and managing stream data. In this paper we focus on a problem of answering range aggregate queries over a set of objects generating data streams. We present a solution consisting of a modified spatial index and a dedicated memory structure, which allows efficient range queries evaluation.

The remaining part of the paper is organized as follows: in Sect. 2 we present details of the example motivating our research and we define a problem we address in this paper. The next two sections contain the details of the process of spatial range query evaluation. Finally, we show experimental tests results and we conclude the paper discussing our future plans.

2 Motivating Example and Problem Definition

Many of the stream data processing researches are motivated by the need of handling endless streams of sensor data [3, 7]. Our motivation is not different – we investigate the operation of a system of integrated utilities meters reading. The system monitors consumption of utilities such as water, natural gas and electrical energy. The meters, located in a region of telemetric installation, send the reading via radio waves to collecting points called nodes. The nodes, using standard TCP/IP network, transfers the readings to telemetric servers which are sources for the ETL process.

Considering the telemetric system operation we can assume that every single meter is an independent source generating an endless stream of readings. The intensity of a stream depends on the given meter configuration parameters defining how often the readings are being sent to the collecting point. User can be interested in the following aspects:

- What is the total number of meters located in a specified region (or regions) of the telemetric installation?
- How much utilities is consumed by inhabitants of the region (or regions)?

Let us consider the above list as a query which answer consists of two parts. The first part provides information about the number of various kinds (water, gas, etc.) of

meters located in the query region. The second part is a merged stream (or streams) of aggregated readings flowing from the meters encompassed by the region query.

There are many spatial indexes which, in a very fast and efficient way, can answer any range query. The best known and the most popular are indexes based on the R-Tree [5] spatial index. The main idea behind the R-Tree is to build a hierarchical indexing structure in which higher level index nodes encompass regions of the nodes on lower levels of the index. Using one of the indexes derived from the R-Tree family (e.g., R*-Tree [2]) we can quickly obtain the first part of an answer.

In order to calculate the second part, we need to apply an indexing structure which, in index nodes, stores aggregates concerning objects located in the nodes regions. The first solution proposed for this problem was aR-Tree [8] (aggregation R-Tree), an index which structure is identical to R-Tree but its nodes located on the higher levels of the hierarchy store the number of objects in the nodes located on the lower levels. Functionality of this solution can be easily extended by adding any kind of aggregated information stored in the nodes, e.g., sums or averages of the objects attribute values. The concept of partial aggregates was used in solution presented in [10], where the authors want to use it for creating a spatio-temporal index. Motivated by the index characteristics the authors suggested sorted hashing tables to be used in indexing structure nodes.

The designers of the mentioned spatial aggregating indices assumed, that the size of the aggregated data is well defined and small enough to fit into the computer main memory. Such an assumption cannot be made for a data with stream nature. In the following sections we present: in Sect. 3 Materialized Aggregates List which is a key component of indexing structure nodes and, in Sect. 4, operation of a spatial index equipped with MAL.

3   Stream Data Processing

The core of our solution is Materialized Aggregates List (MAL). In this section we present only the most important information, more details can be found in paper [4]. The idea of MAL is to process any number of raw data streams to a form of stream of aggregates; the aggregates are calculated for a given time interval called aggregation window. MAL can be applied as an autonomous solution or as a component of hierarchical indexing structure nodes.

MAL bases its operation on a well-known concept of list and iterator. Iterators are used for browsing the generated data and communicating with the list, while the main task of the list is to provide iterators with aggregates. To browse the list the iterator requires a static table (iterator table). The table is divided into logical parts called pages. Pages are used when the table is being filled with aggregates by a one of three multithreaded page-filling algorithms. By using a static table MAL allows processing data streams of any length without imposing memory limitations. MAL provides also an optional materialization mechanism which can significantly speed-up the process of aggregates recreation.

MAL calculates aggregates retrieving source data from one of four data sources:

1. Other MALs. This source is used when MAL works as a component of a node located on intermediate level of an indexing structure. The structure must be hierarchical – the upper level nodes encompass the regions (and also objects) of the nodes on the lower levels and a parent node must be able to identify its child nodes.
2. Stream of sensor readings (in general, a raw data stream). When MAL's aggregates creating component is integrated with the stream ETL system, the aggregates can be created in on-line mode. This mode is much more efficient when compared to retrieving data from database because no I/O operations are performed.
3. Stream history stored in database. Utilizing standard SQL language MAL queries the database and creates aggregates. This source is used only when the stream source cannot be used for some reason (e.g., requested aggregates are no longer available in the stream).
4. Materialized aggregates. Some of the aggregates are materialized for further use and, prior to accessing the database for retrieving aggregates, the MAL engine checks if the required aggregates are present in a dedicated table.

4   Answer Stream Generation Process

As mentioned in Sect. 2, the second part of an answer to a discussed range query is an integrated stream of aggregates. In this section we describe the process of answer stream generation, focusing on the problem of memory management.

The process begins with finding the indexing structure nodes and objects (which, in the context of our motivating example, are meters) answering the query. The spatial index is recursively browsed starting from its root. Depending on the relation of the query region O and the node region N one of the following operations is performed:

- if the window region and the node region share no part, $(O \cap N = \emptyset)$, the node is skipped,
- if the window region encompasses the node region $(O \cap N = N)$, the node is added to a *FAN* set (Full Access Node – a node which participates in the answer generation process with all its objects),
- finally, if the window region and the node region share a part $(O \cap N = O')$ then the algorithm performs a recursive call to lower structure levels, passing parameter O' as an aggregate window.

When traversing to lower tree levels it is possible that the algorithm reaches the node on the lowest hierarchy level. In this case, the algorithm must execute a range query searching for encompassed objects. A set of found objects is marked with a letter M.

4.1 Iterator Table Assignment Algorithm

MAL iterator uses a static table for browsing and managing aggregates stream. The tables are stored in a form of a resource pool. It is possible for an external process to use MAL iterator only if there is at least one free table in the pool. In the case there are no free tables, the external process must wait until at least one table is returned to the pool. An important thing to notice is that MAL is able to operate (read and/or write) on its materialized data only if its iterator owns a table.

Defining the number of tables stored in the pool one can very easily control the amount of memory consumed by the part of the system responsible for processing data streams. In most cases the number of indexing structure nodes involved in the process of answer stream generation is greater than the number of tables available in the table pool. To make the process most efficient we proposed a set of criteria according to which the available tables are distributed to appropriate MALs:

1. The number of materialized data available for a given node. The more materialized data, the higher the position of a node. Only the materialized data that can be used during answer stream generation is taken into account.
2. Generated stream materialization possibility. Every indexing structure node is located on a certain level. One of the configuration parameters is the materialization level, starting from which the streams generated by the nodes are materialized. The node which aggregates will be materialized is always placed at higher position than a node which stream will not be materialized.
3. The last sorting criteria is the amount of objects encompassed by the node's region. The more encompassed objects, the higher the position of the node.

Alike the elements in *FAN* sorted are the objects in the *M* set. In the case of the *M* set, the only sorting criteria is the amount of available materialized data which can be used during the answer stream generation.

After the elements in the *FAN* and *M* sets are sorted, the algorithm can assign the iterator tables to the chosen elements. Let P denotes the iterator table pool, and $|P|$ the number of tables available in the pool. If $((|FAN| = 1$ and $|M| = 0)$ or $(|FAN| = 0$ and $|M| = 1))$ (the answer to the query is a single stream generated by a node or a single object) then one table is taken from the pool and it is assigned to an element generating the answer. The condition that must be fulfilled is $|P| \geq 1$ (there is at least one table in the pool).

In the other case $(|FAN| \geq 1$ and/or $|M| \geq 1)$ the algorithm requires an additional structure called *GCE* (Global Collecting Element). The GCE element is of type MAL and is used for merging streams generated by other elements (nodes and objects). The *GCE* element requires one iterator table. One table in the iterator table pool is enough to answer any region query. If there are some free tables in the pool they are assigned to other elements involved in the result stream generation process. First, the tables are assigned to the elements in the *FAN* set (according to the order of elements set by the sorting operation). Then, if there are still some
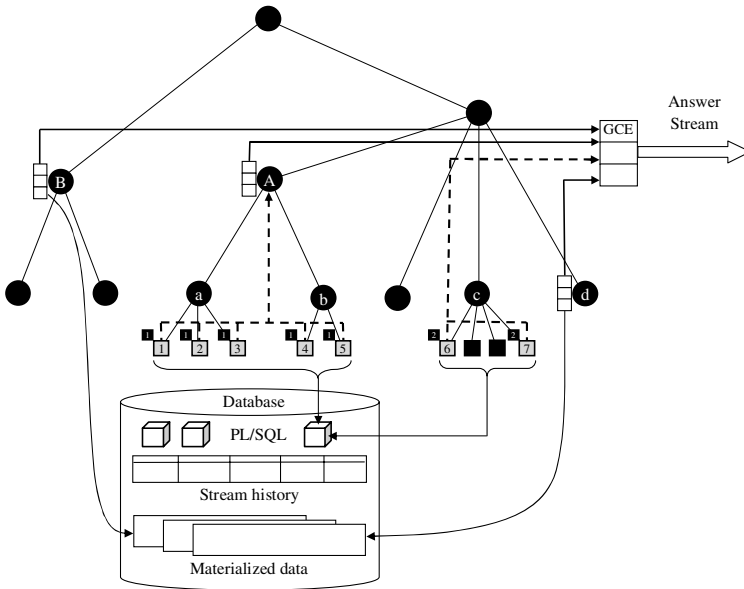
Fig. 1 Schema of the answer stream generation process during which instances of MAL are used as components of hierarchical indexing structure nodes. In the analyzed example the available iterator tables were assigned to *A*, *B*, *d* nodes and to the *GCE* element

free tables, they are assigned to single objects in the *M* set (also in appropriate order).

Stream of aggregates can be materialized only if a generating element is assigned a separate iterator table. If a given element's aggregates stream is at generation stage merged with streams of other elements, the materialization is not performed because it cannot be explicitly determined which element generated the stream. In general, partial answer streams, for example merged in *GCE*, are not materialized. Also, the stream generated by GCE is not materialized either, because it changes with every query. If the system materialize every temporarily created stream, the amount of materialized data would grow very fast.

Figure 1 presents an example of answer stream generation process. The query region encompasses intermediate level nodes *A* and *B*, leaf node *d* and two objects 6 and 7 located in the region of a node *c*. For the purpose of this example we assume there are 4 iterator tables available in the table pool, one of them is assigned to the *GCE* element. After sorting the elements, the *B* and *d* nodes are assigned separate tables (there is materialized data available for them). Also, the *A* node is assigned a table because it encompasses more objects than the *c* node. An important thing to notice is that aggregates created by the *A* node will be materialized after generation but the *c* node aggregates will not (the node has not its own table).

5 Test Results

In this section we present experimental test results. The test model contained 1000 electrical energy meters. Every meter generated a reading in a time period from 30 to 60 minutes. The aggregation window width was 120 minutes and the aggregates were created from the streams history stored in a database. The table storing readings counted over 12 million rows.

We tested the system by first defining a set of query windows and then calculating the answer. The times presented in the charts are sums of answer calculation and stream browsing times.

In order to investigate various query defining possibilities we separated four different query variants:

1. a query encompassing 2 indexing structure nodes (each node encompassed 25 objects) and 50 single objects (variant marked as $50 + 50$),
2. a query encompassing 3 indexing structure nodes and 25 single objects (variant marked as $75 + 25$),
3. a query encompassing 4 indexing structure nodes (marked as 4×25),
4. a query encompassing 1 indexing structure node located on a higher level of the indexing structure (node encompassing 100 objects, marked as 1×100).

Figures 2a and 2b show answer stream browsing times for the above described queries. Figure 2a shows times for the first run when no materialized data is available. In Fig. 2b we can observe times for the second run, when the system can use materialized data created during the first run. The first and the most important observation is that stream materialization has a great positive influence on the query answering process (more detailed information on this topic can be found in [4]). The results show also, that the stream browsing times depends strictly on the number of MALs used for answer stream generation. The less the number of MALs, the less database – application data transfers and the shorter the stream browsing times. But, on the other hand, we observe that even thought the first run times increase with the number of MALs, during the second run the times significantly decrease.

(a)   (b)
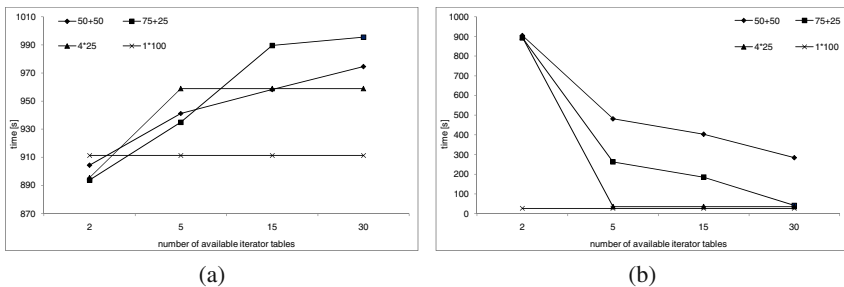
Fig. 2 Answer calculation times for various number of available iterator tables: **(a)** no materialized data, **(b)** using materialized data

6 Conclusions and Future Plans

Our previously designed solution – Materialized Aggregate List (MAL) can be used as a component of any spatial aggregating index processing data streams. In order to apply MAL in indexing structure we needed to define an algorithm that distributes available iterator tables among the elements involved in the answer stream calculation process. The algorithm, using a set of criteria, sorts all the involved elements and then assigns the tables. The most important criteria is the amount of available materialized data that can be used during the answer stream generation process.

In the nearest future we want to precisely check when the materialization process can be omitted. We want to apply many sets of various queries to help us to define a materialization threshold. Basing on tests results analyses we suppose that, if some of the streams generated by the lowest level nodes and single objects are not materialized, it will cause no harm to answer generation time but will significantly reduce the number of materialized data stored in the database.

References

1. Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and issues in data stream systems. In: Proceedings of the PODS Conference, pp. 1–16 (2002)
2. Beckerman, N., Kriegel, H.P., Schneider, R., Seeger, B.: The R*-tree: An efficient and robust access method for points and rectangles. In: Proceedings of SIGMOD International Conference on Management of Data, pp. 322–331 (1990)
3. Bonnet, P., Gehrke, J., Seshadri, P.: Towards sensor database systems. In: Tan, K.-L., Franklin, M.J., Lui, J.C.-S. (eds.) MDM 2001. LNCS, vol. 1987, pp. 3–14. Springer, Heidelberg (2001)
4. Gorawski, M., Malczok, R.: On efficient storing and processing of long aggregate lists. In: Proceedings of the International Conference on Data Warehousing and Knowledge Discovery, Copenhagen, Denmark, pp. 190–199 (2005)
5. Guttman, A.: R-trees: adynamic index structure for spatial searching. In: Proceedings of the SIGMOD Conference, Boston, US, pp. 47–57 (1984)
6. Hellerstein, J., et al.: Adaptive query processing: Technology in evolution. IEEE Data Engineering Bulletin, 7–18 (2000)
7. Madden, S., Franklin, M.J.: Fjording the stream: An architecture for queries over streaming sensor data. In: Proceedings of the International Conference on Data Engineering, pp. 555–566 (2002)
8. Papadias, D., Kalnis, P., Zhang, J., Tao, Y.: Effcient OLAP Operations in Spatial Data Warehouses. In: Jensen, C.S., Schneider, M., Seeger, B., Tsotras, V.J. (eds.) SSTD 2001. LNCS, vol. 2121, p. 443. Springer, Heidelberg (2001)
9. Terry, D., Goldberg, D., Nichols, D., Oki, B.: Continuous queries over append-only databases. In: Proceedings of the SIGMOD Conference, pp. 321–330 (1992)
10. You, B., Lee, D., Eo, S., Lee, J., Bae, H.: Hybrid index for spatio-temporal OLAP operations. In: Yakhno, T., Neuhold, E.J. (eds.) ADVIS 2006. LNCS, vol. 4243, pp. 110–118. Springer, Heidelberg (2006)

LVA-Index: An Efficient Way
to Determine Nearest Neighbors

Piotr Lasek

Abstract. In this paper we present our new *LVA-Index* for indexing multidimensional data. The *LVA-Index* has a layered structure improving performance when searching for nearest neighbors. The index combines some features of the VA-File and the *NBC* algorithm, namely: the idea of approximation of the data vectors and the idea of layers. The crucial advantage of the *LVA-Index* is that it stores n neighbor layers for each cell. For this reason, contrary to the VA-File, the *LVA-Index* does not require scanning of the entire approximation file. Our experiments proved that searching using the *LVA-Index* is faster than searching using the VA-File which was designed to effectively handle multidimensional data.

Keywords: indexing, index structure, VA-File, nearest neighbor search.

1 Introduction

The problem of searching data in multidimensional data spaces has been investigated over the past years. For spaces with less than five dimensions fast solutions exist. However, when the number of dimensions increases, the problem of the nearest neighbors searching requires linear time to be solved. This problem is called the *curse of dimensionality*.

Over the years the number of algorithms for searching data in multidimensional data spaces have been developed, including *Quadtree* [2], *k-d-tree* [6], *Gridfile* [4], *Voronoi Diagrams* [5], *R-Tree* [3].

Piotr Lasek
Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: `p.lasek@ii.pw.edu.pl`

Finally, a simple, yet efficient index known as the *Vector Approximation File* (*VA-File*) [1] was developed at ETH in Zurich in 1997. The crucial feature of the VA-File is that it employs the mechanism of approximations (bit strings of a specific length). The VA-File is simply an array of the approximations of the data vectors. By using the array of the approximations, when searching through the data vectors, a great amount of irrelevant vectors can be easily excluded from farther computations. The mechanism of approximations makes building sophisticated hierarchical structures unnecessary, and solves the problem of the *curse of dimensionality*.

In this paper we introduce the *LVA-Index*. The *LVA-Index* is designed for indexing multidimensional data. It has a layered structure and combines some features of VA-File [1] and *NBC* [8].

The layout of the paper is as follows: the VA-File, basic notions and a layered approach are recalled in Sect. 2. In Sect. 3, we offer the *LVA-Index*. The experimental results are discussed in Sect. 4. We conclude our results in Sect. 5.

2 The VA-File and the Layers

In this section, we recall the basic notions, which are necessary to describe the VA-File and *LVA-Index*. Next, we shortly describe the *layered approach* which was used in the *NBC* algorithm [8].

2.1 Basic Notions

In the sequel we will use the notations presented in Table 1.

Table 1 Basic notations

Notation	Meaning
q	a query point for which the nearest neighbors search is performed
NN	Nearest Neighbors – points which are located in the neighborhood of the query point
kNN	the number of NN to be found
DS	data set – a multidimensional set of points
C	cell – an element of the data space; contains data points
d	the number of dimensions

In Fig. 1, we have plotted a sample two-dimensional data space with some data points. One of this points (a square) is a query point and is located in the middle of the circle. Points which are located within the circle (triangles) are the nearest neighbors of the square query point because the distance from each of these points to the
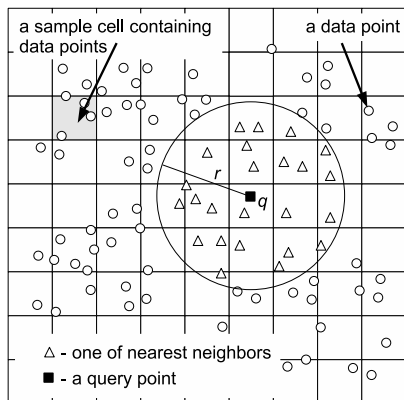
Fig. 1 An example of a two dimensional data space divided into cells containing data points

query point does not exceed r. In many cases, the problem of searching the nearest neighbors focuses on determining the points which belong to the area restricted by the specific radius r. However, in this paper we rather focus on the problem of searching of the k-nearest neighbors (k-NN):

Problem 1 (The k-Nearest Neighbors Search Problem)

- Given: a query vector q, a set P of vectors p_i ($0 \leq i \leq N$), a number k of nearest neighbors to be found and a distance function δ.
- Find: a set $R \subseteq P$ such that $|R| = k \land \forall s \in P \setminus R \land \forall r \in R \; \delta(q,s) \geq \delta(q,r)$.

2.2 The VA-File

The VA-File was proposed in order to reduce the amount of data that must be read during similarity searches [7]. The VA-File divides the data space into 2^b rectangular cells, where b is the number of bits per dimension. The VA-File allocates a string of the length b for each cell and approximates data points that fall into a cell identified by that bit-string.

The VA-File algorithm is based on the idea of approximating vectors. Approximations of vectors are computed using the approximation function $a(p, j)$, where p is a data point and j is an index of a dimension (dimensions are numbered from zero). The approximation function returns the number of the slice in the jth dimension. In order to compute the approximation of a vector the results of approximation function $a(p, j)$ for $j = 0, \ldots, d - 1$ must be concatenated. E.g., the approximation a of p_4 in Fig. 2 is as follows: $a(p_4) = 1100$. In the sequel, we use approximations for computing coordinates of cells.

Having b and the maximum length of the data space in each dimension, *marks* can be determined. Marks represent borders between cells (see Fig. 3). Using marks we can define *slices* – slice is a space between two marks. E.g., the slice 10 (or 2
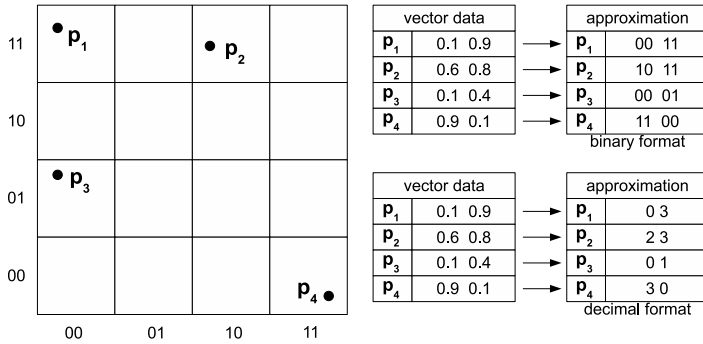
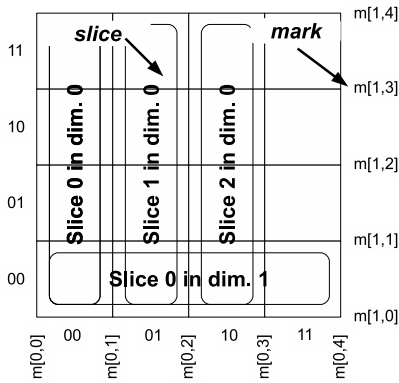**Fig. 2** Building the 'flat' VA-File (from [7])

Fig. 3 The data space with imarks and islices

in decimal format) in zeroth dimension is represented by marks: $l = [0,2]$ (lower) and $u = [0,3]$ (upper). Using the notions of the approximation and the mark, the lower mark l and the upper mark u can be defined: $l_j(p) = m[j, a(p,d)]$, $u_j(p) = m[j, a(p,j)+1]$, where j is the number of a dimension. In the sequel, we also use the notion of the lower bound (*lBnd*), which is used to determine the distance between a query point and a cell's closest mark.

The Layered Approach

The authors of the *NBC* algorithm [8] have used the VA-File for organizing the data. They have also applied the *layered approach* for determining the nearest neighbors. The *layered approach* gives an immediate access to the neighbor layers and improves performance of determining the nearest neighbors. For this reason, we have decided to combine the idea of approximations (from VA-File) with the idea of the layered approach (from *NBC*).

3 The *LVA-Index*

In this section we introduce the *LVA-Index*. First subsections describe the structure of the *LVA-Index*. Next, we introduce a corrected definition of the *layers* and provide a property that improves building the *LVA-Index* and a method of its building. We complete this section with the description of the *LVA-NNS* algorithm, which searches the nearest neighbors in the *LVA-Index*.

3.1 *The Structure of the* LVA-Index

The *LVA-Index* is composed of cells and layers of cells. Cells are defined in terms of marks. As mentioned above, marks are related to the partitioning of the data space. When partitioning the data space, the jth dimension is divided into 2^{b_j} slices with b_j being the number of bits assigned to the jth dimension. Slices are bounded by marks, e.g., the 3rd slice in the 2nd dimension is bounded by marks $m[1,3]$ and $m[1,4]$, as shown in Fig. 4 (the numbering of slices and marks starts from 0).

Using the notion of the mark we can define the cell: $Cell[c_0,\dots,c_j] = \{x_0,\dots,x_n\} \mid m[0,c_0] \le x_0 < m[0,c_0+1],\dots,m[j,c_j] \le x_n < m[j,c_j+1]$, where j is the number of a dimension and $c_0,\dots,c_{d-1}$ are coordinates of the cell.

Example 1. In Fig. 4, the $Cell[2,3]$ has the following coordinates: 2 (in the zeroth dimension) and 3 (in the first dimension) and is represented by the area of the data space that is bounded by marks $m[0,2]$ and $m[0,3]$ (along x axis – 0th dimension) and $m[1,3]$ and $m[1,4]$ (along y axis – 1st dimension).

The layers are sets of cells, which can be determined according to the following definition:
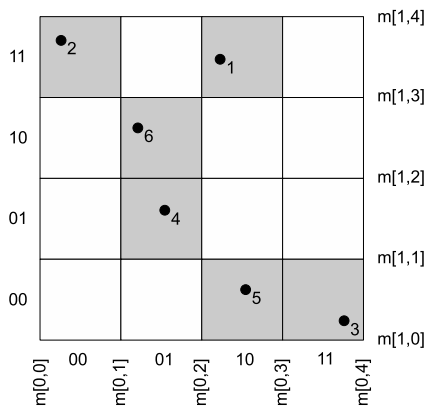
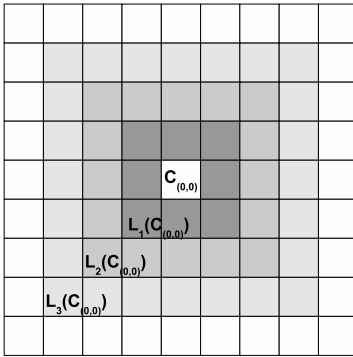**Fig. 4** Cells defined in terms of marks

Fig. 5 Three layers of $Cell[0,0]$

Definition 1. *The layer is defined recursively in terms of cells as follows:*

$$L_0(Cell_{x_1,\ldots,x_d}) = \{Cell_{x_1,\ldots,x_d}\},$$
$$L_n(Cell_{x_1,\ldots x_d}) = \{Cell_{u_1,\ldots,u_d}|u_i = x_i \pm j, 0 < j \leq n,$$
$$Cell_{u_1,\ldots,u_d} \notin L_m(Cell_{x_1,\ldots,x_d}, 0 \leq m < n)\}.$$

Definition 1 is a corrected version of the confusing definition of the layer from [8][1]. The number of cells which belong to the kth layer, where $k > 0$, can be easily counted using the following formula: $n = (2k+1)^d - (2k-1)^d$.

Example 2. In Fig. 5, we have illustrated the concept of a layer for the two dimensional data space. We have marked three layers for the center cell having coordinates $[0,0]$. The first layer has 8 cells, the second one has 16 cells and third one has 24 cells.

The *LVA-Index* is designed so that each non-empty cell contains the list of the references to the vectors laying within the cell and the list of l nearest layers. In Fig. 6, we present the structure of the *LVA-Index* in the two dimensional data space.

In our implementation of the *LVA-Index* the structure storing cells is designed as a multidimensional table. The elements of this table are pointers to the cells. The empty cells have simply *NULL* values. In the near future, we are going to optimize the structure of the *LVA-Index* and implement the rare table for storing the cells pointers.

3.2 Building the LVA-Index

When building the *LVA-Index* the following property is applied:

Property 1. If a cell c_1 belongs to the kth layer of a different cell c_2, then the cell c_2 belong to the kth layer of the cell c_1.

---

[1] The definition from [8] did not take into account that values of coordinates of the nth layer's cells can be different than $x_i \pm n$.
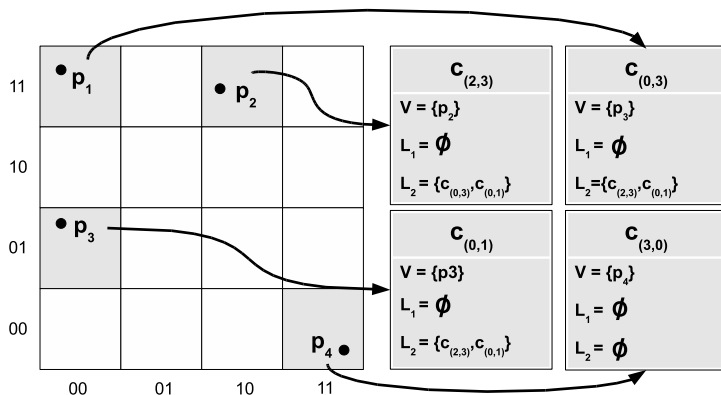
Fig. 6 The structure of the *LVA-Index* in the two-dimensional data space. The data space is divided into the cells which contain sets of data vectors (V) and lists of layers (L_i). The number l of layers stored in each cell is equal to 2

The *LVABuild* (Fig. 7) function takes two arguments: a set of vectors V and the reference to the *LVA-Index*, which initially is empty. The *LVABuild* is composed of the main loop and the nested loop. The former iterates through all vectors of the vectors set and the latter iterates through the layers of the cells. In the main loop the cell for the currently processed vector is determined and this vector is assigned to the cell. After that, in the nested loop, Property 1 is applied in the following way. First, the kth layer of the determined cell is retrieved and non-empty cells of this layer are returned. Then, for each non-empty cell, using the VA-File, the kth layer is determined. The layers are determined using the VA-File (see Fig. 12). If the cell that was determined in the most outer loop was not yet assigned to the kth layer, then it is assigned to this layer. Finally, after iterating through all vectors, the *LVA-Index* is built.

Notation for *LVABuild*

- l – This is the number denoting the maximum number of the layers stored in a cell given by the user.
- *AssignVector*(Cell *cell*, Vector v) – As described in the section regarding the structure of the *LVA-Index*, each cell contains the list of pointers to the vectors laying within this cell. The *AssignVector*() function adds a vector's reference to the list of pointers in *cell*.
- *LVA.DetermineCell*(Vector v) – The *LVA-Index* provides a function *DetermineCell*() for determining a cell for a given vector. In order to return the reference to the appropriate cell the *DetermineCell*() function uses the VA-File. If the cell for the given vector does not exist (the corresponding element in the multidimensional table described in the section regarding the structure of the *LVA-Index* is *NULL*), then it is created.

```
function LVABuild(Vector set V, LVA-Index LVA)
 1.  forall v ∈ V do begin
 2.     cell = LVA.DetermineCell(v)
 3.     AssignVector(cell, v)
 5.     for (k = 1; k < l; k++) do begin
 6.        L_k = GetReducedLayer(c, k)
 7.        forall c ∈ L_k do begin
 8.           R_k = GetLayer(LVA, c, k)
 9.           if (cell ∉ R_k) then
10.              R_k.AssignNonEmptyCell(cell)
11.           endif
12.        endfor
13.     endfor
14.  endfor
```

Fig. 7 Function *LVABuild*. This function builds the *LVA-Index*. It takes two parameters: a reference to a set of input vectors and a reference to an empty *LVA-Index*

- *GetLayer*(Cell *cell*, Integer k) – Returns the kth layer of the *cell*. The layer contains the set of cells which belong to the layer. The *GetLayer* uses the VA-File to determine coordinates of the cells which belong to kth layer of the cell.
- *GetReducedLayer*(Cell *cell*, Integer k) – Returns a set of non-empty cells, which belong to kth layer stored in *cell*.

In Fig. 8, we have shown the *GetLayer* function. This function is designed to return the kth layer for the given cell using the VA-File. The function scans the entire approximation file and returns cells, which belong to the kth layer of the given cell. In order to determine these cells, the *GetLayer* uses the *GetCellDistance* function, which computes the distance between cells according to the following equation: $dist = \max(|a_0 - b_0|, |a_1 - b_1|, \ldots, |a_j - b_j|), 0 < j < d - 1$, where a_j and b_j, are coordinates (approximations) of the cell in jth dimension.

```
function GetLayer(LVA-File lva, Cell c, Integer k)
 1.  forall approx ∈ lva.vafile do begin
 2.     l = GetCellDistance(cellApprox, c)
 3.     if l = k then
 4.        cell = DetermineCell(approx)
 5.        if cell.V ≠ ∅ then
 6.           L_k.AddNonEmptyCell(cell)
 7.        endif
 8.     endif
 9.  endfor
10.  return L_k
```

Fig. 8 Function *GetLayer*. This function returns a set of cells, which belong to the kth layer of the given cell. It takes three parameters: a reference to the VA-File containing approximations of all data vectors, a cell c and a number k of the layer

---

function *LVA-NNS*(LVA-Index *LVA*, Vector *q*, Integer *k*, Real *r*)

- *q* – the query vector for which we search the nearest neighbors
- *k* – the number of the nearest neighbors to be found
- *r* – the maximum radius determining the search area

```
1.  cell = LVA.DetermineCell(q)          // determine a cell for the given query vector q
2.  dist = 0
3.  max = InitCandidate(r)
4.  foreach layer ∈ cell.layers do
5.    foreach cell ∈ layer do
6.      lB = lowerBound(cell.FirstObject(), q)
7.      if (lB < max) then
8.        for each p ∈ cell do                          //p is a candidate point
9.          dist = getDist(p, q)
10.         max = Candidate(p, dist)
11.       end for
12.     end if
13.   end for
14. end for
```

Fig. 9 The *LVA-Index Nearest Neighbors Search Algorithm*

3.3 Searching

In Fig. 9, we have presented the *LVA-Index Nearest Neighbors Search Algorithm* (*LVA-NNS*). The data structures are the same as in the original VA-File except that the cells store information about the neighbor layers. Like in the VA-File, the result of the search is stored in *v* and *dst* arrays. However, the VA-File is not used in *LVA-NNS*. These arrays are of the length *k*, where *k* is the number of nearest neighbors to be found. The *v* array contains the list of the nearest points sorted by the distance to the query point *q*. The *dst* array comprises the distances between points stored in the *v* array and the query point *q*.

4 Experiments

We performed two series of the experiments on several synthetical datasets. Results are presented in Tables 2 and 3 (*DS* – dataset id, |*O*| – number of vectors, |*AT*| – number of attributes, *k* – number of nearest neighbors to be found, A_1 – *LVA-NNS* algorithm, A_2 – *NNSearchSSA* algorithm[2]).

First, we examined the performance of the *LVA-NNS* by changing the number of dimensions (Table 2), next, by increasing the number of points (Table 3). During

---

[2] *NNSearchSSA* is a better readable version of the VA-File *Simple Search Algorithm* (*VA-SSA*) [7] and can be found on the *ETH Zurich – Databases and Informations Systems* website (http://www.dbis.ethz.ch/education/ws0607).

Table 2 Results for datasets 1, 2, 3. $r = 70$. Times in ms

DS	$|O|$	$|AT|$	k	Runtime of A_1	A_2	Ratio
1	2000	2	5	125	1063	9
			10	141	1109	8
			15	188	1171	6
			20	219	1250	6
			25	281	1328	5
2	2000	3	5	281	1469	5
			10	313	1500	5
			15	359	1547	4
			20	406	1578	4
			25	438	1625	4
3	2000	4	5	203	1860	9
			10	234	1875	8
			15	250	1891	8
			20	266	1906	7
			25	282	1890	7

Table 3 Results of finding k-nearest neighbors. Times in ms

DS	$|O|$	$|AT|$	k	Runtime of A_1	A_2	Ratio
4	10000	3	5	15	360	24
			10	0	375	–
			15	16	375	23
			20	15	375	25
			25	32	375	12
5	20000	3	5	15	719	48
			10	16	719	45
			15	31	734	24
			20	31	750	24
			25	32	750	23
6	50000	3	5	16	1828	114
			10	31	1828	59
			15	32	1828	57
			20	32	1843	58
			25	31	1859	60

each experiment reported in Table 2, the VA-File and *LVA-Index* were queried for neighbors for 1000 query objects. One can notice that times shown in Table 2 increase with the growing k (the number of nearest neighbors to be found). On the

other hand, when analyzing Table 2, there is no crucial difference of times with respect to the number of dimensions. In Table 3, we presented the times measured for datasets 4–6.

5   Conclusions

As shown in Sect. 4, the *LVA-Index* algorithm is faster than the VA-File. The increase of the performance ranges from about 7 times for small datasets (i.e., datasets 1, 2, 3), up to even more than 100 times for large dataset (i.e., datasets 4, 5, 6). The algorithm can be very memory consuming when the parameters are chosen incorrectly, especially when the number of neighbor layers stored in the cells is too large. The number of neighbor layers stored in the cells is also crucial when building the *LVA-Index* – if this number is too large the building of the index can take much time. In spite of the above disadvantages, when parameters of the *LVA-Index* are chosen reasonably, the results of the experiments are very promising and encourage us to continue research in this area.

References

1. Blott, S., Weber, R.: A simple vector approximation file for similarity search in high-dimensional vector spaces. Technical Report 19, ESPRIT project HERMES (no. 9141) (1997)
2. Finkel, R.A., Bentley, J.L.: Quad-trees: A data structure for retrieval on composite keys. ACTA Informatica 4(1), 1–9 (1974)
3. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, Boston, US, pp. 47–57 (1984)
4. Nievergelt, J., Hinterberger, H., Sevcik, K.C.: The grid file: An adaptable symmetric multikey file structure. ACM Transactions on Database Systems 9(1), 38–71 (1984)
5. Orenstein, J.A., Merrett, T.H.: A class of data structures for associative searching. In: Proceedings of the ACM Symposium on Principles of Database Systems, Waterloo, Canada, pp. 181–190 (1984)
6. Robinson, J.T.: The k-d-b-tree: A search structure for large multidimensional dynamic indexes. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 10–18 (1981)
7. Weber, R., Schek, H.J., Blott, S.: A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: Proceedings of the 24th VLDB Conference on Very Large Data Bases, New York, US, pp. 194–205 (1998)
8. Zhou, S., Zhao, Y., Guan, J., Huang, J.: A neighborhood-based clustering algorithm. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS, vol. 3518, pp. 361–371. Springer, Heidelberg (2005)

Part XI
Embedded Systems Applications

Basic Component of Computational Intelligence for IRB-1400 Robots

Tadeusz Szkodny

Abstract. In this paper the solution algorithm of the inverse kinematics problem for IRB-1400 robots is presented. This algorithm may be a basic component of future computational intelligence for theses robots. The problem of computing the joint variables corresponding to a specified location of the end-effector is called the inverse kinematics problem. This algorithm may be used with the current computer control algorithms for these robots. It would allow controlling these robots through the use of the vision system, which specifies the required location of the end-effector. This required location makes it possible for the end-effector to approach an manipulation object (observed by the vision system) and pick it up. These robots are equipped with several manipulator which has 6 links connected by the revolute joints. First the location of the end-effector in relation to the base of the manipulator is described. Next the analytical formulas for joint variables (dependent on these location) are derived. These formulas have take into account the multiple solutions for the singular configurations of this manipulator.

Keywords: kinematics, manipulators, mechanical system, robot kinematics.

1 Introduction

Currently the IRB-1400 robot's controllers are using joint variables as a base for controlling [8] a manipulator movement. These variables are then verified by a user

Tadeusz Szkodny, Affiliate Member, IEEE
Institute of Automatic Control, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: Tadeusz.Szkodny@polsl.pl

with a Rapid programming language [1]. For a particular point of trajectory the user obtains the joint variables from a control panel reading, as soon as he manually brings the manipulator to this point. Described procedure is known as teaching by showing [2].

The IRB-1400 robot's controllers can be equipped with an Ethernet network interface. It provides a quick communication between the controller and the terminal. Moreover this interface allows to establish the communication between the terminal and the vision cameras [1]. It is possible to set up a computational intelligence of the IRB-1400 robots by developing a specific terminal software. It calculates the Cartesian variables of the manipulator spotted by the manipulation object cameras [7], and computes the joint variables corresponding to the calculated external variables. Then the resultant joint variables can be sent by the Ethernet network from the terminal to the controller. Thanks to this software it is possible to transfer the joint variables from the terminal to the controller, to obtain a spotted object position without manually bringing the manipulator to this particular point. Such a software must include the analysis algorithms of image and solutions for the inverse kinematics problem. These algorithms are core elements of the robot's computational intelligence. This article presents a derivation of analytical formulas, which represents the solutions of the forward and inverse kinematics problem for the IRB-1400 robot. In the second section the description of the manipulator's kinematics structure is presented, followed by the forward kinematics equations. The solutions for the inverse kinematics problem are derived in Sect. 3. The fourth section contains an example of the joint variables computation. The last section concludes the article.

2   Forward Kinematic Problem

The IRB-1400 manipulator consists of a base link, which is fixed, and a 6 other links, which are movable. The last 6th link will be called an end-effector. The gripping device, or other tool, is attached to this link. The neighbouring links are connected by revolute joint. Figure 1 illustrates the manipulator kinematics schema with the coordinate systems (frames) associated with links according to a Denavit-Hartenberg notation [8, 4, 3, 5, 6]. The $x_7 y_7 z_7$ frame is associated with the gripping device. The position and orientation of the links and tool are described by the homogenous transform matrices. The matrix $\mathbf{A}_i$ describes the position and orientation of the ith link frame relative to $(i-1)$st. $\mathbf{T}_6$ is the matrix, that describes the position and orientation of the end-effector frame relative to the base link. Matrix $\mathbf{E}$ describes the gripping device frame relative to the end-effector frame. Matrix $\mathbf{X}$ describes the position and orientation of the gripping device frame relative to the base link. Matrix $\mathbf{A}_i$ is described by [5, 6]:

$$\mathbf{A}_i = \text{Rot}(z, \Theta_i)\text{Trans}(0, 0, \lambda_i)\text{Trans}(l_i, 0, 0)\text{Rot}(x, \alpha_i) , \qquad (1)$$
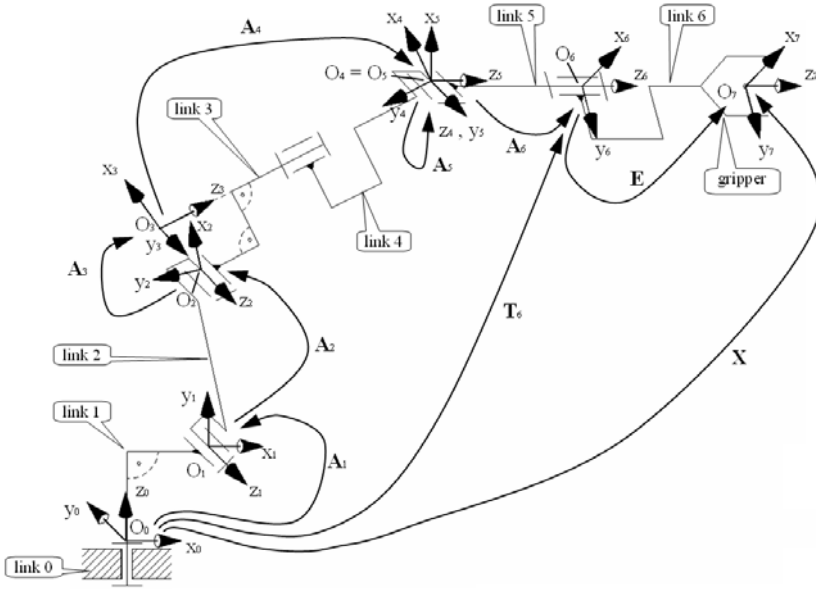
Fig. 1 The kinematic scheme of IRB-1400 Manipulator and link frames

where Θ_i, λ_i, l_i, α_i are the Denavit–Hartenberg parameters. For the description of the kinematics Θ_i' joint variables will be used. The variables $\Theta_i' = \Theta_i$ for $i = 1, 3, 4, 5, 6$ and $\Theta_2' = \Theta_2 - 90°$. For a notation simplicity following denotations will be used: $S_i = \sin \Theta_i'$, $C_i = \cos \Theta_i'$, $S_{ij} = \sin \Theta_{ij}'$, $C_{ij} = \cos \Theta_{ij}'$, $\Theta_{ij}' = \Theta_i' + \Theta_j'$.

The matrix $\mathbf{T}_6$ is derived from:

$$\mathbf{X} = \mathbf{A}_1 \mathbf{A}_2 \mathbf{A}_3 \mathbf{A}_4 \mathbf{A}_5 \mathbf{A}_6 \mathbf{E} =$$

$$= \begin{bmatrix} C_1(-S_{23}(C_4C_5C_6 - S_4S_6) - C_{23}S_5C_6) + S_1(S_4C_5C_6 + C_4S_6) \\ S_1(-S_{23}(C_4C_5C_6 - S_4S_6) - C_{23}S_5C_6) - C_1(S_4C_5C_6 + C_4S_6) \\ C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6 \\ 0 \end{bmatrix}$$

$$\begin{matrix} C_1(-S_{23}(-C_4C_5S_6 - S_4C_6) + C_{23}S_5S_6) + S_1(-S_4C_5S_6 + C_4C_6) \\ S_1(-S_{23}(-C_4C_5S_6 - S_4C_6) + C_{23}S_5S_6) - C_1(-S_4C_5S_6 + C_4C_6) \\ C_{23}(-C_4C_5S_6 - S_4C_6) + S_{23}S_5S_6 \\ 0 \end{matrix} \qquad (2)$$

$$\begin{bmatrix} C_1(-S_{23}C_4S_5 + C_{23}C_5) + S_1S_4S_5 & d_x \\ S_1(-S_{23}C_4S_5 + C_{23}C_5) - C_1S_4S_5 & d_y \\ C_{23}C_4S_5 + S_{23}C_5 & d_z \\ 0 & 1 \end{bmatrix},$$

where:

$$d_x = (C_1(-S_{23}C_4S_5 + C_{23}C_5) + S_1S_4S_5)\lambda_7 + l_1C_1 + \lambda_6S_1S_4S_5 +$$
$$+ (-l_2S_2 - l_3S_{23} + (\lambda_4 + \lambda_6C_5)C_{23} - \lambda_6S_{23}C_4S_5)C_1,$$
$$d_y = (S_1(-S_{23}C_4S_5 + C_{23}C_5) - C_1S_4S_5)\lambda_7 + l_1S_1 - \lambda_6C_1S_4S_5 +$$
$$+ (-l_2S_2 - l_3S_{23} + (\lambda_4 + \lambda_6C_5)C_{23} - \lambda_6S_{23}C_4S_5)S_1,$$
$$d_z = (C_{23}C_4S_5 + S_{23}C_5)\lambda_7 + \lambda_1 + l_2C_2 + (C_4S_5\lambda_6 + l_3)C_{23} +$$
$$+ (\lambda_4 + \lambda_6C_5)S_{23}.$$

Equation (2) allows to compute the position and orientation of the gripping device's frame $x_7y_7z_7$ relative to the base link's frame $x_0y_0z_0$ for the given joint variables Θ_i'. It is the forward kinematics problem of the manipulator.

3 Inverse Kinematic Problem

Solving the inverse kinematics problem is a matter of the computing the joint variables $\Theta_1' \div \Theta_6'$ for the given matrix $\mathbf{X}_{req}$. In a computations we will use a matrix method [2, 4, 3, 5, 6], which involves the manipulator kinematics equations $(1) \div (2)$.

3.1 Computing of Joint Variables $\Theta_1' \div \Theta_3'$

The matrix $\mathbf{E}$ is independent of the joint variables, that's why the inverse kinematics problem can be solved regardless of the gripping device structure. For this purpose we will use the matrix $\mathbf{T}_{6\,req} = \mathbf{X}_{req}\mathbf{E}^{-1}$, which describes the manipulator's end-efector. Let's suppose, that the matrix $\mathbf{X}_{req}$ has the form:

$$\mathbf{X}_{req} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

To compute the variables $\Theta_1' \div \Theta_3'$ we will use the following matrix:

$$\mathbf{T}_{6\,req}' = \mathbf{X}_{req}\text{Trans}(0,0,-\lambda_{67}) = \begin{bmatrix} a_x & b_x & c_x & d_x - \lambda_{67}c_x \\ a_y & b_y & c_y & d_y - \lambda_{67}c_y \\ a_z & b_z & c_z & d_z - \lambda_{67}c_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

$$\lambda_{67} = \lambda_6 + \lambda_7.$$

3.1.1 Computing Θ_1'

The range of the Θ_1' angle changes is $-170° \le \Theta_1' \le 170°$. Let's assume a following designation: $d_x' = d_x - \lambda_{67}c_x$, $d_y' = d_y - \lambda_{67}c_y$ and $d_z' = d_z - \lambda_{67}c_z$.

In the process of determining the angle Θ_1' the variables p and pp, which can take a value of 0 or 1, will be used. The values of this variable depend of whether the

points $P(d'_x, d'_y, d'_z)$ and $P'(-d'_x, -d'_y, d'_z)$ belong to the manipulator's workspace. If the point P belongs to this workspace then $p = 1$. Otherwise $p = 0$. When the point P' belongs to this workspace then $pp = 1$. Otherwise $pp = 0$.

For a computation of the variable Θ'_1 the angle φ, described by (4), is needed, in which d'_x and d'_y are the elements of the matrix $\mathbf{T}'_{6\,req}$, when d_x and d_y are the elements of the matrix $\mathbf{X}_{req}$.

$$\varphi = \begin{cases} \arctan 2(d'_y, d'_x) \in [-180°, 180°] & \text{for } d'^2_x + d'^2_y > 0, \\ \arctan 2(d_y, d_x) \in [-180°, 180°] & \text{for } d'^2_x + d'^2_y = 0 \text{ and } d^2_x + d^2_x > 0, \\ \varphi - \text{not limited, e.g.,} \varphi = 0° & \text{for } d'^2_x + d'^2_y = 0 \text{ and } d^2_x + d^2_x = 0. \end{cases} \quad (4)$$

The variable Θ'_1 is computed according to the following rules:

- For $-10° < \varphi < 10°$ or $\varphi \in \langle -170°, -10°) \cup \langle 10°, 170°\rangle$ and $pp = 0$: $\Theta'_1 = \varphi$.
- For $\varphi > 170°$ and $pp = 1$: $\Theta'_1 = \varphi - 180°$.
- For $\varphi < -170°$ and $pp = 1$: $\Theta'_1 = \varphi + 180°$.
- For $10° \leq \varphi \leq 170°$ and $pp = 1$: $\Theta'_1 = \varphi$ or $\Theta'_1 = \varphi - 180°$.
- For $-170° \leq \varphi \leq -10°$ and $pp = 1$: $\Theta'_1 = \varphi$ or $\Theta'_1 = \varphi + 180°$.

3.1.2 Computing Θ'_3

A range of the Θ'_3 angle changes is $-70° \leq \Theta'_3 \leq 65°$. In the process of determining the angle Θ'_3 we will use:

$$A_1^{-1}\mathbf{X}\text{Trans}(0,0,-\lambda_{67}) = A_1^{-1}\mathbf{X}_{req}\text{Trans}(0,0,-\lambda_{67}), \quad (5)$$

$$A_1^{-1} = [\text{Rot}(z, \theta_1)\, \text{Trans}(0,0,\lambda_1)\, \text{Trans}(l_1,0,0)\, \text{Rot}(x, 90°)]^{-1} =$$

$$= \begin{bmatrix} C_1 & S_1 & 0 & -l_1 \\ 0 & 0 & 1 & -\lambda_1 \\ S_1 & -C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$A_1^{-1}\mathbf{X}\text{Trans}(0,0,-\lambda_{67}) = \mathbf{T}^1_6 =$$

$$= \begin{bmatrix} -S_{23}(C_4C_5C_6 - S_4S_6) - C_{23}S_5C_6 & S_{23}(C_4C_5S_6 + S_4C_6) + C_{23}S_5S_6 \\ C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6 & -C_{23}(C_4C_5S_6 + S_4C_6) + S_{23}S_5S_6 \\ S_4C_5C_6 + C_4S_6 & -S_4C_5S_6 + C_4C_6 \\ 0 & 0 \end{bmatrix}$$

$$\begin{matrix} -S_{23}C_4S_5 + C_{23}C_5 & -l_2S_2 - l_3S_{23} + \lambda_4C_{23} \\ C_{23}C_4S_5 + S_{23}C_5 & l_2C_2 + l_3C_{23} + \lambda_4S_{23} \\ S_4S_5 & 0 \\ 0 & 1 \end{matrix} \Bigg],$$

$$\mathbf{A}_1^{-1}\mathbf{X}_{\text{req}}\text{Trans}(0,0,-\lambda_{67}) = \mathbf{T}_{6\,\text{req}}^1 =$$

$$= \begin{bmatrix} C_1 a_x + S_1 a_y & C_1 b_x + S_1 b_y & C_1 c_x + S_1 c_y \\ a_z & b_z & c_z \\ S_1 a_x - C_1 a_y & S_1 b_x - C_1 b_y & S_1 c_x - C_1 c_y \\ 0 & 0 & 0 \end{bmatrix}$$

$$\left. \begin{array}{c} -l_1 + S_1 d_y + C_1 d_x - \lambda_{67}(C_1 c_x + S_1 c_y) \\ -\lambda_1 + d_z - \lambda_{67} c_z \\ S_1 d_x - C_1 d_y - \lambda_{67}(S_1 c_x - C_1 c_y) \\ 1 \end{array} \right].$$

By comparing the elements (1,4) of (5), the equation can be obtained:

$$-l_2 S_2 - l_3 S_{23} + \lambda_4 C_{23} == -l_1 + S_1 d_y + C_1 d_x - \lambda_{67}(C_1 c_x + S_1 c_y) = w_1. \quad (6)$$

By comparing the elements (2,4) of (5), the equation can be obtained:

$$l_2 C_2 + l_3 C_{23} + \lambda_4 S_{23} = -\lambda_1 + d_z - \lambda_{67} c_z = w_2. \quad (7)$$

After squaring both sides of (6) and (7), and adding them by sides, we will obtain:

$$w_1^2 + w_2^2 = l_2^2 + l_3^2 + \lambda_4^2 + 2l_2 l_3 C_3 + 2l_2 \lambda_4 S_3.$$

After substituting $x = \tan(\Theta_3'/2), S_3 = 2x/(1+x^2), C_3 = (1-x^2)/(1+x^2), a = 2l_2 l_3,$
$b = 2l_2 \lambda_4, c = w_1^2 + w_2^2 - l_2^2 - l_3^2 - \lambda_4^2$ we will obtain:

$$a(1-x^2) + 2bx = c(1+x^2) \quad \Rightarrow \quad \Theta_3^* = 2\arctan\frac{b \pm \sqrt{a^2 + b^2 + c^2}}{a+c} \quad (8)$$
$$\Theta_3' = \Theta_3^*.$$

3.1.3 Computing Θ_2'

The range of the Θ_2' angle changes is $-70° \leq \Theta_2' \leq 70°$. From (6) and (7) as well as $S_{23} = S_2 C_3 + C_2 S_3$ and $C_{23} = C_2 C_3 - S_2 S_3$ we receive the system of equations (9) for S_2 and C_2 along with (10) for the variable Θ_2'.

$$\left. \begin{array}{c} -l_2 S_2 - l_3 S_2 C_3 - l_3 C_2 S_3 + \lambda_4 C_2 C_3 - \lambda_4 S_2 S_3 = w_1 \\ \\ l_2 C_2 + l_3 C_2 C_3 - l_3 S_2 S_3 + \lambda_4 S_2 C_3 + \lambda_4 C_2 S_3 = w_2 \end{array} \right\} \Rightarrow$$

$$S_2 = \frac{w_1(l_2 + l_3 C_3 + \lambda_4 S_3) - w_2(-l_3 S_3 + \lambda_4 C_3)}{-[l_2^2 + l_3^2 + \lambda_4^2 + 2l_2(l_3 C_3 + \lambda_4 S_3)]}, \quad (9)$$

$$C_2 = \frac{-w_1(-l_3 S_3 + \lambda_4 C_3) - w_2(l_2 + l_3 C_3 + \lambda_4 S_3)}{-[l_2^2 + l_3^2 + \lambda_4^2 + 2l_2(l_3 C_3 + \lambda_4 S_3)]},$$

$$\Theta_2' = \arctan \frac{S_2}{C_2}. \qquad (10)$$

3.2 Computing Joint Variables $\Theta_4' \div \Theta_6'$

The matrix $\mathbf{T}_6^1$, which occurs in (5), describes the position and orientation of the frame $x_6 y_6 z_6$ in the relation to the frame $x_1 y_1 z_1$. Let's denote as $_6^1 Rot$ the matrix, that describes the orientation from the matrix $\mathbf{T}_6^1$. We get the matrix $_6^1 Rot$ by zeroing the (1,4)÷(3,4) elements from the matrix $\mathbf{T}_6^1$. Likewise let's denote by $_6^1 Rot_{req}$ the matrix, that describes the orientation from the matrix $\mathbf{T}_{6\,req}^1$ in (5). Now we can obtain (11) as a result from (5).

$$_6^1 Rot = {}_6^1 Rot_{zad},$$
$$_6^1 Rot = \mathrm{Rot}(z, \Theta_{23}')\mathrm{Rot}(x, \Theta_4')\mathrm{Rot}(z, \Theta_5')\mathrm{Rot}(x, \Theta_6')\mathbf{P},$$
$$\mathbf{P} = \begin{bmatrix} 0\,0\,1\,0 \\ 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,0\,1 \end{bmatrix}. \qquad (11)$$

After applying the matrix $\mathrm{Rot}_{x'z'x'}(\Theta_4', \Theta_5', \Theta_6') = \mathrm{Rot}(x, \Theta_4')\,\mathrm{Rot}(z, \Theta_5')\,\mathrm{Rot}(x, \Theta_6')$ [2] into (11) we get:

$$\mathrm{Rot}_{x'z'x'}(\Theta_4', \Theta_5', \Theta_6') = \mathrm{Rot}(z, -\Theta_{23}')_6^1 Rot_{req}\mathbf{P}^{-1}, \qquad (12)$$

$$\mathrm{Rot}_{x'z'x'}(\Theta_4', \Theta_5', \Theta_6') =$$
$$= \begin{bmatrix} C_5 & -S_5C_6 & S_5C_6 & 0 \\ C_4S_5 & C_4C_5C_6 - S_4S_6 & -C_4C_5S_6 - S_4C_6 & 0 \\ S_4S_5 & S_4C_5C_6 + C_4S_6 & -S_4C_5S_6 + C_4C_6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (13)$$

We can calculate $\Theta_{23}' = \Theta_2' + \Theta_3'$ from (8) and (10). Likewise we can calculate $_6^1 Rot_{req}$ from the matrix $T_{6\,req}^1$. Therefore, before computing the variables Θ_4', Θ_5', Θ_6', we can calculate the left side of (12) and put it in the form of (14).

$$\mathrm{Rot}(z, -\theta_{23})_6^1 Rot_{zad}\mathbf{P}^{-1} = \begin{bmatrix} AX\ BX\ CX\ 0 \\ AY\ BY\ CY\ 0 \\ AZ\ BZ\ CZ\ 0 \\ 0\ \ 0\ \ 0\ \ 1 \end{bmatrix},$$

$$\begin{bmatrix} AX\ BX\ CX\ 0 \\ AY\ BY\ CY\ 0 \\ AZ\ BZ\ CZ\ 0 \\ 0\ \ 0\ \ 0\ \ 1 \end{bmatrix} = \begin{bmatrix} C_{23}(C_1 c_x + S_1 c_y) + S_{23}c_z \\ -S_{23}(C_1 c_x + S_1 c_y) + C_{23}c_z \\ S_1 c_x - C_1 c_y \\ 0 \end{bmatrix} \qquad (14)$$

$$\begin{matrix} C_{23}(C_1 a_x + S_1 a_y) + S_{23}a_z & C_{23}(C_1 b_x + S_1 b_y) + S_{23}b_z & 0 \\ -S_{23}(C_1 a_x + S_1 a_y) + C_{23}a_z & -S_{23}(C_1 b_x + S_1 b_y) + C_{23}b_z & 0 \\ S_1 a_x - C_1 a_y & S_1 b_x - C_1 b_y & 0 \\ 0 & 0 & 1 \end{matrix}.$$

The methodology of the computing the variables Θ_4', Θ_5', Θ_6' depends on the sum $AY^2 + AZ^2$ for the matrix (14).

3.2.1 Computing $\Theta_4' \div \Theta_6'$ for $AY^2 + AZ^2 > 0$

From the comparison of the matrices (13) and (14) we obtain (15)$\div$(18). Equation (15) for $-150° \leq \Theta_4' \leq 150°$ has a following form:

$$\Theta_4' = \begin{cases} \Theta_4^* & \text{or} \quad \Theta_4^* - 180° & \text{for} \quad \Theta_4^* > 0°, \\ \Theta_4^* & & \text{for} \quad \Theta_4^* = 0°, \\ \Theta_4^* & \text{or} \quad \Theta_4^* + 180° & \text{for} \quad \Theta_4^* < 0°, \end{cases} \tag{15}$$

$$\Theta_4^* = \arctan \frac{AZ}{AY}.$$

From (16) we can calculate $-115° \leq \Theta_5' \leq 115°$.

$$\Theta_5' = \begin{cases} \Theta_5^* & \text{for} \quad AX \geq 0, \\ \Theta_5^* - 180° & \text{for} \quad AX < 0 \quad \text{and} \quad C_4 AY + S_4 AZ \leq 0, \\ \Theta_5^* + 180° & \text{for} \quad AX < 0 \quad \text{and} \quad C_4 AY + S_4 AZ \geq 0, \end{cases} \tag{16}$$

$$\Theta_5^* = \arctan \frac{C_4 \times AY + S_4 \times AZ}{AX}.$$

From (17)–(18) we can calculate $-300° \leq \Theta_6' \leq 300°$.

$$S_6 = -S_4 \times BY + C_4 \times BZ, \quad C_6 = -S_4 \times CY + C_4 \times CZ, \tag{17}$$

$$\Theta_6^* = \arctan \frac{S_6}{C_6},$$

$$\Theta_6' = \begin{cases} \Theta_6^* & \text{or} \quad \Theta_6^* - 360° & \text{for} \quad S_6 > 0 \quad \text{and} \quad C_6 > 0, \\ \Theta_6^* & & \text{for} \quad S_6 = 0 \quad \text{and} \quad C_6 > 0, \\ \Theta_6^* & \text{or} \quad \Theta_6^* - 360° & \text{for} \quad S_6 < 0 \quad \text{and} \quad C_6 > 0, \\ \Theta_6^* - 180° & \text{or} \quad \Theta_6^* + 180° & \text{for} \quad C_6 \leq 0. \end{cases} \tag{18}$$

3.2.2 Computing $\Theta_4' \div \Theta_6'$ for $AY^2 + AZ^2 = 0$

From (12), the matrices (13) and (14) we can conclude, that $AY^2 + AZ^2 = S_5^2 = 0$, it means, that $S_5 = 0$. For Θ_5' in a rage of $-115° \leq \Theta_5' \leq 115°$ we obtain solution:

$$\Theta_5' = 0°. \tag{19}$$

That is why in the matrix (13) we need to set $S_5 = 0$ and $C_5 = 1$. After such substitutions only elements (2,2), (2,3), (3,2), (3,3) still depend on Θ_4' i Θ_6'. Other elements are fixed and equal to 0 or 1. From such form of the matrix (13) it is impossible to compute the angles Θ_4' and Θ_6' simultaneously. After simplification the elements (2,2) and (3,3) are equal to C_{46}, element (3,2) is equal to S_{46}, and element (2,3) to

$-S_{46}$. Therefore, after such simplifications, we can calculate from the matrices (13) and (14) only a sum $\Theta'_{46} = \Theta'_4 + \Theta'_6$. This situation causes the angles Θ'_4 and Θ'_6 to have infinite solutions. The sum $-450° \leq \Theta'_{46} \leq 450°$ can be computed from (20).

$$
\begin{aligned}
&\Theta^*_{46} = \arctan \frac{-CY}{CZ}, \\
&\text{for } CY = 0 \text{ and } CZ > 0 \quad \Theta'_{46} = 0° \text{ or } 360°, \\
&\text{for } CY \neq 0 \text{ and } CZ \geq 0 \quad \Theta'_{46} = \Theta^*_{46} \text{ or } \Theta^*_{46} - 360° \text{ or } \Theta^*_{46} + 360°, \\
&\text{for } CZ < 0 \quad \Theta'_{46} = \Theta^*_{46} - 180° \text{ or } \Theta^*_{46} + 180°.
\end{aligned}
\tag{20}
$$

4 Examplery Calculations

The equations presented in the article served as a base for *inverse_kinematics* program written in Matlab. The program computes the joint variables corresponding to matrix $\mathbf{T}_{req}$. For matrix (21) the program computed five possible sets of solutions.

$$
\mathbf{T}_{6\,req} = \begin{bmatrix}
0.8029 & -0.4568 & -0.3830 & -242.1596 \\
0.1050 & 0.7408 & -0.6634 & -419.4327 \\
0.5858 & 0.4924 & 0.6428 & 1648.1849 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\tag{21}
$$

Three sets $(\Theta'_1, \Theta'_2, \Theta'_3, \Theta'_{46}, \Theta'_5)$ are following: $(-120°, 20°, 20°, -40°, 0°)$, $(-120°, 20°, 20°, -400°, 0°)$, $(-120°, 20°, 20°, 320°, 0°)$.

Additional two sets $(\Theta'_1, \Theta'_2, \Theta'_3, \Theta'_4, \Theta'_5, \Theta'_6)$ are: $(60°, 48.36°, 41.71°, 0°, -49.93°, -220°)$, $(60°, 48.36°, 41.71°, 0° - 49.93°, 140°)$.

First three sets contain the sums $\Theta'_{46} = \Theta'_4 + \Theta'_6$. Therefore in each of those sets the variables Θ'_4 and Θ'_6 can have any value. It means, that those sets describe an infinite number of the solution sets $(\Theta'_1, \Theta'_2, \Theta'_3, \Theta'_4, \Theta'_5, \Theta'_6)$. The additional two sets contain the variables in the separate form.

The computations proves, that a difference between the elements of the matrices $\mathbf{T}_{6\,req}$ and $\mathbf{T}_6$ (calculated for those sets according to (2)) is not larger then 0.2258×10^{-12}.

5 Conclusions

The *inverse_kinematics* program allows to compute all sets of the joint variables corresponding to the given matrix $\mathbf{X}_{req}$. The sets are realisable by the manipulator IRB-1400. This program can be implemented in the terminal's programming environment connected with the IRB-1400 robot's controller by the Ethernet network. It is also possible to connect to this terminal, through the Ethernet network, the visual system, which is observing a manipulation object. After installing this system's software in the terminal it will be possible to compute the $\mathbf{X}_{req}$ matrix, which corresponds to the observed manipulation object. Integration of such a software with the *inverse_kinematics* program will enable to transmit the joint variables between the

terminal and controller to obtain an observed point in the manipulator workspace, without necessity of manually bringing the manipulator to this point. Nowadays it is done by manually presetting the manipulator in desired location and reading the joint variables from the terminal.

Therefore, the *inverse_kinematics* program is one of the essential elements of the future computational intelligence of the IRB-1400 robots.

References

1. Abb flexible automation: Product specification IRB 1400
2. Craig, J.J.: Introduction to Robotics, Mechanics and Control, ch. 2–3. Addison-Wesley Publishing Company, New York (1989)
3. Jezierski, E.: Dynamika i sterowanie robotów, ch. 2. Wydawnictwa Naukowo-Techniczne, Warsaw, Poland (2006)
4. Kozłowski, K., Dutkiewicz, P.: Modelowanie i sterowanie robotów. ch. 1. Państwowe Wydawnictwa Naukowe, Warsaw (2003)
5. Szkodny, T.: Modelowanie i symulacja ruchu manipulatorów robotów przemysłowych. ch. 2. Wydawnictwo Politechniki Śląskiej, Gliwice, Poland (2004)
6. Szkodny, T.: Zbiór zadań z podstaw robotyki, ch. 2. Wydawnictwo Politechniki Śląskiej, Gliwice, Poland (2008)
7. Tadeusiewicz, R.: Systemy wizyjne robotów przemysłowych. Wydawnictwa Naukowo-Techniczne, Warsaw, Poland (1992)
8. Węgrzyn, S.: Podstawy Automatyki. Wydawnictwa Naukowo-Techniczne, Warsaw, Poland (1978)

Factors Having Influence upon Efficiency of an Integrated Wired-Wireless Network

Bartłomiej Zieliński

Abstract. The paper considers the problem of an integration of the wired and wireless network segments. As characteristics of such segments differ, protocols for both segments must in many cases differ as well. Thus, protocol conversion is necessary. A protocol converter is an autonomous microprocessor-based device. A good example of such solution is a TNC controller used, among others, in amateur Packet Radio network. Several types of TNC controllers are available that differ in processing power. Additionally, TNC control software sometimes introduces some limitations and protocol implementation details that also have some influence upon operational network parameters visible to the user. Presented results have been achieved in an experimental Packet Radio network and compared to the theoretical estimations.

Keywords: netwrok integration, protocol conversion, efficiency estimation.

1 Introduction

Wireless transmission may be an interesting alternative to a wired transmission of voice or data. In the era of cellular telephony this thesis does not require any further confirmation. Similarly, last years have shown that wireless local area networks may also become very popular; recent research is focused on personal area networks (not excluding remote control networks) and broadband access (metropolitan) networks [3].

Using wireless transmission media, one can provide connectivity in the situations, when usage of wires is inconvenient, or even impossible [4]. It comprises, for example, large areas not equipped with communication infrastructure. On the other hand, wireless media can be used in city centres to provide connectivity

Bartłomiej Zieliński
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: `Bartlomiej.Zielinski@polsl.pl`

between buildings or within buildings. Wireless transmission, especially based on radio waves, is also the only reasonable choice when the stations in the network must be mobile. Wireless networks are also more flexible than wired ones in terms of network capacity [3].

We must however take into account that most of the computer networks are still wired ones. Thus, an interesting issue of network integration arises. It has been shown [4] that it can be solved using a microprocessor-based so-called protocol converter. A good example of such a converter is a so-called Terminal Node Controller (TNC) that can be seen as a network interface for amateur Packet Radio network [2]. It converts data format from RS-232 serial port into AX.25 protocol belonging to HDLC family.

2 Efficiency Analysis

AX.25 [1] protocol is an HDLC derivative. Its behavior can be adjusted using a number of parameters. The most important ones from the point of view of effective transmission speed seem the following:

- R_{wl} – wireless link transmission rate;
- N_1 – maximum capacity of data field in a frame; not larger than 256;
- k – window size (maximum number of commonly acknowledged frames); in most software versions not larger than 7;
- T_2 – time that elapses between the end of latest I frame and acknowledge; depending on software, may be set manually or automatically; typically, does not occur when maximum window size is used;
- T_{102} – slot duration for carrier sense-based collision avoidance; typically about 50 to 300 ms;
- T_{103} – time for stabilisation of transmitter parameters after transmission start and for carrier detection in receiver; depends on radio transceiver capabilities and varies from few tens to few hundreds milliseconds;
- p – persistence parameter of carrier sense-based collision avoidance (T_{CS}); typically equal to 64 which means transmission probability of 25%.

Some of these parameters are explained in Fig. 1.

On the other hand, TNC controller – as a microprocessor based device – may also have influence upon effective transmission speed. There are several TNC types available that are built using various microprocessors running at various clock frequencies. Additionally, several software types and versions exist that may differ from the point of view of network efficiency.
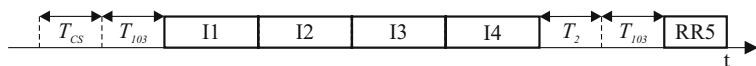
Fig. 1 AX.25 protocol frame exchange rules for $k = 4$

2.1 Theoretical Analysis

When using TNC controllers, transmission between computers runs in three stages, as shown in Fig. 2. The middle stage duration $T_p = T_{TR} - T_{tr}$ corresponds to wireless transmission depends on AX.25 protocol parameters and TNC capabilities, while the duration of $T_a = T_{TR} - T_{CT}$ and $T_z = T_{rc} - T_{tr}$ depend on RS-232 serial link effective speed only.
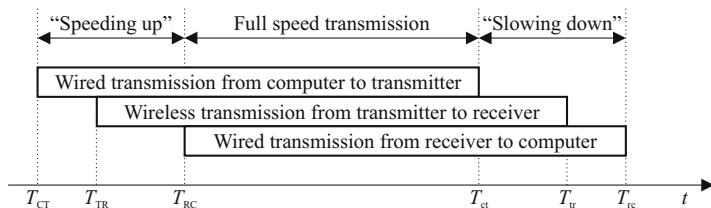
Fig. 2 Transmission stages using TNC controllers

It can be proved [6] that transmission time of an L_D-bytes long information using AX.25 protocol on half-duplex radio link can be expressed as:

$$T_p = \left\lceil \frac{L_D}{kN_1} \right\rceil \left(\frac{256T_{102}}{2(p+1)} + 2T_{103} + T_2 + \frac{63(8kN_1 + 160(1+k))}{62R_{wl}} \right). \qquad (1)$$

It is worth notice that when L_D is sufficiently large, or effective transmission speed of radio link is low, T_p dominates, thus T_a and T_z are in most cases negligible [5].

2.2 Experimental Results

Tests of TNC controllers efficiency were conducted in an experimental network, containing one or two PC-class computers and two controllers. A single PC computer is sufficient if it is equipped with two RS-232 ports or USB, depending on controllers used in a given test. The controllers were connected with cables. This unusual – as for circuits designed for wireless communication – configuration was chosen in order to avoid negative influence of radio interference upon transmission quality. Besides, in such a network it was possible to set any parameter values freely, which allowed for testing cases not very common in practice. As the radio transceiver is always under full control of TNC, lack of transceiver does not influence on transmission time. Network configuration is shown in Fig. 3, while microprocessors used in selected TNC controllers, and their clock frequencies f_{clk}, are listed in Table 1.

The tests were conducted, transmitting a file of 8 or 16 KB (depending on transmission rate) for various values of window size and data field capacity of AX.25 protocol information frame. AX.25 protocol was set up to the maximum theoretical
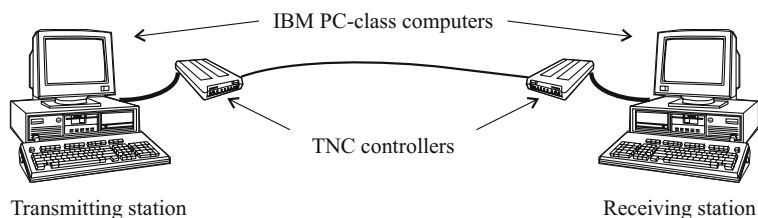
Fig. 3 Experimental network configuration

Table 1 Microprocessors in TNC controllers used during experimental tests

Controller	Microprocessor	f_{clk} [MHz]
TNC2	Z80	2.4576
TNC2D	Z80	4.9152
Spirit-2	Z80	19.6608
KPC-9612+	68HC11	16.0000
TNC3	68302	14.7456
TNC7	LPC2106	58.9824
DLC7	S3C4530	49.1520

efficiency ($k = 7, N_1 = 256$). The controllers operated in half-duplex Asynchronous Balanced Mode, as is typical for Packet Radio communication.

2.2.1 Hardware Comparison

For hardware comparison, several TNC types were selected. They are built using various microprocessor types running at various clock frequencies (Table 1) and they ran under the control of a default software updated, if possible, to the latest available version. They were connected in pairs, i.e., transmitter and receiver were of the same type.

Measurements results of effective transmission speed for few selected TNC controllers, operating at various window sizes and maximum length data frames ($N_1 = 256$) are shown in Fig. 4. Transmission rate was equal to 19.2 kbps on serial port and 1.2 kbps on radio link. For comparison, the graph contains also a curve determining theoretical capabilities of AX.25 protocol. On the graph we can see that the results do not differ very much. Some controllers (e.g., TNC2, TNC2D) can not make use of window size 4 and above – increasing this parameter does not practically increase transmission speed. KPC-9612+ behaves similarly. Faster TNC3 and TNC7 controllers, unexpectedly, behave worse than the others for window sizes less than 7. A more detailed analysis conducted in monitoring mode showed that these controllers did not request immediate acknowledgement by setting Poll/Final bit in AX.25 protocol control field[1]. Thus, the recipient waits for T_2 time for

---

[1] Such behaviour is neither forbidden nor required by AX.25 protocol specification.
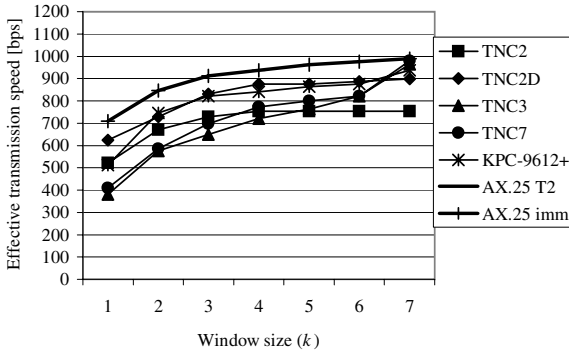
Fig. 4 Effective transmission speed on 1200 bps radio link

possible consecutive frames and sends the acknowledgement only afterwards. However, when window size equals to 7, TNC3 and TNC7 achieve higher throughput than other controllers, close to theoretical values. It is possible, because, for maximum window size allowed by protocol definition, recipient does not have to wait T_2 time.

Results of similar measurements, conducted at radio link transmission rate of 9.6 kbps, are presented in Fig. 5. In this case, difference between various TNC controllers is much more visible. In general, effective speed is higher for TNC's built using faster microprocessors. Depending on controller type, maximum effective speed varies from about 1.5 kbps (TNC2D) to almost 5 kbps (TNC7). For comparison, the graph presents two curves showing theoretical capabilities of AX.25 protocol with immediate acknowledge generation (AX.25 imm) or with T_2 delay (AX.25 T2).

Presented results show that effective transmission speed achieved in a given configuration depends on not only hardware – particularly microprocessor type and its clock frequency – but also properties of software that controls TNC controller
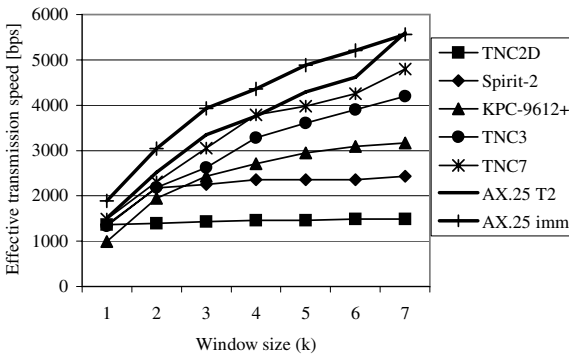
Fig. 5 Effective transmission speed on 9600 bps radio link

operation. The following factors, depending exclusively on software, may have influence over circuit performance:

- full utilisation of window size for every data field capacity of a frame,
- sufficiently high AX.25 protocol frames processing speed,
- immediate acknowledgement of proper information frame reception,
- immediate acknowledgement request by setting of Poll/Final bit in the latest information frame within a window.

Inability of full utilisation of window size is especially annoying in Z80-bases TNC controllers, practically regardless of its clock frequency and memory capacity. However, type and version of software used in TNC has some influence upon its performance.

2.2.2   Software Comparison

For Z80-based TNC controllers, there are several control software types and versions available. It allows determine how software used in a given TNC controller influences on effective throughput. During the tests, TNC2 and TNC2D controllers were used. They ran under the control of MFJ software (1.1.4, 1.1.9, and Muel versions), TF (2.1, 2.3, and 2.7 versions) and Spirit (5.0 version). These controllers acted as transmitters or receivers. TNC3 and TNC7 controllers were used on the other side. It was assumed that their computing power was sufficiently higher than that of any Z80-based TNC, so they would not slow down the transmission significantly. Indeed, the most visible differences are for transmission from TNC2 or TNC2D to either of fast receivers. These results are shown in Figs. 6 and 7, respectively.

In the presented graphs one can see that MFJ software versions do not differ very much in terms of effective transmission speed. Spirit software behaves similarly. Effective transmission speed grows rapidly when window size (k) increases from 1 to 4. However, further increasing of k does not bring significant
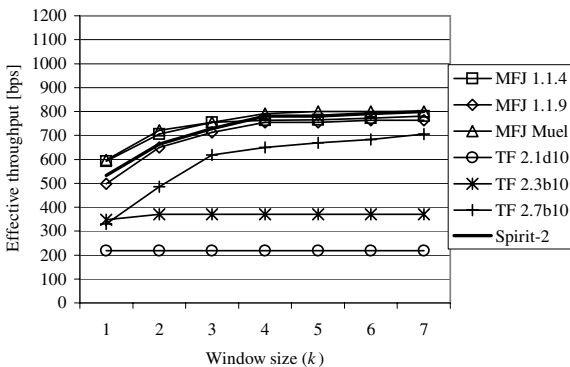
Fig. 6 Effective transmission speed for various software for TNC2 controller (2.4956 MHz)
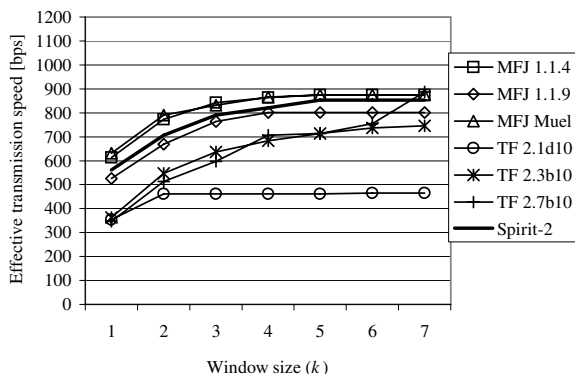
Fig. 7 Effective transmission speed for various software for TNC2D controller (4.9152 MHz)

improvement of transmission speed. Probably there are some limitations in the software, e.g., buffer capacity, that does not allow to transmit, in average, more than 4 maximum-length frames. Maximum achievable effective transmission speed varies from about 750 to 800 kbps for slower TNC and from 800 to 900 kbps for the faster one.

TF software behaves completely different. Version 2.1 seems the most ineffective in the role of the sender and the achievable transmission speed is independent of window size. Similar is version 2.3, however, only when run on the slower TNC. Thus, one might conclude that it requires more processing power than this TNC can offer. Version 2.7 is the best one regardless of TNC speed. However, when run on the faster one, it achieves the same results as MFJ software for $k = 7$.

3 Conclusions

Presented results show that effective transmission speed achievable between end devices in an integrated wired-wireless network depend on several factors. First of all, wireless link protocol has several parameters that have influence upon its efficiency. Some of these parameters are transmission hardware dependent so we cannot adjust them freely. In any case, knowing protocol properties we can estimate the effective transmission speed on a radio link using analytical methods. Results achieved this way, unfortunately, do not take into account properties of a transmission hardware, such as computing power or memory capacity that are not unlimited. However, hardware is not the only thing that may confine effective transmission speed. As the experimental results show, software that runs on a TNC controller also plays a significant role from the point of view of network efficiency. This results from software-dependent protocol implementation details and some possible limitations found in some software types and versions.

References

1. Beech, W.A., Nielsen, D.E., Taylor, J.: AX.25 Link Access Protocol for Amateur Packet Radio. Tucson Amateur Packet Radio Corporation, Tucson, US (1997)
2. Karn, P.R., Price, H.E., Diersing, R.J.: Packet radio in the amateur service. IEEE Journal on Selected Areas Communications 3(3), 431–439 (1985)
3. Tanenbaum, A.: Computer Networks, 4th edn. Prentice Hall, Upple Saddle River (2003)
4. Zieliński, B.: Wireless computer networks using protocol conversion. Ph.D. thesis, Silesian University of Technology, Gliwice, Poland (1998) (in Polish)
5. Zieliński, B.: An analytical model of TNC controller. Theoretical and Applied Informatics 21 (in press, 2009)
6. Zieliński, B.: Efficiency estimation of AX. 25 protocol. Theoretical and Applied Informatics 20(3), 199–214 (2008)

FFT Based EMG Signals Analysis on FPGAs for Dexterous Hand Prosthesis Control

Jacek Góra, Przemysław M. Szecówka, and Andrzej R. Wołczowski

Abstract. This paper relates to the research on a dexterous hand prosthesis conducted at the Wroclaw University of Technology. The possibility of recognizing human intention by digital processing of electromyographic (EMG) signals is presented. The realization of the decision-making logic in specialized digital circuits is investigated into. The article specifically deals with the issue of the Fast Fourier Transform (FFT) algorithm floating-point implementation.

Keywords: EMG, FPGA, FFT, signal processing.

1 Introduction

The dexterity of a human hand is an outcome of both mechanical properties and the performance of its control system – the central nervous system. Construction of a highly advanced multi-joint anthropomorphic prosthesis is not a problem at the present state of the art. Many such constructions have been presented in literature on the subject [2]. However, the problem of developing applicable control system still remains unresolved.

Jacek Góra
Nokia Siemens Networks, RTP Wrocław, Poland
e-mail: jacek_gora@o2.pl

Przemysław M. Szecówka
Faculty of Microsystem Electronics and Photonics, Wrocław University of Technology,
Wrocław, Poland
e-mail: przemyslaw.szecowka.pwr.wroc.pl

Andrzej R. Wołczowski
Faculty of Electronics, Wrocław University of Technology,
Wrocław, Poland
e-mail: andrzej.wolczowski@pwr.wroc.pl

In the perfect situation the prosthesis reacts on the user's will of action, just as it takes place in the natural hand. However, as it is difficult and inefficient to use neural impulses to drive an artificial mechanism, a different approach is required. The practice is to use other signals that normally accompany hand's movement. During muscle activation, small electric potentials appear. Their measurement and analysis is called electromyography (EMG). Using EMG signals for driving prosthesis is very feasible since many of the muscles controlling the movement of fingers are located in the forearm, and are still available after hand's amputation. Additionally, they can be detected in an non-invasive way, which is a big advantage for the comfort of use.

The dexterity of an artificial hand requires the ability to perform various movements. The EMG signals processing gets more complicated as the prosthesis's number of degrees of freedom increases. Recognition of user's intention with a very low error probability is required in order to achieve high reliability and comfort of use. To increase the chance of a proper classification, signals from many electrodes can be observed and analyzed simultaneously. This, however, places greater emphasis on the processing capabilities of the decision-making logic. In this paper some solutions for designing digital circuitry based on Field Programmable Gate Arrays (FPGA) are presented. Special focus has been put on implementation of the Fast Fourier Transform (FFT) algorithm that can be very useful in the EMG signals analysis process.

2 EMG Signals Analysis

The recognition of user's intention comprises three stages [5]: the acquisition of EMG signals; extraction of features differentiating the movements; assigning measured signals to proper types of movement. Each stage has an influence on the quality of the whole process and the functionality of the prosthesis.

2.1 Signals Acquisition

EMG signals are registered by placing electrodes on the surface of the skin above tested muscles. This method is less precise than the invasive one, but it is preferred for its convenience and simplicity. To reduce the influence of interferences in the acquisition process on the overall error, differential measurement and high precision A/D conversion are recommended. At the Wroclaw University of Technology an 8-channel system is used. Coupled active electrodes are connected to a ADC board through high CMRR differential amplifiers. Registered EMG signals in a digital form can be transmitted directly to a decision-making logic or stored in a data base for later analysis [5, 4].

2.2 Features Extraction

Extraction of features is the first step of the EMG digital processing. Proper selection of parameters that describe movements is highly relevant for their distinction. They

should be selected so as to allow proper intention recognition. The number of factors cannot be too high because of the limited time available for the processing and the performance of the decision-making logic [5].

2.3   Signals Classification

Hand grasps can be divided into several classes (Fig. 1). The basic task of the logic circuit is to establish to which of the types the measured EMG signals belong. In order to do so, extracted vectors of features are compared with the reference data representative for each class (comparative method), or are put on inputs of a neural network that has been taught on such vectors (heuristic method) [3].

Concentric   Cylindrical   Pliers   Suitcase   Lateral
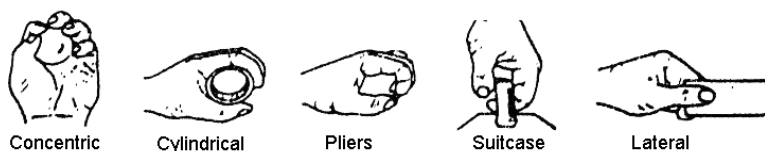
Fig. 1 Exemplary hand grasps classes

To attain the prosthesis functionality comparable with a natural hand's, the analysis of EMG signals has to be done in a real time. It is assumed that in order to achieve this goal, a maximum acceptable delay between the user's attempt of action, to the mechanism's reaction is 300 ms [6]. In that time samples from all the channels have to be gathered, converted to a digital form and the whole digital processing has to be done producing a final, precise result in form of control commands for the prosthesis mechanism.

3   The Experiment

To confirm the applicability of the Fourier transform in the EMG signals analysis an experiment has been conducted. Signals from 4 pairs of differential electrodes have been registered for three classes of hand movements: pliers, suitcase and concentric grasp. Each gesture has been repeated several times. Measurements have been analysed using 256-point Fourier transform [3]. Afterwards, the results have been compared within and between classes.

Figure 2 shows measured EMG signals for pliers grasp. Similar diagrams have been obtained for other classes of hand movements. The sampling frequency was 1 kHz. Data from each channel has been divided into 256-samples long frames and analysed using FFT algorithm. As a result of the Fourier transform a series of amplitude spectrums has been obtained for every channel.

The FFT spectrum is a compound of two symmetric parts. Utilizing this property, only half of each spectrum (128 samples per channel) can been used in further
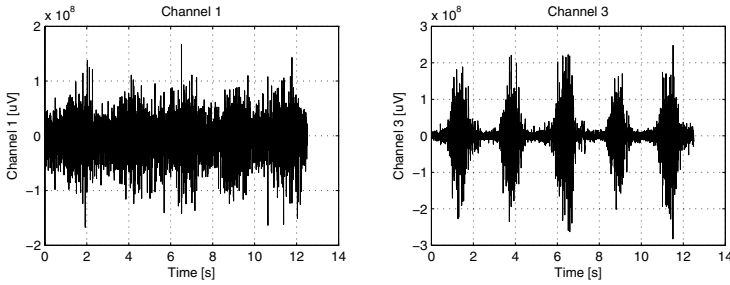
Fig. 2 Measured EMG signals for pliers grasp (channels 1 and 3)
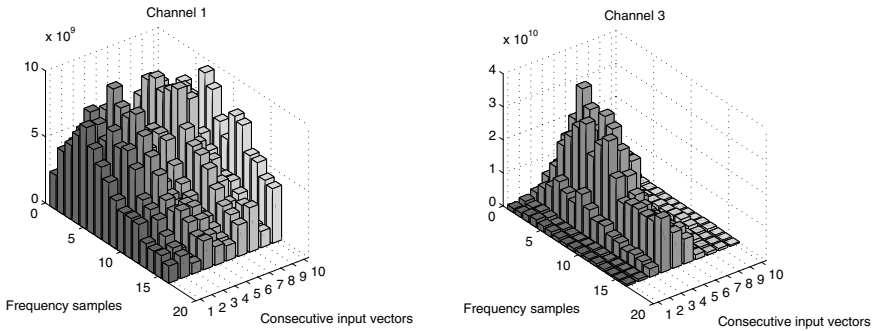
Fig. 3 Extracted vectors of features for the pliers movement (channels 1 and 3)

analysis without loss of information. In spite of it, the amount of data after the reduction (512 samples for 4 channels) is still too large to be used as an input vector for the decision-making logic. Averaging every 16 samples results in 64 samples long input vectors. This allows conducting fast and reliable analysis of EMG signals. Additionally, using in the analysis the data registered earlier results in an increase in the probability of a proper identification (sequential classification[3]). Figure 3 presents exemplary vectors obtained for the pliers grasp.

Received series of input vectors have been further analysed. Correlation coefficient has been used as an indicator of similarity between registered EMG signals. Comparison of data belonging to the same movement classes resulted in values of coefficients ranging from 0.82 to 0.88 and from 0.69 to 0.79 for signals belonging to different classes. These data confirm that presented approach can be used as a part of EMG signals analysis process.

4   Hardware Realization of EMG Processing Algorithms

High performance signal processing logic can be created basing on a few technologies. The most common are Digital Signal Processors (DSP) and Programmable Logic Devices (PLD). In this paper several solutions targeted for Field Programmable

Gate Arrays (FPGA) are presented. This technology has been chosen because it allows creating efficient, application specific constructions.

The following section presents a hardware realization of the FFT algorithm used in the experiment. The implementation has been prepared in a way that allows maximum flexibility. The length of transform and the number of arithmetic blocks working in parallel can be easily changed to meet user's requirements. In the basic configuration 256-point FFT with 4 floating-point arithmetic blocks has been used [1]. The presented designs have been prepared targeting the Xilinx Spartan-3A DSP XC3SD1800A device but can be successfully implemented in other FPGAs. In the following paragraphs compound elements of the FFT design have been described.

4.1 Floating Point Representation

Fixed-point devices constitute most of the digital arithmetic circuits that are currently in use. They are able to do calculations on numbers represented in a machine code with a constant precision and range. However, it is often required to work in one process on both very small and very big numbers or the results of calculations can exceed the range of accepted data format. In such situations, fixed-point circuits show lack of flexibility and precision.

The solution for this problem is using floating-point arithmetics. The advantage of this type of coding is that there is always a constant number of significant digits in the representation and the range of the supported values is much wider than for a fixed-point representation of the same size:

$$x = s \times b^e, \tag{1}$$

where s – value of the significand, b – base, e – exponent.

Making calculations on floating-point numbers requires higher processing capabilities than in the case of a fixed-point system. It is so because both the significand and the exponent (1) have to be processed separately. That is why this type of representation has not been very popular in PLDs so far. However, the developments that recently occurred in the FPGA technology, especially increase in size and implementation of hardware arithmetic blocks, have made it practically possible to design fixed-point circuits.

There are many floating-point formats in use. The most popular one is the IEEE 754 standard. It is highly precise and can represent numbers from a wide range. It has also a long representation and therefore requires extended arithmetic circuits and a lot of memory for storage. During projects conducted at the Wroclaw University of Technology it has been established that a modified, shorter format would be more useful for most of the applications (6 bit long exponent, 9 or 18 bit long significand for single- and double-precision format respectively and the base is 8)[1, 6]. The proposed format is less precise and can represent narrower range of values, but the gain in simplification of the functional logic is considerable. This allows creating much smaller and faster designs without noticeable loss in accuracy.
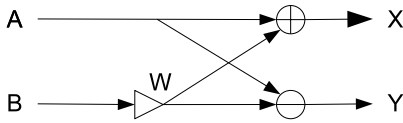
Fig. 4 Structure of the computational block ('butterfly')

Basing on the proposed floating-point format, several functional blocks have been designed. Circuits have been prepared for both single- and double-precision. The overall maximum error has been estimated to about 1.37% and 0.003% for the single- and double-precision calculations respectively. Using format with a longer significand results in a lower error level but in a bigger size of functional blocks, as well. Due to those properties, the double-precision format is intended to be used in the parts of circuits in which error cumulation can occur, for example in accumulators.

4.2 FFT Computational Block

The base building block in the simplest FFT algorithm (radix-2) is the so-called 'butterfly' (Fig. 4). In one calculation cycle two complex data are taken from memory and processed producing two complex numbers.

The calculations realized by the computational block are described by equations:

$$\text{Re}(X) = \text{Re}(A) + \text{Re}(B) \times \text{Re}(W) - \text{Im}(B) \times \text{Im}(W), \tag{2}$$

$$\text{Im}(X) = \text{Im}(A) + \text{Re}(B) \times \text{Im}(W) + \text{Im}(B) \times \text{Re}(W), \tag{3}$$

$$\text{Re}(Y) = \text{Re}(A) - \text{Re}(B) \times \text{Re}(W) + \text{Im}(B) \times \text{Im}(W), \tag{4}$$

$$\text{Im}(Y) = \text{Im}(A) - \text{Re}(B) \times \text{Im}(W) - \text{Im}(B) \times \text{Re}(W), \tag{5}$$

where X, Y – results of the 'butterfly' calculations, A and B – arguments, W – Fourier transform coefficient.
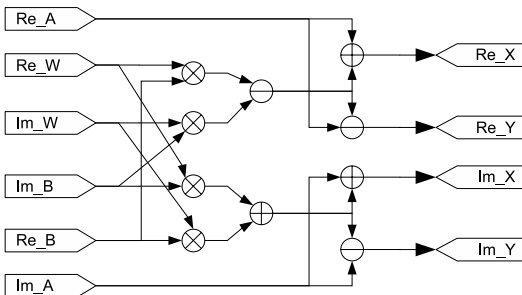
Fig. 5 Detailed structure of the designed arithmetic block

Table 1 Synthesis results of the designed computational block[1]

Parameter	Non-pipelined design	Pipelined design (12 steps)
FF (33280)	0 (0%)	770 (2%)
LUT (33280)	2702 (8%)	2691 (8%)
DSP48A (84)	4 (4%)	4 (4%)
Maximal clock frequency:	18.373 MHz	116.292 MHz

The detailed schematic of the designed computational block is presented in Fig. 5. Arithmetic components used in the project are the floating-point components described in Sect. 4.1. To increase the performance of calculations pipelining can be used. It allows processing several consecutive data at a time with a higher frequency than in the case of a non-pipelined implementation. Synthesis results in FPGA device for both variations are presented in Table 1. The pipelined design with a series of 12 registers has proven to be much more efficien than the other solution without a significant increase in used hardware elements.

4.3 FFT Implementation

In the basic configuration a 256-point FFT algorithm with 4 computational blocks working in parallel has been prepared [1]. Additionally to computational components (Sect. 4.2), control logic and some memory are needed (Fig. 6).

During the calculations data is kept in two registers. One is the source and the other is the destination register. The arguments of the calculations are taken from the former and the results are saved in the latter one. After each series of calculations the data is copied from the destination register to the source register. For the 256-point FFT it is necessary to conduct 8 series of 128 calculations. To control the proper sequence of calculations, a state machine has been used.

The synthesis results of the 256-point FFT design with 4 computational blocks in FPGA are presented in Table 2. Two variations have been tested — with pipelined and non-pipelined 'butterflies'. According to the obtained results, the prepared FFT implementation utilizes more than half of the basic hardware FPGA elements (Flip-Flops, Look-Up-Tables) and the complete time of calculation of the transform is at the level of a few microseconds.
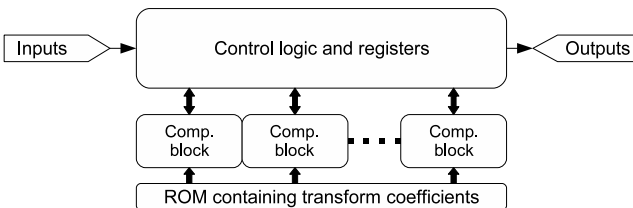
Fig. 6 Block scheme of the FFT design

Table 2 Synthesis results of the complete 256-point FFT design with 4 computational blocks [1]

Parameter	Non-pipelined design	Pipelined design (12 steps)
FF (33280)	16403 (49%)	19479 (58%)
LUT (33280)	27285 (81%)	23154 (69%)
DSP48A (84)	16 (19%)	16 (19%)
BRAM (84)	2 (2%)	2 (2%)
Maximal clock frequency:	18.373 MHz	116.292 MHz
Time of full calculations:	14.936 μs	3.030 μs

Using Xilinx Spartan-3A DSP XC3SD1800A device or similar, it is possible to implement in one FPGA efficient 256-point Fourier transform, memory to save results of analysis, and additional processing logic. The FPGA technology allows creating processing systems matched for a specific problem on a level not possible to reach for any other solution. In this paper some exemplary designs targeting digital signal processing have been presented. The designs can be successfully utilized in user intention recognition process for control of a dexterous hand prosthesis.

References

1. Góra, J.: Implementation of DSP algorithms in specialised digital circuit. Master's thesis, Wroclaw University of Technology, Wroclaw, Poland (2008)
2. Rodriguez-Cheu, L.E., Aguilar, M.E., Cuxar, A., Navarro, X., Casals, A.: Feedback of grasping forces to prosthetic hand users. Challenges for Assistive Technology 20 (2007)
3. Wołczowski, A.R., Kurzyński, M.: Control of dexterous hand via recognition of EMG signals using combination of decision-tree and sequential classifier. Advances in Soft Computing 45 (2008)
4. Wołczowski, A.R., Prociow, P.: EMG and MMG signal fusion for dexterous hand prosthesis control. Challenges for Assistive Technology 20 (2007)
5. Wołczowski, A.R., Suchodolski, T.: Bioprosthesis control: Human-machine interaction problem. Challenges for Assistive Technology 20 (2007)
6. Wołczowski, A.R., Szecówka, P.M., Krysztoforski, K., Kowalski, M.: Hardware approach to the artificial hand control algorithm realization. In: Oliveira, J.L., Maojo, V., Martín-Sánchez, F., Pereira, A.S. (eds.) ISBMDA 2005. LNCS (LNBI), vol. 3745, pp. 149–160. Springer, Heidelberg (2005)

The VHDL Implementation
of Reconfigurable MIPS Processor

Adam Ziębiński and Stanisław Świerc

Abstract. This article presents a project of an embedded system realized on a programmable matrix. The main element is a reconfigurable processor of MIPS architecture. It was implemented in the VHDL in such way that its instruction set can be reduced to the set of instructions present in the program memory. As the result a processor will contain the logic that is absolutely necessary. This solution yields a device that requires fewer gates to be synthesized in the programmable matrix and has a potential to increase the speed of the information processing performed by the system in the target FPGA.

Keywords: DSP, FPGA, MIPS, VHDL.

1   Introduction

The embedded systems are commonly described as an information processing systems embedded into a larger product dedicated towards a certain application [5].

There are may ways to create such system. One of them is the programmable matrix technology. This solution is applicable in two situations. First is when the total market demand is not big enough to cover manufacturing costs of a full custom hardware chip. Second, when there is a possibility of alteration in the device hardware configuration.

Despite the diversity of the embedded systems they all need computational power. Thus, they must contain either a microprocessor or a processor, depending on the

Adam Ziębiński
Institute of Informatics, Silesian University of Technology,
Akademicka 16, Gliwice, Poland
e-mail: adam.ziebinski@polsl.pl

Stanisław Świerc
e-mail: stanislaw.swierc@gmail.com

scale of the system. In the case of the Field Programmable Gate Array (FPGA) approach there exist at least three solutions:

- use one of the modern FPGA arrays with integrated CPU core (Xilinx's Virtex family – Power PC 405, Atmel's FPSlic – AVR),
- use an *Intellectual Property* (IP) component,
- design and implement a CPU capable of performing required task.

Although, the last option is the most time-consuming and requires some knowledge in processor's architecture. It can potentially yield better performance because the core can be optimized for a given task. It could be even possible to design the instruction set covering only instructions required for a specific algorithm.

This idea became an inspiration for for carrying out a research on design and implementation of reconfigurable processor's core for embedded applications. Configuration would be done by setting some parameters indicating which instructions should be included in the instruction set.

2   Project

The main task of the research was to implement a processor core from scratch in the VHDL (Very High Speed Integrated Circuits Hardware Description Language) in such way that its instruction set could be adjusted by a set of parameters. As a base for the research an existing architecture was adopted. Namely, it was a MIPS architecture which is considered as the most elegant solution among all effective RISC architectures [8].

At the beginning the MIPS was developed as an academic project led by John L. Hennessy at the Stanford University [6]. Shortly after the first MIPS processor was released it achieved commercial success mainly in embedded application. Even today it holds its leading position in that field. In 2004 more than 100 million MIPS CPU's were shipped into embedded systems [8].

The MIPS won its popularity thanks to pipeline processing. The earliest MIPS operated on 32-bit words and had five stages in its pipeline. Such version was chosen for the research. Figure 1 depicts its this idea.
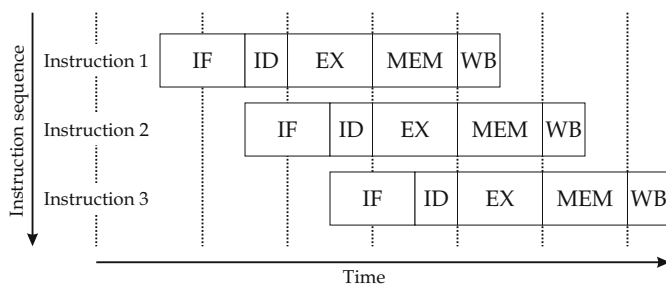
Fig. 1 The MIPS pipeline schema
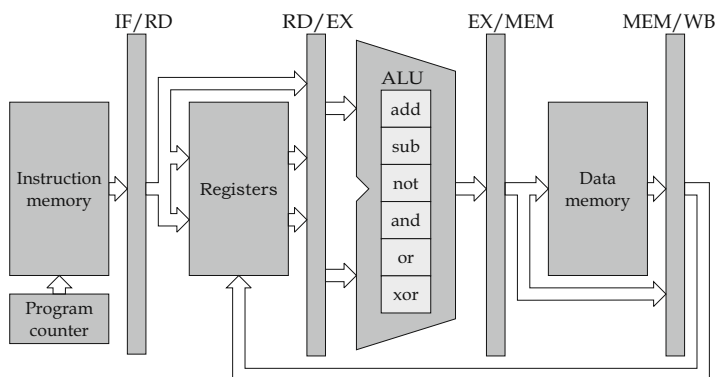
Fig. 2 The datapath

The pipeline stages have the following purposes:

- IF (instruction fetch) – gets the next instruction from the memory or cache,
- ID (instruction decode) – decodes the instruction and reads the data from the register file,
- EX (execution) – performs an arithmetical/logical operations,
- MEM (memory) – reads/writes data memory or cache,
- WB (write back) – stores the value obtained from an operation back to the register file.

This solution yields greater computational power. Such processors can perform some calculations approximately five times faster than a processor with the same logic but with single-cycle organization [6].

The processors design was created basing on the specification obtained from *MIPS Technologies* [7] and John L. Hennessy's architecture description [6]. In order to make the project synthesizable in relatively small FPGA, the instruction set was reduced to the following functional groups: load and store, ALU, jump and branch [7]. This subset is sufficient for some simple applications.

When the CPU design was ready it was analyzed in order to determine which modules take part in execution of certain instructions. Knowing the dependencies between instructions and the modules, it was possible to show how the processor should be built if it has to support a specific subset of instructions.

Figure 2 depicts the simplified datapath. Those elements which are absolutely necessary for the processor to operate and thus have to be included in the core are marked with dark grey. On the other hand the ALU is decomposed into some light grey blocks that stand for certain operations. Their presence in the system is conditional since their exclusion would affect only a small subset of the instructions.

The MIPS derives from the Harvard architecture. It has two separate busses for the instructions and the data. The communication with such processor can be based either on a cache or two separate memory modules. Assuming that typically in embedded applications programs do not change once they are loaded into devices, the
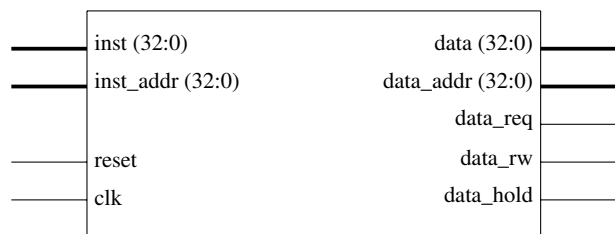
Fig. 3 The processor's interface

second solution was found to be more applicable [5]. Moreover, in the case of two separate memory modules, their organization can differ. Thus they can be better adjusted for their tasks in the system.

In order to adjust system for a specific application the following steps should be taken:

- program analysis,
- used instruction subset determination,
- processor's core configuration.

3 Implementation

The designed core was implemented in the VHDL [2, 1, 10]. It communicates with two separate memory modules. The program memory is built on the ROM cells. Whereas the data memory unit is a hybrid consisting of RAM cells and some registers of the devices' controllers mapped to the address space. Through those registers processor can control its peripherals.

The data bus communication adhere to the req-hold protocol exploiting the following ports:

`data_req` – requests bus access,
`data_rw` – sets data direction,
`data_hold` – holds the processor when the data is not ready.

The VHDL language does not offer a mechanism known from the 'C' programming language as conditional compilation. The synthesis tool process the all HDL statements present in the source file. Nevertheless, it was noticed that in the optimization stage of the synthesis all `IF` statements where the value of the condition is known and it is false are omitted. Exploiting this feature one can regulate the hardware organization of the product device.

This idea was applied in our project by introducing a set of boolean parameters. They are related with the instruction in one-to-one relationship. Therefore, the processor configuration is quite intuitive. Their values determine which modules are to be synthesized. Consequently it influences the effective instruction set. Figure 4 presents the idea of conditional synthesis.

```
ïżłENTITY cpu IS
GENERIC(
    --xor conditional parameter
    inst_xor    : boolean := false;
[...]
ARCHITECTURE behaviour OF cpu IS
    --control signal determining which operation should ALU perform
    SIGNAL func         : std_logic_vector(5 DOWNTO 0);
    --first ALU argument
    SIGNAL alu_arg1     : std_logic_vector(32 DOWNTO 0);
    --second ALU argument
    SIGNAL alu_arg2     : std_logic_vector(32 DOWNTO 0);
    --ALU product
    SIGNAL alu_prod     : std_logic_vector(32 DOWNTO 0);
    [...]
BEGIN
    [...]
    --instruction processing
    CASE func is
        [...]
        WHEN func_xor =>
            --test if the instruction should be supported
            IF inst_xor THEN
                alu_prod <= alu_arg1 XOR alu_arg2;
            ELSE
                alu_prod <= (OTHERS=>'-');
            END IF;
        WHEN OTHERS =>
            alu_prod <= (OTHERS=>'-');
    END CASE;
    [...]
END behaviour;
```

Fig. 4 The conditional synthesis

At the current stage of the research all parameters have to be set manually according to the results of the program analysis. It is worth mentioning is that the process described is based on the boolean functions evaluation. The aim of these functions is to determine if a certain instruction is present in the given machine code or not.

Finally, the system was realized on a ZL6PLD development board containing a Spartan 3 family FPGA – XC3S200 [3]. Apart from the array the board has some integrated peripherals such as: buttons, multiplexed seven-segment display (4 digits). These devices were included in the project. This operation required creation of new controllers to communicate with them.

Synthesis and array programming were done by tools integrated into *Xilinx ISE WEB PACK 9.2i* from XILINX company [9].

4 Testing

The processor was implemented in conformity with the MIPS architecture specification. Consequently, it can execute any program saved as the MIPS machine code as long as there are only instructions from the subset chosen at the design stage.

Table 1 The maximal frequency comparison

Optimization mode	Maximum Frequency [MHz]	
	Full Instruction Set	Reduced Instruction Set
Speed	34.342	37.764
Area	28.796	27.463

Table 2 The device utilization comparison

Optimization mode: Area		Full Instruction Set		Reduced Instruction Set	
	Available	Used	Utilization	Used	Utilization
Logic Utilization					
Number of Slice Flip Flops	3 840	813	21%	800	20%
Number of 4 input LUTs	3 840	2 764	71%	1 471	38%
Logic Distribution					
Number of occupied Slices	1 920	1 918	99%	1 283	66%

For the testing purposes a simple application was created in the MIPS assembly language. It calculates the value of the biggest Fibonacci number that can be stored in the 32-bit, unsigned variable. An iterative algorithm was chosen due to the stack size limits. When the result is ready it is shown at the multiplexed seven-segment display. The main operation in this problem is addition. Therefore, one may infer that in this case instruction set reduction has potential to improve performance.

The source code was translated by an application created by James R. Larus – SPIM [4]. It is a MIPS core simulator. However, it can also produce the log files with the machine code written as an ASCII string. It is convenient since the code has to be placed in the VHDL source file as a CPU entity parameter.

5 Results

The system was synthesized with two different optimization criteria: speed and area. In both cases there were two runs. At first a special, global flag was set to disable instruction set reduction. Afterwards, it was enabled. Then the results were compared against each other.

The data gathered during this process can be found in Tables 1 and 2.

The data in Table 1 shows that the instruction set reduction can influence the maximum frequency the system can work at. The highest value was reached when the reduction mechanism was enabled. If the device was to support all the instructions it would work slower by approximately 10%.

The positive effects of the instruction set reduction are more evident after Table 2 inspection. It shows how many system blocks of the array were used to synthesize the device. The total number of LUT blocks occupied decreased almost to a half of its initial value after the reduction mechanism was enabled.

Furthermore, it is worth noticing that when the device supported the complete instruction set it was hardly synthesizable utilizing 99% of slices available in the given FPGA.

6 Conclusion

It was proved that the instruction set reduction not only can decrease amount of resources present in the system but also can speed up information processing.

However, this method has of course some disadvantages. First, there are some problems in evaluation if the device works properly. Software modules should not be tested separately, owing to the fact that when the program changes the CPU may change as well. An obvious solution is to place some dummy instructions just to force their support but one may find it awkward. Second, once the system is finished any alterations may be burdensome. In order to update the program the whole core has to be synthesized. This operation may cause changes in timing characteristics of the device.

Even though the solution presented in this paper has some drawbacks, it has potential to improve performance of embedded systems based on FPGA. Therefore, the research will be continued. The next aim is to integrate the processor and the instruction memory into a single entity that could accept the program machine code as a parameter. It should be possible to force the synthesis tool to perform the program analysis and configure the core afterwards. It would make our solution more accessible.

References

1. Ashenden, P.J.: The Student's Guide to VHDL. Morgan Kaufmann, San Francisco (1998)
2. Ashenden, P.J.: The Designer's Guide to VHDL. Morgan Kaufmann, San Francisco (2002)
3. Kamami: The development board ZL6PLD,
   http://www.kamami.pl/?id_prod=2212800
4. Larus, J.R.: SPIM S20: A MIPS R2000 simulator. Computer Sciences Department, University of Wisconsin-Madison (1993)
5. Marwedel, P.: Embedded System Design. Springer, Dordrecht (2004)
6. Patterson, D.A., Hennessy, J.L.: Computer Organization and Design, 3rd edn. Morgan Kaufmann Publishers, San Francisco (2005)
7. Price, C.: MIPS IV instruction set revision 3.2, mips technologies (1995)
8. Sweetman, D.: See MIPS Run, 2nd edn. Morgan Kaufmann Publishers, San Francisco (2007)
9. Xilinx: Software and hardware for FPGA, http://www.xilinx.com/
10. Zwoliński, M.: Digital System Design with VHDL. Pearson Prentice Hall, London (2004)

Time Optimal Target Following by a Mobile Vehicle

Krzysztof Skrzypczyk

Abstract. The paper addresses the problem of mobile robot navigation in case of moving target. The solution of such problem requires the prediction of a future location of the target and fast tracking algorithm must be provided. In this work a method of time optimal target tracking is presented. The proposed method is very effective and can be applied even to very simple processing unit. Results of simulations are presented to prove an efficiency of the proposed approach.

Keywords: game theory, motion planning, prediction control, time optimality, adaptive control, human machine interaction.

1 Introduction

The tasks mobile robots are considered to perform are mainly related to a problem of moving the robot from the initial location to the desired one. An execution of the task impose serious problems on the control system. The problem is getting more complicated when an assumption is made that the navigational target is the moving one. Then the problem is not trivial even when the robot workspace is free of obstacles. This kind of problem is often named the tracking one [5, 2]. In practice such problem has to be considered especially in case of designing the system dedicated to handle human-robot interaction tasks [6]. In such case human is interpreted as the moving target the robot is intended to follow. Solving such problem needs both the prediction of movement of the target and fast tracking must be taken into account. The problem of prediction of a trajectory is well represented in the literature. The most popular approach to a solution of this problem is Kalman Filtration

Krzysztof Skrzypczyk
Institute of Automatic Control, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: `krzysztof.skrzypczyk@polsl.pl`

[1, 4]. Another one is the probabilistic approach named particle filter [3]. Unfortunately if an application of these methods to real robotic system is considered they must be simple enough. In the work by [7] simple algorithm of prediction was presented. Although it is very effective it lacks the feature of adaptation. In this paper new adaptive prediction algorithm based on the work by [7] is presented. Thanks to ability of the adaptation it is more effective and more robust in comparison to the original one. Another problem is related to tracking a given trajectory. If the trajectory is known the task is rather simple and there exist many methods of solution of the problem. But when the target must be tracked using only an estimation of its location the problem is getting more complicated especially due to the errors of the prediction method. Another issue that complicates the problem is unpredictability of the target movement. Then robust algorithm of tracking must be applied. Additionally when there is a need of minimizing the time of tracking the problem is more complicated. Although time optimal control domain is well represented in the control literature many of approaches are also too complicated to apply them in a simple robotic systems. In this paper a heuristic approach to the time optimal tracking problem is presented. An algorithm that utilizes some common-sense knowledge and prediction technique is discussed. Proposed approach gives good results what proved multiple simulations.

1.1 System Architecture Overview

Figure 1 shows a general diagram of the robot control system. The robot is assumed to have sensory system that is able to provide the location of the tracking target. These data are stored in a state buffer of length M and are used to build a target trajectory model. Using the prediction model the location of the target is computed. Next minimal time needed to reach the target is determined. Next stage of the work of the system is to plan tracking trajectory of the robot. During an execution of current motion plan the errors of prediction are monitored. In case the tracking error becomes greater than some threshold value the procedure described above is repeated with new adapted parameters.
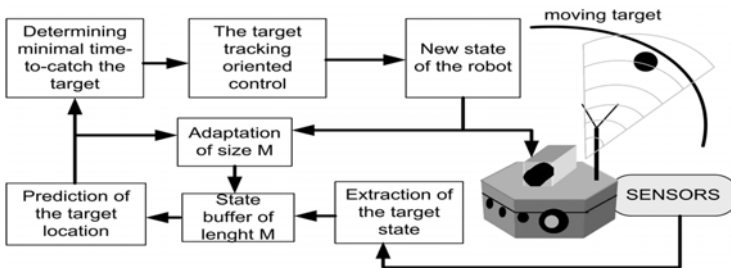
Fig. 1 Diagram of control system

1.2 The Model of Robot

Hereafter the control of the unicycle mobile robot is considered. Such robot is controlled by changing velocities of its right and left wheel, but for the purpose of this work the control is defined as:

$$u = [v, \omega], \quad v \in \langle 0, v_{max} \rangle, \quad \omega \in \langle -\omega_{max}, \omega_{max} \rangle, \tag{1}$$

where (v, ω) correspondingly, denote linear and angular velocity of the robot. Motion control algorithm discussed in further part of this paper generates set points (1) for the motion controller. It is necessary to include in computation the real acceleration profile of the controller. This profile is used to determine predictive control of the robot. The controller uses linear acceleration profile. The values of the accelerations used in experiments are equal $a_v = 20 \, [cm/s^2]$ and $a_\omega = 424 \, [°/s^2]$.

2 Problem Statement

In order to clearly address the problem let us introduce the notation that will be used hereafter. Let us denote the pose of the robot at discrete moment in time n as a vector $\mathbf{X}(n) = [x_r(n), y_r(n), \theta_r(n)]$, where its elements determine the location and the heading of the robot. The tracked target moves along a trajectory a model of which is unknown to the robot control system. The location of the target in time n is denoted by $\mathbf{P}(n) = [x_T(n), y_T(n)]$. Now the problem can be stated as follows: Find the control trajectory of the robot that allows to reach the target within the shortest time $T_{min} = 1, 2, \ldots$ Thus the solution of the problem is the vector:

$$\mathbf{U}^* = [u(n), u(n+1), \ldots, u(n + T_{min} - 1)] \tag{2}$$

that provides $\|\mathbf{X}(T_{min}) - \mathbf{P}(T_{min})\| \leq \varepsilon_T$ where the distance is measured according to euclidean norm and the threshold ε_T is assumed accuracy of tracking.

3 Adaptive Prediction Method

The problem of prediction considered in this paper is stated as follows: Using $M+1$ past observations of the target location:

$$\mathbf{P}(n-M), \ldots, \mathbf{P}(n-2), \mathbf{P}(n-1), \mathbf{P}(n) \tag{3}$$

find an estimate of the target location $\hat{\mathbf{P}}(n+H)$ in the future. The value H hereafter will be named the horizon of the prediction. The method proposed in this paper can be described as follows. At any given time n, using $M+1$ past observations of the target location (3) the model of the target trajectory is estimated. The key issue of the proposed approach is to find the trajectory for each coordinate of the target location separately. Therefore the following second order model is used:

$$\hat{\mathbf{P}}(h) = \begin{bmatrix} \hat{x}_T = f_x(h) \\ \hat{y}_T = f_y(h) \end{bmatrix} = \begin{bmatrix} a_x h^2 + b_x h + c_x \\ a_y h^2 + b_y h + c_y \end{bmatrix} \quad h = n - M, n - M - 1, \ldots \tag{4}$$

Unknown coefficients a, b, and c can be computed by doing a least squares fit. Using the model (4) the future location of the target can be determined. It is easy to distinguish two main issues that influence an accuracy of the prediction method stated above. The first is predictability of the target. If the target changes a direction of its movement very rapidly that means its movement is unpredictable and the method fails. The second is the number M of observations of a state of the target. But assuming the trajectory is predictable it is possible to adapt the parameter M. In this work the method of adaptation of this parameter, described in [7] was applied. Although the method is very simple the multiple simulations proved its efficiency. The algorithm provides satisfactory prediction simultaneously reducing the number of M to the range $M \in \langle 3, 5 \rangle$.

4 Time Optimal Tracking

Using the prediction of the target location the control (2) that enables the robot reaching the target in the minimal time $H_{\min}$ must be found. The problem of time minimal control is well known and there are many successive solutions of the problem in case the trajectory of the target is known. Moreover even in such case the solution methods are quite complex what implies difficulties in their implementation. Therefore we found reasonable to revisit the problem and to find as simple as possible solution of the problem.

4.1 Estimation of Minimal Time Horizon

First step of the algorithm is to determine minimal time horizon $H_{\min}$. Let us introduce the notation that will be used hereafter. The value n determines the time the process of tracking starts in. The $v(n), \omega(n)$ are linear and angular velocities of the robot moves in time n. The $\theta_d(n + H_{\min})$ is the desired heading of the robot necessary to reach the target in the given time. The value:

$$\Delta\theta(H_{\min}) = |\theta_d(n + H_{\min}) - \theta(n)| . \tag{5}$$

is an increment of heading of the robot necessary to reach the target in the minimal time $H_{\min}$. Linear translation of the robot id defined:

$$L(H_{\min}) = \left\| \hat{P}(n + H_{\min}) - X(n) \right\| . \tag{6}$$

The distance (6) is calculated according to euclidean norm. The $t_{\omega\max}$ and $t_{v\max}$ are times the robot needs to reach correspondingly its maximal angular and linear velocity. They are calculated according to:

$$t_{\omega\max} = \frac{\omega_{\max} - \omega(n)}{a_\omega}, \qquad t_{v\max} = \frac{v_{\max} - v(n)}{a_v} \tag{7}$$

Step 1: Set $H_{min} = 1$

Step 2: Calculate the speed of the robot after the time t, according to:
$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} v_0 + a_v t \\ \omega_0 + a_\omega t \end{bmatrix}$$

Step 3: Check if $v(t) < v_{max} \wedge \omega(t) < \omega_{max}$

Step 3.1: If yes, calculate the translation and rotation of the robot according to:
$$\begin{bmatrix} s(t) \\ \alpha(t) \end{bmatrix} = \begin{bmatrix} v_0 t + 0.5 a_v t^2 \\ \omega_0 t + 0.5 a_\omega t^2 \end{bmatrix}$$

　　　Go to step 4th.

Step 3.2: If no, calculate the translation and rotation of the robot according to:
$$\begin{bmatrix} s(t) \\ \alpha(t) \end{bmatrix} = \begin{bmatrix} v_0 t_{vmax} + 0.5 a_v t_{vmax}^2 + (t - t_{vmax}) v_{max} \\ \omega_0 t_{\omega max} + 0.5 a_\omega t_{\omega max}^2 + (t - t_{\omega max}) \omega_{max} \end{bmatrix}$$

Step 4: Check if $\alpha(t) < \Delta\Theta(t) \wedge s(t) < L(t)$

Step 4.1: If yes $T_{min} = H_{min}$
　　　Go to step 5th.

Step 4.2: If no $H_{min} = H_{min} + 1$, $t = H_{min} \Delta t$
　　　Go to step 2nd.

Step 5: Stop:

Fig. 2 The algorithm that determines the minimal time needed to reach the target

4.2 Target Tracking

If the prediction was perfect the problem would be rather simple. Unfortunately it is not, and the control algorithm has to cope with this fact. In this paper the method based on Game Theory is proposed. The process of determining the control of the robot moving toward the target is modeled as the game between two imaginary players. The first one could be perceived as control system that 'wants' to minimize the distance between the robot and the target. The second one is nature which is perceived as a source of uncertainty and errors. Let us define the game associated with each discrete moment in time:

$$G(k) = \{N, I, A\} \quad k = n, n+1, \ldots, H-1, \tag{8}$$

where N is a number of players (in this case $N = 2$), $I : A \rightarrow \Re$ is the cost function which determines costs associated with particular actions adopted by players. In this case in time k the function is defined as $I(a_1, a_2) = f(a_1(k), a_2(k), X(k), \hat{P}(k+H))$ where $a_{1/2} \in A_{1/2}$ are actions adopted by first and second player correspondingly and $A = A_1 \times A_2$ is the action space of the game. The action of the robot is the control defined by (2). Since it is assumed that there is finite number of actions the system

can adopt, the control space must be discretized. In this work the discretization was made around some distinctive values v_0, ω_0 with a precision $\Delta v, \Delta \omega$:

$$A_1 = V \times \Omega, \quad A_1 = \{(v, \omega) : v \in V \wedge \omega \in \Omega\},$$
$$V = \{v_0, v_0 + \Delta v, v_0 + 2\Delta v, \dots, v_0 + N_V \Delta v\},$$
$$\Omega = \{\omega_0 - N_\Omega \Delta \omega, \dots, \omega_0 - \Delta \omega, \omega_0, \omega_0 + \Delta \omega, \dots, \omega_0 + N_\Omega \Delta \omega\},$$

and N_V, N_Ω are the number of discrete values of velocities. The values v_0, ω_0 are determined in such a way that enable to reach the predicted location of the target after the time H the shortest way.

Second player symbolizes nature that is a source of uncertainty of prediction. Since the model of discrete game is considered finite set of possible actions must be determined. Particular 'actions' of nature are perceived as deviations of the estimations of the target location from its original value. In the discrete process some number of representative values of an overall space has to be taken into account. In the approach presented in this work a space around the $\hat{P}(n+H)$ is divided into a number of circular sectors inside of which an exemplary value is chosen in a random way. So the action space of the second player is given by:

$$A_2 = \Lambda \times R \quad A_2 = \{(\alpha, r) : \alpha \in \Lambda \wedge r \in R\}, \tag{9}$$

where

$$\begin{aligned}
\Lambda &= \{\alpha_i\}, \quad \alpha_i = \tfrac{2\pi}{L}(i + \delta_\alpha - 1), \quad i = 1, 2, \dots, K \\
R &= \{r_j\}, \quad r_j = \Delta r(j + \delta_r - 1), \quad j = 1, 2, \dots, J
\end{aligned} \tag{10}$$

The δ_r, δ_α in (10) are random numbers.

The cost function applied in this work has two-component form:

$$I(a_1, a_2) = \hat{L}_{R,T}(a_1) + k_e \left| \hat{L}_{R,T}(a_1) - \hat{L}^*_{R,T}(a_1, a_2) \right| \tag{11}$$

where $\hat{L}_{R,T}(a_1)$ is predicted distance between the robot and the target determined from:

$$\begin{aligned}
\hat{L}_{R,T}(a_1) &= \sqrt{\left(\hat{x}_T - \hat{x}_R^{a_1}\right)^2 + \left(\hat{y}_T - \hat{y}_R^{a_1}\right)^2}, \\
\hat{x}_R^{a_1} &= x_R + v(a_1)\Delta t \cos(\Theta_R + \omega(a_1)\Delta t), \\
\hat{y}_R^{a_1} &= y_R + v(a_1)\Delta t \sin(\Theta_R + \omega(a_1)\Delta t),
\end{aligned}$$

where $v(a_1), v(a_2)$ denote values of linear and angular velocity set as a result of choosing the action $a_1 \in A_1$. Similarly (12) denotes a distance between the robot and the target that will be if the first player applies the action a_1 and an error of prediction represented by a_2 appears:

$$\begin{aligned}
\hat{L}^*_{R,T}(a_1, a_2) &= \sqrt{\left(\hat{x}_T^{a_2} - \hat{x}_R^{a_1}\right)^2 + \left(\hat{y}_T^{a_2} - \hat{y}_R^{a_1}\right)^2}, \\
\hat{x}_T^{a_2} &= \hat{x}_T + r(a_2)\cos(\alpha(a_2)), \\
\hat{y}_T^{a_2} &= \hat{y}_T + r(a_2)\sin(\alpha(a_2)).
\end{aligned}$$

The solution of the problem given by (8) is the action a_{10} that provides target-oriented motion of the robot. Moreover this action should soften results of inaccurate prediction. In case of high uncertainty conditions good solution is so called min-max strategy that provide obtaining 'the best from the worst' solution. Thus we have:

$$a_{10} = \min_{a_1 \in A_1} \max_{a_2 \in A_2} I(a_{10}, a_2). \qquad (12)$$

Applying (12) in each time the control vector (2) is obtained that enables robot to reach the target location in the presence of uncertainty. Certainly $\hat{P}(n+H)$ is only the prediction of the target location calculated according the model (4) which often differ much from a real location of the target. So if we want to reach the real location of the target with a given accuracy a feedback mechanism has to be applied.

5 Simulation Results

In order to verify the proposed approach a number of simulations were performed using MATLAB. The simulated process was a typical tracking problem with additional time constraints. The problem consists in tracking by the robot the target (moving along unknown trajectory) in this way to reach the target location after minimal time $H_{\min}$. Results of two simulations are presented in Fig. 3a and b. A plot of a part of the trajectory of the target P is depicted with a solid line. The robot is depicted by a polygon. The process of tracking starts in time $t_n = t_{n0}$.

Initial state of the robot and the target are denoted by $X(t_n = t_0)$ and $P(t_n = t_0)$ respectively. Successive stages of prediction are depicted with crosses and estimates of the location for the minimal time horizon are depicted with circles. The assumed accuracy of tracking is equal $\varepsilon_T = 3\,[cm]$. The sampling time $\Delta T = 0.1\,[s]$. Both in first and second the robot captured the the target in the minimal time. In first experiment the algorithm needed to make two corrections of the estimate the target location. In second one only one was needed. It is worth to notice that the direction
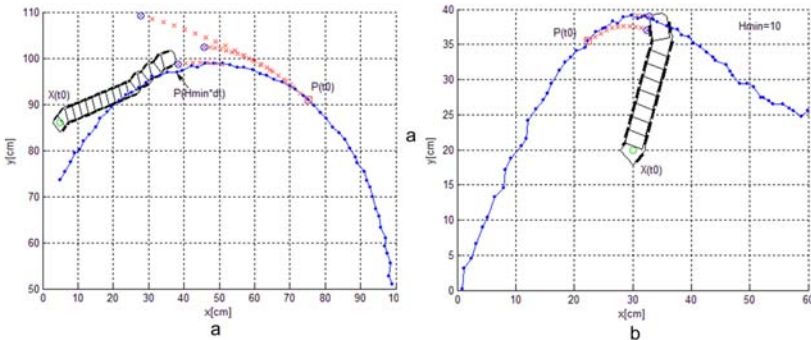
Fig. 3 Results of Tracking Experiments

of movement is not the simple, target driven one. The algorithm takes into account possible errors of prediction.

6 Conclusion

This paper presents an approach to a synthesis of the robust algorithm of tracking moving target. The model of the tracking process is built as the game between player (robot) and nature (errors of prediction). Discussed approach gives good results what proved multiple simulations. Analyzing obtained results a few conclusions can be drawn. First thing is that the tracking algorithm is convergent. The control generated by presented algorithm has the property that it still pushes the robot toward the target location. On the other hand the algorithm can not guarantee that the target will be reached in an assumed minimal time. The main cause of this fact is the target trajectory is unpredictable. But great number of experiments proves that the algorithm works well in most analyzed cases. Additional advantages of the proposed method of prediction as well as the tracking method are their simplicity and robustness to uncertain data.

References

1. Aström, K., Wittenmark, B.: Computer-Controlled Systems. Prentice-Hall, Englewood Cliffs (1997)
2. Gass, S.: Predictive fuzzy logic controller for trajectory tracking of a mobile robot. In: Proceedings of the Mid-Summer Workshop on Soft Computing in Industrial Applications Helsinki University of Technology, Espoo, Finland (2005)
3. Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., Nordlund, P.J.: Particle filters for positioning, navigation, and tracking. IEEE Transactions on Signal Processing, 425–437 (2002)
4. Julier, S., Uhlmann, J.: A new extension of the Kalman filter to nonlinear systems. In: Proceedings of the 11th International Symposium On Aerospace/Defence Sensing, Simulation and Controls, Orlando, US. SPIE, vol. 3068, pp. 182–193 (1997)
5. Liao, L., et al.: Voronoi tracking: Location estimation using sparse and noisy sensor data. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (2003)
6. Riley, M., Ude, A., Atkeson, C.: Methods for motion generation and interaction with a humanoid robot: Case studies of dancing and catch. In: Proceedings of the AAAI and CMU Workshop on Interactive Robotics and Entertainment, Pittsburgh, US (2000)
7. Skrzypczyk, K.: Robust prediction and tracking method for a mobile robot navigation. In: Proceedings of the International Supercomputing Conference, Palermo, US (2006)

Improving Quality of Satellite Navigation Devices

Krzysztof Tokarz and Michał Dzik

Abstract. Global Positioning System (GPS) technology has become an essential tool used for localization, navigation, tracking, mapping and timing. The typical accuracy of position measurement for stationary receivers and these in motion vary from 3 to 20 m. This accuracy is affected by the variety of factors such as fluctuation of time in which the signal passes through ionosphere and troposphere, inaccurate time measuring in the receiver, or the multipath propagation caused by reflections from buildings and other objects. The localization accuracy can be improved by differential measuring, usage of some additional sensors, or postprocessing. The article presents discussion of possibilities of increasing position measurement accuracy to be applied in the walking assistant device, designed to enhance quality of life for blind people by helping them in outdoor activities within the urban environment.

Keywords: GPS, navigation, accuracy, DGPS.

1 Introduction

GPS was first developed in the late 1970s for the military use, with the first satellites launched in 1978. Although GPS was made officially available to public use in 1995, first civilian receivers were available in the mid-1980s and their accuracy was initially intentionally degraded to 100 meters by dithering satellite clocks. This technique called Selective Availability (SA) was turned off in 2000. With SA off, common devices could reach the accuracy of 20 meters, but with time it improved to 3–5 meters.

The most popular area of use for civilian GPS receivers is navigation of moving vehicles – on ground, sea or sky. Vehicles, for example cars, are generally moving

Krzysztof Tokarz · Michał Dzik
Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
e-mail: {Krzysztof.Tokarz,Michal.Dzik}@polsl.pl

fast and in concrete direction and the navigation GPS receivers are optimized for this type of motion. They obtain the best results in determining position and direction when the user is moving with speed of at least 20 km/h. There are also some tourist receivers for pedestrians, intended for usage with lower speed. Their accuracy allows the walker to keep the azimuth but cannot lead him precisely along the path.

Accuracy of measurement is acceptable when results are not critical (as for the tourist usage). For advanced applications, for example aviation, the accuracy of popular receivers is insufficient and special techniques must be used to improve it. To do this the sources of errors should be known.

Worsening of the accuracy is caused mainly by the inaccurate measurement of time in which the signal reaches the receiver. Passing through the atmosphere can take different time, depending on conditions. While space segment (satellites) is well synchronized by atomic clocks and auxiliary data from control stations on Earth, receivers use cheap and not precise quartz clocks, which cause fix errors. Signals can also reflect from high objects around the user, causing the multipath propagation or interference with other signals.

Attempts at improving accuracy apply several techniques. The most important is differential technique with using more than one GPS receiver. One receiver, stationary and placed in some known position, measures the current error and sends the result to the other device. Another technique bases on statistical postprocessing the data. Very good results can be obtained with measuring the phase of carrier signal instead of the code but the cost of devices using this technique is very high.

The paper presents investigations on improving the accuracy of GPS positioning used in the walking assistant device, designed to help visually impaired people to move in the urban environment. The device enables the user to determine the azimuth, even while not moving. It can also lead the user to the destination by audio messages with description of points passed by. The localization accuracy with GPS receiver the device is equipped with enables establishing position to within 3 meters which cannot be relied upon as entirely safe, thus there is provided discussion on ways of improving it.

2   GPS – How Does It Work?

The Global Positioning System is made up of three segments: Space, Control and User. The Control segment consists of 12 ground stations [3] with 7 Monitor Stations, 3 Antenna Stations and one Master Control Station. In the Master Control Station the signals from Monitor Stations are processed and using the Antenna Stations send to the satellites that form the Space segment. The Space segment consists of 30 satellites with at least 24 in use. In the User segment there are included all receivers with capability of receiving and processing signals from currently visible satellites.

In the GPS system, determining position is based on measuring distance between satellites and a receiver. It is calculated from time which it takes the signal to move from a satellite to the receiver. The system uses two channels, L1 and L2. The carrier frequency of L1 signal is 1575.42 MHz, while for L2 it is 1227.60 MHz. L2 signal is

encrypted and available only for US military users. Satellites use the carrier signal to send the navigation message that carries parameters to the receiver, some correction data and information about the state of satellite. It takes 12.5 minutes to sent the whole message but important navigation parameters are sent every 30 s. Signals sent from the satellites are modulated using CDMA to allow all satellites to broadcast on the same frequency. Each satellite has its own set of codes used for distance measurements and for identifying the satellite. Codes have the length of 1023 bits and are sent out at bitrate of 1.023 MHz, so it takes 1ms to send one code. On L2 frequency the codes are sent with bitrate of 10.23 MHz allowing for higher precision of measurements.

The receiver using its own clock generates the copy of codes and compares them to codes received. As the result it obtains the time in which the signal goes from the satellite to the receiver, which can be used to calculate the range between the receiver and satellites.

Current position of the receiver is calculated using at least three satellites in field of sight. Knowing the time it takes the signal from these satellites it is possible to measure the distance to them. Using trilateration method (Fig. 1) it is easy to calculate current position of the receiver. By measuring the distance to one point it is known that the position is on a circle around that point. Measuring against another point narrows down the choices to the places where the circles intersect. A third measurement singles out the position. In three dimensions circles should be replaced with spheres.

Fig. 1 The principle of trilateration
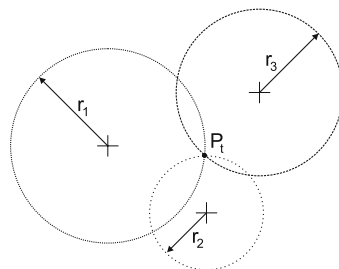
The simple basis of the GPS is to calculate a position from the distances to three known positions in space. By noting the time it takes a signal to go from a satellite to the receiver the distance between the satellite and the receiver can be measured. Then it is only a matter of geometry to compute the user position.

The unknown position of the receiver, the user position, is designated

$$\vec{x}_u = \begin{pmatrix} x_u \\ y_u \\ z_u \end{pmatrix}, \tag{1}$$

while the known position of satellite k $(H = 1, 2, 3, \ldots)$ is

$$\vec{x}^{(k)} = \begin{pmatrix} x^{(k)} \\ y^{(k)} \\ z^{(k)} \end{pmatrix}. \tag{2}$$

This gives the system of non-linear equations

$$r^{(1)} = \sqrt{(x_u - x^{(1)})^2 + (y_u - y^{(1)})^2 + (z_u - z^{(1)})^2}\,, \tag{3}$$

$$r^{(2)} = \sqrt{(x_u - x^{(2)})^2 + (y_u - y^{(2)})^2 + (z_u - z^{(2)})^2}\,,$$

$$r^{(3)} = \sqrt{(x_u - x^{(3)})^2 + (y_u - y^{(3)})^2 + (z_u - z^{(3)})^2}\,.$$

Unfortunately such simple solution is not possible because there are number of error sources that influence the accuracy of measurements. Even when installed in the stationary position a simple GPS receiver taking measurements returns a typical scatter of points, as depicted in Fig. 2, one unit of the grid representing 10 meter distance.

Fig. 2 A scatter of the points given by a GPS receiver installed in a fixed position
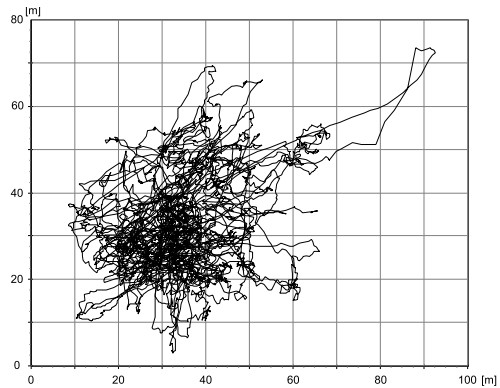
3　Sources of Errors

There are several sources of measurement errors [1], the first group of which constitute timing errors related to inaccuracy of a receiver and satellites clocks, relativistic effects and errors caused by the control segment. Second important group of errors is related to disturbances during signal propagation especially when passing through the ionosphere. Errors can also result from the noise of elements in the receiver and other electronic devices placed near the receiver.

These errors can be characterised as follows:

a) Timing errors – Quartz oscillators used in receivers have the advantage of being cheap, light, small and consuming little power but they are not as stable as

atomic frequency sources used in GPS satellites. Special techniques are used to correct the clock frequency and synchronize it with the incoming signal but changes in environmental factors (temperature, pressure) always influence the oscillator stability. Satellites use the atomic frequency sources but they also have the drift that must be corrected by the control segment, which calculates the current position of satellites and also predicts their position in the close future. This prediction is valid for some time and the error increases with time from the last update.

b) Ionospheric and tropospheric delay – Ionosphere slows the radio signals and propagation time depends on the ionization caused mainly by UV radiation from the Sun, changing during the day and related to the season and solar activity. The delay depends on the length of the path that the signal must pass through the ionosphere thus different elevation of satellites causes different length of the path. GPS receivers use the model of the ionospheric delay with parameters calculated in the central station and broadcasted by the satellites. Using both L1 and L2 signals could fix the error caused by ionospherical delay but L2 signal is encrypted and available only for US military users. Troposphere also causes the delay but with smaller influence on accuracy than ionosphere. There is no special signal broadcasted to reduce the tropospheric effect so receivers usually use the model based on the current season and geographic position.

c) Multipath propagation – The signal can be reflected by the ground or other objects, reaching the receiver directly and passing some longer way, which causes that main signal interfere with its own delayed copy. Although the reflected signal is less strong than the original one, it can affect the amplitude and phase of the received signal. This factor is especially important in environments with many high objects as the city canyons [4]. The solution for avoiding reflection from the ground is special antenna construction, shielded at the bottom. The simple solution used to lower the influence of reflections from other objects is calculating the average of the measurements.

d) Satellite geometry – The most advantageous position of satellites happens when one satellite is directly above the receiver and three others equally spaced around the horizon. The worse case is when the satellites are close to each other. Special parameter called Dilution Of Precision (DOP) is defined to describe the influence of satellite geometry on the accuracy.

e) Selective Availability – Selective Availability is the name of the technique used for intentional dithering of timing of the satellite signal to lower the accuracy of civilian receivers. It is inactive from May 2000 but it can be always switched on in state of military or terrorist threat. It can lower the positioning accuracy is much higher degree than any other error source.

All these errors can accumulate to measuring position with error per satellite ranging from the distance below half a meter to even a hundred meters as specified by Table 1.

Table 1 Typical errors for different sources (in meters per satellites)

Error Type	Standard GPS error
Satellite Clocks	2.0
Receiver Clock	3.0
Orbit Errors	2.5
Ionosphere	5.0
Troposphere	0.5
Receiver Noise	0.3
Multipath	1.0
Selective Availability	100

4 Increasing Accuracy

Position determination errors described in the previous section can be minimized by using better receivers, with higher sensitivity, lower noises or more precise clock. Military users can use L2 signal, but in low-cost civilian receivers some auxiliary methods can be used to improve quality:

a) Differential GPS – DGPS corrections are used to minimize affect of atmosphere and are calculated by stations that know their exact geographical coordinates. In such station the position determined by the satellite navigation receiver is compared with the true position and calculated error broadcasted as correction data. Corrections depend on atmospheric influence, hence differential corrections are only usable near to the measurement station and at the time they were prepared [2]. They must be sent to receivers immediately, so they are broadcasted by radio, but on different frequencies than GPS signals [5]. Although DGPS improves quality, it is available in only few places. In fact they can be used mostly in port towns, where the signal is used for marine navigation.

b) Wide Area Augmentation System (WAAS) – Differential corrections are sent also by satellites. Two most popular systems are European Geostationary Navigation Overlay System (EGNOS) and WAAS in North America. Both systems consist of groups of measurement stations (38 stations for WAAS and 34 for EGNOS), which are spread over some area and calculate corrections. These corrections are next collected in control stations and sent to geostationary satellites (2 for WAAS and 3 for EGNOS). Satellites transmit data on standard GPS L1 frequency. The greatest limitation of such systems is the fact that they use geostationary satellites, which are located over the equator. On greater geographic longitudes satellites are seen low over the horizon and can be easily blocked out by high objects.

c) SISNeT – The GPS corrections calculated by EGNOS system can be downloaded from the Internet using wired or wireless connection [9]. The project SiSNET uses the Internet to make EGNOS corrections data available to interested users. Data can be sent to mobile devices by wireless connection for

example WiFi local network or over cellular phone systems using GPRS or UMTS technique.

All these improving techniques can reduce the overall error from about 15 meters to less than 3 meters.

5 Walking Assistant Device

Walking assistant device is designed to help blind people to walk independently. It is a portable device equipped with GPS receiver and communicating by audio messages. The device uses the set of defined points and can guide a blind person to any chosen point. Points are defined by their geographical coordinates and stored in non-volatile memory. They can be organized in routes and the user can be guided through all of them. The device can also read out names of passed points. It is useful the area known by the user and can be used in public transport to read names of approaching stops.

Walking assistant devices still cannot fully replace white canes used by blind persons. The device cannot warn the user against stairs or a pothole in the sidewalk. Such obstacles are not presented on any map neither are they constant. Changing conditions would cause any such map to be out of date quickly but there is also the problem of low accuracy of satellite navigation. The parameters of the applied GPS receiver FGPMMOPA2 with capability of receiving DGPS signal are given by Table 2.

Table 2 Characteristics of the FGPMMOPA2 GPS receiver

Parameter	
Chipset	MTK MT3318
Number of channels	32, all-in-view tracking
Sensitivity	$-158\,dBm$
GPS protocol	NMEA 0183 V 3.01
DGPS	RTCM/EGNOS/WAAS/MSAS
Average acquisition rate	1 second

The primary objective of the device is to guide a user through a grid of streets. In such situation the required accuracy is not very high, but it cannot be low either. 3-meter error might mean for the blind user that he is in the middle of the roadway instead of the sidewalk. City environment causes degradation of the accuracy of satellite navigation and to achieve sufficient quality of navigation, some of described methods were applied.

To determine direction the azimuth the device uses magnetic compass that measures Earth magnetic field with magnetoresistive sensors. Azimuth can be also calculated by GPS receiver itself, but the receiver can fix it only while moving in one

direction for few seconds. The compass can determine directions and guide the user in correct way even before he starts walking.

To communicate with the user the device uses the microphone and earphones. Since blind persons are using in everyday life mainly their hearing sense, so all sounds can be turn down with one switch. All messages are recorded and are stored in the device as audio files. The device itself is unable to synthesize speech. As to not disturb the blind user, guiding in correct direction is done by a single tone. This sound is spatial – it is heard from the direction in which the user should turn. The input is provided by keyboard with descriptions written in braile.

6 Ways of Position Accuracy Improvement

The accuracy of the GPS receiver used in the walking assistant device is insufficient to be the only source of navigation for a visually impaired person. Therefore, navigation methods require some improvement, if not by increasing accuracy of GPS which for small and cheap devices is next to impossible, then by applying other techniques.

DGPS solution is the first choice method. Unfortunately the accuracy it provides is still not enough. Another drawback of this solution is that the only way to receive the correction data in the urban environment is by using cellular telephony data channel such as GPRS. It requires active GPRS connection that costs some energy which significantly shortens the time that the device can work without recharging the battery.

Application of software solutions like using statistical algorithms basing on series of measurements taken is inadequate to the main purpose of the walking assistant device because of poor results in processing short term and rather fast changing data.

GPS receiver is capable of fixing a direction in which a user in moving (azimuth). It done by keeping in memory results from some previous location measurements. These points make a line that points the direction in which the device is moving. Unfortunately this technique fails in case of pedestrians because their moving speed is too low and a distance passed between subsequent measures is comparable to error caused by inaccuracy of measures. The solution for improving the accuracy of fixing direction is by using the classical compass which measures Earth's magnetic field and can determine how a user is turned toward north direction even if he does not move. It is very important for blind persons because one can be guided from the very first moment, when he activates this function. The compass also reacts immediately to changes of direction when the user is walking.

Another possible element that can improve the navigation accuracy of the device is the accelerometer. The main purpose to use it is to help the electronic compass to determine the direction. While using the compass with two magnetoresistive sensors the surface of both sensors must be horizontal, parallel to Earth surface. Usage of the accelerometer allows to measure the angle between the gravitational acceleration vector that is always perpendicular to the Earth surface and the current position of

the device. The accelerometer can be also used for improving the calculation of current position of the device. Measurements can be processed with the algorithm that recalculates them to the distance of movement and results can be used for correcting the current position determined by GPS receiver. Such method has been proposed in [8, 7] for pedestrian movement and in [6] for vehicle tracking.

7   Conclusions

Although the accuracy of GPS receiver is not very high the proposed device can be helpful for people with visual impairment to move around the urban environment. From the above discussion of sources of errors and present solutions to rectify them, it is rather clear that without changes in GPS system it is impossible to obtain significantly better accuracy in civilian low cost receivers. However, improvements can be done with DGPS technique with data downloaded from the Internet. It is also possible to develop and apply algorithms for calculating the data from additional sensors like accelerometer.

References

1. Di Lecce, V., Amato, A., Piuri, V.: Neural technologies for increasing the GPS position accuracy. In: Proceedings of the IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, Istanbul, Turkey (2008)
2. Dimc, F., Music, B., Osredkar, R.: Attaining required positioning accuracy in archeogeophysical surveying by GPS. In: Proceedings of the 12th International Power Electronics and Motion Control Conference, Portoroz, Slovenia (2006)
3. Fang, R.J., Su, K.I., Lu, H.C., Wang, C.C., Lin, C.C.: Application of Global Positioning System (GPS) in Earth Sciences teaching. In: Proceedings of the 6th WSEAS International Conference on Applied Computer Science, Hangzhou, China, pp. 267–271 (2007)
4. Hentschel, M., Wulf, O., Wagner, B.: A GPS and laser-based localization for urban and non-urban outdoor environments. In: Proceedings of the International Conference on Intelligent Robots and Systems, Nice, France (2008)
5. Huang, G.S.: Application of the vehicle navigation via GPS carrier phase. In: Proceedings of the 6th WSEAS International Conference on Robotics, Control and Manufacturing Technology, Hangzhou, China, pp. 218–223 (2006)
6. Jackson, J.D., Callahan, D.W., Wang, P.F.: Location tracking of test vehicles using accelerometers. In: Proceedings of the 5th WSEAS International Conference on Circuits, Systems, Electronics, Control & Signal Processing, Dallas, US, pp. 333–336 (2006)
7. Mezentsev, O., Collin, J., Kuusniemi, H., Lachapelle, G.: Accuracy assessment of a high sensitivity GPS based pedestrian navigation system aided by low-cost sensors. Gyroscopy and Navigation 4(47), 49–64 (2004) (in Russian)
8. Petovello, M., Mezentsev, O., Lachapelle, G., Cannon, M.E.: High sensitivity GPS velocity updates for personal indoor navigation using inertial navigation systems. In: Proceedings of the Navigation GPS Conference, Portland, US (2003)
9. Sandhana, L.: GPS to help the blind navigate (2003),
   http://www.wired.com/medtech/health/news/2003/06/59174

Author Index